Manifold representations of musical signals and generative spaces

Axel Chemla-Romeu-Santos

Università Degli Studi di Milano
Department of Computer Science

Université Paris-Sorbonne
École doctorale Informatique,
Télécommunications et Électronique (Paris)

PhD thesis in partnership

Manifold representations of musical signals and generative spaces

Axel Chemla–Romeu-Santos
Matricolo : R11707

Academic year 2018-2019
XXXIIth cycle
PhD defense : Jan 30, 2020

Director :
Goffredo HAUS

Director:
Gérard ASSAYAG

Co-director :
Philippe ESLING

Coordinator :
Paolo BOLDI

Coordinator :

Jury committee :

Pr. Antonio CAMURRI (president)
Pr. Federico FONTANA
Pr. Antonio RODÁ
Pr. Goffredo HAUS
Pr. Nicolas BODINI
Pr. Nicolas BREDECHE
Pr. Geoffroy PEETERS
Pr. Philippe ESLING

Details

This PhD, funded by the Computer Science Department of Universitá Degli Studi di Milano in the Laboratorio d'Informatica Musicale (LIM), has been accomplished in partnership with the doctoral school EDITE of Sorbonne Universités, in the IRCAM STMS lab.

Referees

Advisors

- Pr. Goffredo HAUS
- Pr. Gérard ASSAYAG

Co-advisor

- Pr. Philippe ESLING

Reviewing Committee

- Pr. Shlomo DUBNOV
- Pr. Davide ROCCHESSO
- Pr. Michèle SEBAG

Defense Committee

- Pr. Antonio CAMURRI (president)
- Pr. Federico FONTANA
- Dr. Antonio RODÁ
- Pr. Goffredo HAUS
- Pr. Geoffroy PEETERS
- Pr. Nicolas BREDECHE
- Pr. Olivier BODINI
- Pr. Philippe ESLING

Abstract (english)

Among the diverse research fields within computer music, *synthesis* and *generation* of audio signals epitomize the cross-disciplinarity of this domain, jointly nourishing both scientific and artistic practices since its creation. Inherent in computer music since its genesis, audio generation has inspired numerous approaches, evolving both with musical practices and scientific/technical advances. Moreover, some synthesis processes also naturally handle the reverse process, named *analysis*, such that synthesis parameters can also be partially or totally extracted from actual sounds, and providing an alternative *representation* of the analyzed audio signals.
On top of that, the recent rise of machine learning algorithms earnestly questioned the field of scientific research, bringing powerful data-centred methods that raised several epistemological questions amongst researchers, in spite of their efficiency. Especially, a family of machine learning methods, called *generative models*, are focused on the generation of original content using features extracted from an existing dataset. In that case, such methods not only questioned previous approaches in generation, but also the way of integrating this methods into existing creative processes. While these new generative frameworks are progressively introduced in the domain of image generation, the application of such generative techniques in audio synthesis is still marginal.

In this work, we aim to propose a new audio analysis-synthesis framework based on these modern generative models, enhanced by recent advances in machine learning. We first review existing approaches, both in sound synthesis and in generative machine learning, and focus on how our work inserts itself in both practices and what can be expected from their collation. Subsequently, we focus a little more on generative models, and how modern advances in the domain can be exploited to allow us learning complex sound distributions, while being sufficiently flexible to be integrated in the creative flow of the user. We then propose an *inference / generation* process, *mirroring* analysis/synthesis paradigms that are natural in the audio processing domain, using *latent models* that are based on a continuous higher-level space, that we use to *control* the generation. We first provide preliminary results of our method applied on spectral information, extracted from several datasets, and evaluate both qualitatively and quantitatively the obtained results. Subsequently, we study how to make these methods more suitable for learning audio data, tackling successively three different aspects. First, we propose two different latent regularization strategies specifically designed for audio, based on and signal / symbol translation and perceptual constraints. Then, we propose different methods to address the inner *temporality* of musical signals, based on the extraction of multi-scale representations and on prediction, that allow the obtained generative spaces that also model the dynamics of the signal.

As a last chapter, we swap our scientific approach to a more *research & creation*-oriented point of view: first, we describe the architecture and the design of our open-source library, `vsacids`, aiming to be used by expert and non-expert music makers as an integrated creation tool. Then, we propose an first musical use of our system by the creation of a real-time performance, called *aego*, based jointly on our framework `vsacids` and an explorative agent using reinforcement learning to be trained during the performance. Finally, we draw some conclusions on the different manners to improve and reinforce the proposed generation method, as well as possible further creative applications.

Abstract (italiano)

Tra i diversi campi di ricerca nell'ambito dell'informatica musicale, la sintesi e la generazione di segnali audio incarna la pluridisciplinalità di questo settore, nutrendo insieme le pratiche scientifiche e musicale dalla sua creazione. Inerente all'informatica dalla sua creazione, la generazione audio ha ispirato numerosi approcci, evolvendo colle pratiche musicale e gli progressi tecnologici e scientifici. Inoltre, alcuni processi di sintesi permettono anche il processo inverso, denominato *analisi*, in modo che i parametri di sintesi possono anche essere parzialmente o totalmente estratti dai suoni, dando una rappresentazione alternativa ai segnali analizzati.
Per di più, la recente ascesa dei algoritmi di l'apprendimento automatico ha vivamente interrogato il settore della ricerca scientifica, fornendo potenti *data-centered* metodi che sollevavano diversi epistemologici interrogativi, nonostante i sui efficacia. Particolarmente, un tipo di metodi di apprendimento automatico, denominati *modelli generativi*, si concentrano sulla generazione di contenuto originale usando le caratteristiche che hanno estratti dei dati analizzati. In tal caso, questi modelli non hanno soltanto interrogato i precedenti metodi di generazione, ma anche sul modo di integrare questi algoritmi nelle pratiche artistiche. Mentre questi metodi sono progressivamente introdotti nel settore del trattamento delle immagini, la loro applicazione per la sintesi di segnali audio e ancora molto marginale.

In questo lavoro, il nostro obiettivo e di proporre un nuovo metodo di audio sintesi basato su questi nuovi tipi di generativi modelli, rafforazti dalle nuove avanzati dell'apprendimento automatico. Al primo posto, facciamo una revisione dei approcci esistenti nei settori dei sistemi generativi e di sintesi sonore, focalizzando sul posto di nostro lavoro rispetto a questi disciplini e che cosa possiamo aspettare di questa collazione. In seguito, studiamo in maniera più precisa i modelli generativi, e come possiamo utilizzare questi recenti avanzati per l'apprendimento di complesse distribuzione di suoni, in un modo che sia flessibile e nel flusso creativo del utente. Quindi proponiamo un processo di inferenza / generazione, il quale rifletta i processi di analisi/sintesi che sono molto usati nel settore del trattamento del segnale audio, usando modelli latenti, che sono basati sull'utilizzazione di un spazio continuato di alto livello, che usiamo per *controllare* la generazione. Studiamo dapprima i risultati preliminari ottenuti con informazione spettrale estratte da diversi tipi di dati, che valutiamo qualitativamente e quantitativamente. Successivamente, studiamo come fare per rendere questi metodi più adattati ai segnali audio, fronteggiando tre diversi aspetti. Primo, proponiamo due diversi metodi di regolarizzazione di questo generativo spazio che sono specificamente sviluppati per l'audio : una strategia basata sulla traduzione segnali / simboli, e una basata su vincoli percettivi. Poi, proponiamo diversi metodi per fronteggiare il aspetto *temporale* dei segnali audio, basati sull'estrazione di rappresentazioni multiscala e sulla predizione, che permettono ai generativi spazi ottenuti di anche modellare l'aspetto dinamico di questi segnali.

Per finire, cambiamo il nostro approccio scientifico per un punto di visto piú ispirato dall'idea di *ricerca e creazione*. Primo, descriviamo l'architettura e il design della nostra libreria open-source, `vsacids`, sviluppata per permettere a esperti o non-esperti musicisti di provare questi nuovi metodi di sintesi. Poi, proponiamo una prima utilizzazione del nostro modello con la creazione di una performance in real-time, chiamata *ægo*, basata insieme sulla nostra libreria `vsacids` e sull'uso di une agente di esplorazione, imparando con rinforzo nel corso della composizione. Finalmente, tramo dal lavoro presentato alcuni conclusioni sui diversi modi di migliorare e rinforzare il metodo di sintesi proposto, nonché eventuale applicazione artistiche.

Abstract (français)

À travers les différents domaines de recherche de la musique computationnelle, l'*analysie* et la *génération* de signaux audio sont l'exemple parfait de la trans-disciplinarité de ce domaine, nourrissant simultanément les pratiques scientifiques et artistiques depuis leur création. Intégrée à la musique computationnelle depuis sa création, la synthèse sonore a inspiré de nombreuses approches musicales et scientifiques, évoluant de pair avec les pratiques musicales et les avancées technologiques et scientifiques de son temps. De plus, certaines méthodes de synthèse sonore permettent aussi le processus inverse, appelé *analyse*, de sorte que les paramètres de synthèse d'un certain générateur peuvent être en partie ou entièrement obtenus à partir de sons donnés, pouvant ainsi être considérés comme une *représentation* alternative des signaux analysés. Parallèlement, l'intérêt croissant soulevé par les algorithmes d'apprentissage automatique a vivement questionné le monde scientifique, apportant de puissantes méthodes d'analyse de données suscitant de nombreux questionnements épistémologiques chez les chercheurs, en dépit de leur effectivité pratique. En particulier, une famille de méthodes d'apprentissage automatique, nommée *modèles génératifs*, s'intéressent à la génération de contenus originaux à partir de caractéristiques extraites directement des données analysées. Ces méthodes n'interrogent pas seulement les approches précédentes, mais aussi sur l'intégration de ces nouvelles méthodes dans les processus créatifs existants. Pourtant, alors que ces nouveaux processus génératifs sont progressivement intégrés dans le domaine la génération d'image, l'application de ces techniques en synthèse audio reste marginale.

Dans cette thèse, nous proposons une nouvelle méthode d'analyse-synthèse basés sur ces derniers modèles génératifs, depuis renforcés par les avancées modernes dans le domaine de l'apprentissage automatique. Dans un premier temps, nous examinerons les approches existantes dans le domaine des systèmes génératifs, sur comment notre travail peut s'insérer dans les pratiques de synthèse sonore existantes, et que peut-on espérer de l'hybridation de ces deux approches. Ensuite, nous nous focaliserons plus précisément sur comment les récentes avancées accomplies dans ce domaine dans ce domaine peuvent être exploitées pour l'apprentissage de distributions sonores complexes, tout en étant suffisamment flexibles pour être intégrées dans le processus créatif de l'utilisateur. Nous proposons donc un processus d'*inférence / generation*, reflétant les paradigmes d'analyse-synthèse existant dans le domaine de génération audio, basé sur l'usage de *modèles latents* continus que l'on peut utiliser pour *contrôler* la génération. Pour ce faire, nous étudierons déjà les résultats préliminaires obtenus par cette méthode sur l'apprentissage de distributions spectrales, prises d'ensembles de données diversifiés, en adoptant une approche à la fois quantitative et qualitative. Ensuite, nous proposerons d'améliorer ces méthodes de manière spécifique à l'audio sur trois aspects distincts. D'abord, nous proposons deux stratégies de régularisation différentes pour l'analyse de signaux audio : une basée sur la traduction signal/ symbole, ainsi qu'une autre basée sur des contraintes perceptuelles. Nous passerons par la suite à la dimension *temporelle* de ces signaux audio, proposant de nouvelles méthodes basées sur l'extraction de représentations temporelles multi-échelle et sur une tâche supplémentaire de prédiction, permettant la modélisation de caractéristiques dynamiques par les espaces génératifs obtenus.

En dernier lieu, nous passerons d'une approche scientifique à une approche plus orientée vers un point de vue *recherche & création*. Premièrement, nous présenterons notre librairie open-source, `vsacids`, visant à être employée par des créateurs experts et non-experts comme un outil intégré. Ensuite, nous proposons une première utilisation musicale de notre système par la création d'une performance temps réel, nommée *ægo*, basée à la fois sur notre librarie et sur un agent d'exploration appris dynamiquement par renforcement au cours de la performance. Enfin, nous tirons les conclusions du travail accompli

jusqu’à maintenant, concernant les possibles améliorations et développements de la méthode de synthèse proposée, ainsi que sur de possibles applications créatives.

Acknoledgements

First, I would like to warmly thank Pr. Goffredo Haus for his trust during these three years, and for having me offered the chance of working on this fascinating topic. Then, I would also like to thanks and their valuable support during these three years : Giorgio, Luca, Adriano, Stavros, and Federico for their great welcome in the LIM, and Lorena, that helped me a lot through all the difficulties of a PhD in partnership. These years at the LIM in Milano have been very interesting, and I will always be thankful for the opportunity it offered to me.

On the French side, I would like to thank Gérard, for his unwavering support since the internship he offered me 5 years ago, and of course Philippe, who I had the luck to meet during my early years at IRCAM and who, in addition to be a precious friend, also revealed to be a precious mentor (and still did not demand the divorce after 5 years, thank you for that). I would also like to thank the marvelous ACIDS team, from the old members (as me, I'm afraid) to the newest : Léopold (we are the elders my friend), Tristan, Mathieu, Adrien, Constance, and Mikhaël, the *Representations Musicales** team Carlos, Jean-Louis, Karim, Jean. A special thought to my "lab-mate" Hugo Scurto, whose desk contiguity was a substantial contribution in this kind of post-semantic concept that is real life. To finish, I would also like to thank Sylvie, Benjamin, Jérôme, and to apologize in advance for the people I forgot.

Finally, I would also like to sincerely thank my parents, who in addition to their substantial contribution to my simple existence have brought me a critical support during all these years, my family, present and missing ones, and of course all of my friends and music/theatre fellows, that I cannot mention here, but who definitely contributed and will contribute again to the extension of my understanding of the world.

* en francais dans le texte.

Detailed Contents

Figures

Tables

Introduction and background 1

In this PhD thesis, we study the utilization of recent unsupervised learning techniques, that aroused in the machine-learning domain, to provide a novel framework for digital audio synthesis. The proposed techniques, based on the extraction of invertible features with deep representation learning, can thus be used at the same time for an analysis purpose, providing a *compressed* representation of the analyzed data, and for generation, using the extracted features as defining a *generative space* that can be explored and controlled by a human user. Contrary to most machine-learning applications developed so far in the domain of digital audio synthesis, we aim to foster an *experimental* approach (that we call *implicit*) to the use of these frameworks, rather motivated by the emerging behavior of these models than by their ability to perform external tasks. In this introduction, we will glance over the related domains broached in this work, from sound synthesis, generative systems, computational creativity, and summarizing the motivations underlying this work.

Analysis and synthesis. Sound synthesis is a seminal field of digital audio processing, that motivated the development of diverse systems focused on both human creativity and the understanding of acoustical and perceptual phenomena. While one could anachronistically describe lute-making as a first incarnation of sound synthesis, involving to understand the physical properties of the instruments to target a given sound and a specific behavior, the technical possibility to *experiment* acoustical phenomenons greatly enhanced the understanding of human perception and of audio signals, both from a musical and a scientific point of view. However, dissecting the inner mechanisms of sound needs to divide the integrity of natural phenomena in interpretable and controllable parts. This process, called *analysis* in philosophy, thus aims to extract individual concepts from an complex observation, that can be then described and experimented separately. Then, these elements of knowledge have to be gathered together whether to re-create the original phenomenon, evaluating the consistency of the analysis method, or to allow new combinations or transformations of the extracted concepts, that can be respectively be called *reasoning* or *imagination*.

In audio processing, this dichotomy between *analysis* and *synthesis* is seminal, describing accurately the reflexive process between experimentation and abstraction that is specific to natural sciences. Indeed, while the experimentation of acoustical and percpetual phenomena was already done with musical instruments through numerous practices (orchestration, acoustics, lute-making...), the possibility brought by analog instruments to manipulate even smaller components opened

significantly the field of experimentation : first, single-frequency oscillators (from analogical theremin to complex additive synthesis), noise filtering (substractive synthesis), and afterwards micro-sounds (granular synthesis), recordings (sampling), physical modeling, and many others. Hence, this *analytic* process of decomposing complex acoustical phenomena in smaller components could be whether used to propose scientific descriptions of audio signals (resp. time-frequency representations, linear predictive coding, wavelets, acoustics), or to be transformed, altered, composed, and *re-synthesized*, to allow the generation of sounds that do not exist in nature. This experimental approach to sound synthesis was seminal in the work of electro-acoustical pioneers such as Max Mathews [1], Jean-Claude Risset [2], John Chowning [3] and many others, that somehow decomposed the *real* to create the *unreal* by extending or transforming its generating rules. And, reversely, synthesis could also be used to evaluate a given analysis under a set of criteria method through simulation, hence closing the loop of a reflexive and iterative discovery process.

Generative systems and artificial intelligence. The cognitive process of detecting the existence of structuring regularity patterns of a given set of observations is fundamental in sciences, and bases the extraction of *rules* underlying the corresponding phenomenon. Among the wide variety of methodologies that exist to describe and understand reality, the difference between the *description* of a *phenomenon*, and the act of *modeling* it, is seminally different, although complementary. Indeed, while the first rather targets to find an appropriate semantic grid or a relevant description to describe a phenomenon, the second will *infer* its underlying structure using an external definition. While numerous modeling methodologies have since been developed to this end, the *generative* approach consists in developing a set of generation rules that describe the construction process of a given set of data. This approach can consist in defining a set of transformation rules, such as *generative grammars* [4–6], mathematical morphology [7], or generative approaches to music theory [8–13]. In that case, the likelihood of inferred generative rules is obtained by comparing the generated outputs to the targeted observations ; then, the model can be updated whether by changing the underlying model behind rules (*top-down approaches*) or, inversely, to extract the generative rules from the data itself (*down-top approaches*).

However, the extraction of generative patterns from the data itself is highly non-trivial. Especially, designing domain-independent algorithms for generative modeling has to rely on objective descriptions of the observations. A first approach to this purpose, called *information theory*, is a seminal signal processing framework brought by Shannon during the 50's that quantifies the amount of *information* transmitted by a given signal by analyzing its recurring patterns [14]. Fundamental notions coming from information theory, such as the concept of *information*, quantifying the amount of surprisal of a signal element from a sequence, or the concept of *entropy*, quantifying the total disorder amount of a signal. Such notions porvided the basis to an analytic

methodology to communication, and have since been extended to other domains, even in philosophy with the *philosophy of information*, that proposes to use this concept as a "*first-order phenomena represented by the world of information, computation, and the information society*" [15].
Alternatively, another generic method for structure extraction rather grounds on *statistical inference*, aiming to describe a given set of observations using density estimation techniques based on probabilities. This methods can be based on an explicit probabilistic parametric model, whose parameters can be optimized using criteria (such as data likelihood), and then describe the observed data with the corresponding parameters, at the cost of dropping the exact reversibility of the model. Advanced statistical inference methods, such as Bayesian or latent models, allow more the definition of more complex inference systems that can be used to extract *latent representations* of a given dataset, brdiging with representation learning and information theory [16, 17].
Despite thier differences, these two approaches are related by their common based of *compression* , as they are both based on the extraction of regularity patterns underlying a target process [18]. While these patterns can have different names or formulations depending on the domain (*syntax* in generative grammar, *code* in information theory, *model* in inferential statistics), they all aim to reduce the complexity of a target data by extracting invariant structures, based on non domain-specific properties of the data. This underlying relation between compression and complexity motivated criteria for model selection : based on the idea of *Kolmogorov complexity*, that defines the complexity of a signal by the bit length of the smaller code that can generate it, methods such as *Minimum Description Length* use this generative criterion to select the one that achieves the best compression, as an information-theoretic version of Occam's razor [19].

This intrinsic correspondence between compressibility and explainability is also underlying *representation learning* methods, that are based on the representation of a set of high-dimensional data in a smaller space. While such methods can be beneficially leveraged for visualization, they also have a seminal interest in *machine-learning*, a domain based on the automatic execution of a given set of tasks. This data-centered approach to automatic feature extraction recently raised a significant interest in the scientific community, was caused not only by the recent increase of computation power allowed by parallel computing, enabling the use of connectionist approaches (grounded by cybernetics) such as neural networks that enhanced the modeling abilities of such methods, but also by the ever growing amount of available data. While the domain of machine-learning is rather focused on the extraction of optimal representations for a given set of *external* tasks, the recent merge of *unsupervised* approaches with machine-learning methods allowed to leverage these modern techniques for learning compressed representations of a given set of data. Such approaches, mostly grounded on manifold hypothesis [20], assert a given set of high-dimensional data to lie on a sub-manifold of the embedding space. These methods are then aiming to the extract this manifold to

recover the underlying generative process, and are then related to the compression-based systems explained above, and bridging representation learning and generative models [21]. Moreover, recent hybridizations between generative and machine-learning models gave birth to a flourishing class of expressive density estimation methods, enabling the development of powerful data-centered approaches to generative models (variational auto-encoders [22], generative adversarial nets [23], and recently generative normalizing flows [24]).

Computational creativity, autonomy and empowerment. While these recent advances in the domain of data-centered approaches for data modeling engendered powerful systems for numerous applications, can generative models be called *creative*? Indeed, the recent raise of interest in machine-learning was intrinsically driven by the growing interest in *autonomous* systems, that in this case are able to learn a given set of tasks. However, as generative models are based on the reconstruction of the original data, their *creative* aspect may be tempered.
The seminal works of Margaret Boden [25] are enlightening on this point, grounding an accurate thinking about creativity and machines. Indeed, the distinction between *P-creativity*, defining objects that are only unknown to the generating agent, and *H-creativity*, defining objects things are *historically* new, emphasizes the fact that the *creative* dimension of an artifact is in part (if not totally) driven by the context. Furthermore, Boden also underlines the co-existence of several *modalities* of creation, distinguishing *exploratory* creativity, that is the discovery of new objects by navigating an existing *generative space*, or the *transformative* creativity, that is the creation of new objects by the transformation of existing ones. *Combinatorial* creativity, added afterwards by Pasquier [26], also underlines the distinction between transforming and composing objects.

Modern machine-learning models that have been developed to directly address creativity are rather focused on *autonomous* systems, whose emerging behavior is generally driven or learned with a set of *reward signals*. These models, based on reinforcement or active learning, update their internal state by observing a suitable reward signal, whether *internal* and based on an the autonomous analysis on a given creativity measure, or *external*, provided by the external world (interaction with human, active observation with receptors) [18, 27, 28].
However, how to evaluate the creative aspects of non-autonomous generative models, such as the ones investigated in this thesis? Indeed, the creative constraints of machine-learning based methods for generation seems rather driven by another paradox unveiled by Boden (also addressed by Boulez in his fundamental essay *A la limite du pays fertile*) : "*The more stylistically fundamental the altered constraint, the more surprising—even shocking— the new ideas will be.*". Indeed, the direct generation of data judged "realistic" or at least "understandable" for a human is a very complex task, that is not eased with the computational power of these methods. Indeed, modern generative models are rather focused on their ability to reconstruct the original data, in other terms the performance obtained on their training task. However, at which

extent can they be used for creation? On this point, the role of *representation* seems mandatory : allowing external *interaction* with the model (by the human or another model) implies to get a understandable description of the system's behavior. Hence, using the Boden modalities summarized above, what are the possibilities offered to *explore* the generative abilities of the model, and how can they *transform* or *compose* inner properties of the data?

Towards an experimental approach to machine-learning Audio synthesis methods, that on these creative aspects are not different from generative models, alleviated this *creative* evaluation with the close relationship it maintained with creation and production, being directly embedded in musical practices since its creation. However, the utilization made so far of machine-learning techniques in audio processing domain are still what we could call *extrinsic*, i.e. constrained to the correct execution of a given set of external tasks.
While the integration of machine-learning based generation models starts to establish in the domain of image creation with well-known systems like DeepDream, GANs, or the recently released Runway platform [*], there is no similar trend in audio synthesis (apart from works of Magenta[†], interestingly focusing on musical creative flows [29, 30]). Indeed, machine-learning generative models for music are so far constrained in the symbolic domain, targeting to model musical styles such as AIVA [‡]. In the signal domain, recent commercial software propose to introduce machine learning in mixing or mastering to target specific aesthetics, such that Lindr [§] or iZotope Neutron [¶]. However, these algorithms are still leveraging machine-learning to *reproduce* existing structures in the musical domain, and do not propose an explicit relation to the *experimental* approach that is yet seminal in musical creation. However, these models can also be used from an *instrisic* manner, aiming to really explore the inner behaviour of this new methods and proposing a new framework for digital synthesis, involving their specificities (complexity, learning, modularity, stochasticity) in a real creative process. Such approaches could then address

- *transformational* creativity, allowing to transform external data with the system or transforming the system itself
- *exploratory* creativity, providing an browsable representation whose design is involved the creative process
- *crafting*, involving the definition of model and of the desired properties of the representation into the creative process

aiming to find this combination of mastering and surprise that make audio synthesis methods really *creative* and *enjoyable*. However, to challenge this objective, we had to tackle several scientific questions such that the *representativeness* of these extracted synthesis space, that led us to the development of regularization strategies for focused on

[*] https://runwayml.com/
[†] https://magenta.tensorflow.org/
[‡] aiva.ai
[§] https://www.landr.com/fr/
[¶] https://www.izotope.com/en/products/neutron.html

audio signals, as well as the introduction of their temporal dimension in the used frameworks. Then, as this approach also aims to integrate the developed framework into the creative flow of a human user, expert or non-expert, we experimented our framework by proposing a novel musical piece based on this system, *ægo*, that provided a first step towards a *research and creation* approach. We know that this work is still preparatory, but we hope to propose an accurate way of involving these systems into music creation with an experimental perspective, rather than just as way of mocking existing content. Good reading!

Generative models, representation learning and variational inference

2

In this work we investigate the use of a recent trend of generative models, based on advanced machine learning techniques for data modeling, as a novel method for sound synthesis. Such approaches could then be complementing the existing sound generation methods, based on *data-centered* processes that extract generative patterns directly from existing data. The recent improvements of these systems allowed the development of powerful modeling techniques, enhanced by the use of high-capacity function approximators that increased their *expressiveness*. However, the direct use of these methods for creative applications is still an on-going problem, mainly due to the lack of ways to *control* their generation, hence limiting their operability as artistic tools. Hence, the use of these techniques as interactive audio generative models also implies to both model the targeted data and obtain a suitable representation, called a *generative space*, that the user could use to control the generated output. Such models could then be thought as mirroring *analysis-synthesis* methods in the Digital Signal Processing (DSP), and then based on the extraction of high-level features from the data (analysis) that could then be inverted in the data domain (synthesis).

However, using data-centered approaches for the extraction of these generative spaces is a non-trivial task, and can then be obtained through different approaches. A first way to obtain representations from a given data is to leverage probabilities, aiming to extract meaningful descriptors by analyzing the global distribution of the observed samples. This method can be whether based on statistical descriptors of the data, retrieving indicators about the global distribution of the data sample (descriptive statistics), or based on its *modeling* using a defined *generative distribution* $p(\mathbf{x})$ (then called *statistical inference*). While the invertibility of the first approach is non-trivial, making it irrelevant for generative purposes, modeling a given set of data as samples drawn from $p(\mathbf{x})$ allows to obtain new examples from the extracted structure.
Furthermore, *Bayesian* approaches to statistical inference allows to model both the distribution of the data and of the model parameters (see sec. 2.1), and even conditioning the generative distribution with an exterior set of random variables, called *latent variables* (see sec. 2.1.2). Such statistical inference methods allow to retrieve interpretable features from the data, as the obtained parameters and latent variables have an explicit definition in the corresponding model, and can be both used for generation and inferred from the data. However, such methods generally impose strong tractability assumptions, limiting the complexity and then the expressiveness of the involved models, requiring approximation schemes to extend their modeling capacities (see sec. 2.1.3).

Alternatively, some methods drop the probabilistic assumption of the data distribution and rather analyze geometric or topological properties from the dataset, providing low-dimensional representations (see sec. 2.2.1) that aim to reflect information about the data structure. Differently, some methods leverage the use of high-capacity function approximators such as neural or convolutional networks (see sec. 2.2.2) to extract representation from the data, whether based on an external task that enforce a given structure to the representation (such as classification or regression), hence called *supervised learning*, or trained to act as a compressed space representing the data, hence called *unsupervised learning* (see sec. 2.2.3). While the first can be very efficient to obtain representations that can be used for feature extraction, the high-capacity of the involved feature extractors generally lead to degenerated representations, that overemphasize the target task at the cost of not extracting any structural information about the data. Furthermore, supervised methods assume to have the desired outputs for every sample of the dataset, that can be an important limitation for the creative applicability of our method. Conversely, unsupervised learning is appealing because of its independence to externally defined tasks ; however, the capacity of the involved function often prevent the extracted representations to represent interpretable generative factors of the input data.

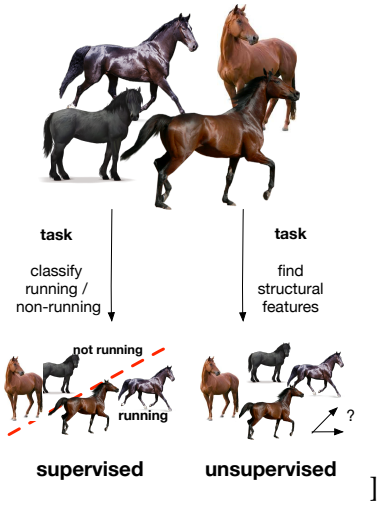


Figure 2.1: *Supervised* approaches vs. *unsupervised* approaches.

Going back to our case, what would be the requirements for such extracted representations in order to be used as a control space for sound synthesis? ALl the different representations properties described above could find its place in a framework, especially if the definition of the model should make part of the creative process. Indeed, it seems mandatory that the extracted control space should be low-dimensional and represent structural properties of the data, otherwise the exploration of the provided representation would be insensitive and discouraging for the human user. As we described previously, Bayesian latent models allow to obtain sensitive and interpretable control parameters, but unfortunately their limited modeling capacity often yield to a weak expressiveness, that could provoke a diminished interactive interest. Reversely, representations obtained with high-capacity function approximators such that neural networks got the target expressiveness between the control and data space, but are often degenerated and high-dimensional. Hence, the ideal model for our purpose would be an hybridization of both, allowing the ensure the consistency of the extracted representation but that would be also based on expressive inference and generation processes. Furthemore, the introduction of explicitly defined tasks could be also interesting, allowing the representation to represent user-defined factor of variations, but still reflecting structural properties of the data. The desired framework should then be flexible enough to allow all of these criteria, but also be light enough to be easily trained by the human user.
In the last section of this chapter, will described the chosen framework, called *Auto-Encoding Variational Bayes* (AEVB), that bridges Bayesian inference processes and deep representation learning. This framework, that can be understood as an hybridation between *auto-encoders* (see sec. 2.2.3) and Bayesian latent models, advantageously leverages the

use of neural networks to model expressive relationships between the data space and the higher-order features. This flexible framework, proposed simultaneously by Kingma & Welling [31] and Rezende & al. [32], raised a significant interest in the machine-learning community, and since gave birth to many developments and further analysis, notably on theoretical and optimization perspectives.
Hence, in this chapter, we will first provide a state of the art of latent Bayesian models focusing on an approximation scheme, called *variational learning*, that allows to drop the strong tractability assumptions of direct Bayesian inference. In the next section, we will rather focus on non-probabilistic systems based on the extraction of representations from the data, then called *representation learning*, by describing dimensionality reduction methods, neural networks properties and optimization, and neural-network based generative models. Finally, we will present an extensive state of the art of the used framework, that leverage techniques of these two domains, summarizing the emerging properties of these systems and various developments that can be advantageously used for the extraction of interactive generative spaces.

2.1 Bayesian probabilistic generative models

As we approached in the introduction, a *generative model* is a system that can output a finite or infinite set of realizations, whose generation process is defined by a set of rules (physical, mathematical, logical) that define the *identity* of the corresponding model. Hence, extracting a generative model from a finite set of samples comes back to recovering the underlying generation rules of the corresponding system, resorting to various identification methods. While some identification methods assume strong properties about the structure of the generative model, developing general methods for model identification resorting the minimum set of assumptions is a challenging task. A method to address this challenge are *statistics*, resorting to probabilistic models to describe, or model, a restricted set of data.
Though, there exists different trends in statistics, differing in how they extract these information, leading to different philosophies (see fig. 2.2)). *Descriptive statistics* focuses on how to accurately describe a sample of data, and aims to formulate *statistical descriptors* of the properties of the sample (distribution, covariance, dispersion...), without adding any further assumption on the data structure. While this framework provides efficient methods for data analysis and representation, the absence of an underlying probabilistic model prevents to generate new content from the observed data, and does not infer any explaining factor that would describe the generative process. Conversely, *inferential statistics* rather consider the observed data $\mathbf{x}$ as a set of *realizations*, that are drawn from a probability distribution $p(\mathbf{x})$. Hence, inferential statistics aims to recover this underlying distribution, then recovering the inner structure behind the observed data. Furthermore, if we are able to sample the inferred distribution $p(\mathbf{x})$, we can then generative new data that matches the extracted structure, then proposing a probabilis-

tic method for *generative modeling*. However, extracting the underlying distribution from a limited amount of data is not a trivial task, as the exact extraction of the underlying probabilistic model would require an infinite amount of observations.

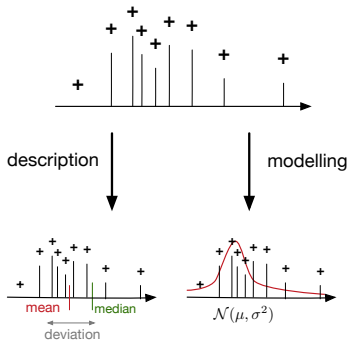


Figure 2.2: *Descriptive* and *inferential* statistics

Therefore, several strategies can be used to address this challenge, that are quite philosophically different even if based on the same mathematical framework. The first strategy is to choose a *frequentist* approach, that defines the probability of an event by the number of its occurrences within the observations. The underlying concept is that the most probable events of a random process are the ones that happens the most, then assuming that the process is composed by a lot of reproducible and independent trials. The frequentist approach then models the probability distribution underlying a set of observations by accurately modeling its events' frequency. This idea, based on the seminal *Law of Large Numbers*, ensures the convergence of the obtained estimation as the number of observed events goes to $+\infty$. However, this approach raises several difficulties : first, the number of samples needed to accurately model the target distribution can be very high, and is not able to model probabilities of rare events. Another problem is that this approach does not model the uncertainty of the model, such that it may be *over-fitting* the distribution, and performs badly in case of noisy distributions.

2.1.1 Bayesian learning for statistical inference

Bayesian inference. The Bayesian philosophy of probabilities is then rather different from its related frequentist, detaching probability from the idea of repetition to translate it in the domain of **believes**, modeling the certainty of a given assumption after some observations. While the first were targeting to uncover the underlying distribution $p(\mathbf{x})$ without any further assumption on the observed data, Bayesian learning models a *prior* knowledge on the process, that is progressively modified as the number of observed data items increase.

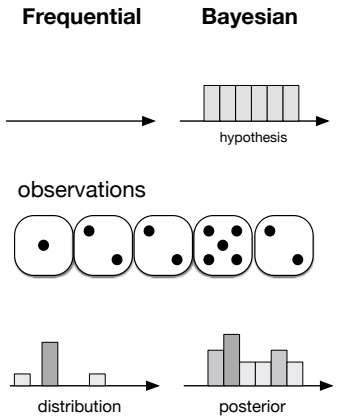


Figure 2.3: Difference between frequentist and Bayesian observation processes.

Let be an hypothesis H on an experiment realizing observations O, that can be observed by the emitter. Given an *a-priori* probability distribution $p(H)$ defined on the space of hypothesis, the observer will then expect a distribution $p(O|H)$ over the observations O, given the prior hypothesis H. However, after an observation O, the emitter will refine its hypothesis accordingly to $p(H|O)$, then modifying its prior assumptions. We can see that this formulation differs from the previous one by the introduction of a *prior* knowledge on the observation, that will be progressively biased by the observed realizations obtained from the experiment O. Furthermore, the hypothesis H will not be modified to much if the observation O was already considered as probable, while an unexpected observation will incite the *posterior* $p(H|O)$ to deviate the original hypothesis $p(H)$ more substantially. This trade-off can be expressed with the Bayes' rule, that can be easily obtained from the conditional decomposition of the joint distribution $p(O, H)$

$$p(O,H) = p(O|H)p(H) \quad = p(H|O)p(O) \underbrace{p(H|O)}_{\text{posterior}} = \frac{p(H,O)}{p(O)} = \frac{\overbrace{p(O|H)}^{\text{likelihood}} \overbrace{p(H)}^{\text{prior}}}{\underbrace{p(O)}_{\text{marginal likelihood}}}$$

The Bayes' equation thus identify four terms : the *posterior* distribution $p(H|O)$ of the hypothesis H after some observations O, the prior knowledge $p(H)$, the *likelihood* $p(O|H)$ of the observation O under the hypothesis H, and the probability of observation $p(O)$. Bayesian statistics are then more robust under small variance noise over O, as the posterior distribution $p(H|O)$ won't be modified if the probability of the observation $p(O)$ corresponds to the likelihood $p(O|H)$. Furthermore, this Bayesian formulation allows us to define a probability distribution over H rather than a point-estimate, such that the posterior process also gives information about the *uncertainty* of the prediction.

Likelihood estimation. In statistical inference, we target to recover the generative distribution $p^\star(\mathbf{x})$ underlying some finite observations $\{\mathbf{x}_i\}_{i=1\dots N}$. As the true distribution is unknown to us, statistical inference relies on modeling this distribution with a defined model $p(\mathbf{x})$ that we will force to match with the observations. These methods can be whether *non-parametric*, using descriptive statistics estimators to model the distribution, or *parametric*, such that the model $p(\mathbf{x};\boldsymbol{\theta})$ depends on a set of parameters $\boldsymbol{\theta}$. While the first can perform well on simple data, the second allows us to define a precise structure for the distribution $p(\mathbf{x})$, then enforcing some properties of the distribution. However, estimating the parameters $\boldsymbol{\theta}$ of a model $p(\mathbf{x};\boldsymbol{\theta})$ is not straightforward, and requires an explicit criterion to optimize. The most common approach to maximize the likelihood of the data given the model p_θ, optimizing the parameters $\boldsymbol{\theta}$ of the model [33]

$$\boldsymbol{\theta}^\star = \max_{\boldsymbol{\theta}} p(\mathbf{x};\boldsymbol{\theta}) \tag{2.1}$$

In some cases this optimization is tractable, such that the optimal parameters $\theta^\star$ can be directly obtained from the data. Furthermore, maximum likelihood has been shown to be a *consistent* (converging in probability to the real density $p(\mathbf{x})$), and *efficient* (having the smallest variance) estimator when the sample size tends to infinity. However, tractable models for log-likelihood maximization are often not expressive enough to shape the underlying complexity of the data, such that more elaborated schemes have to be used such as gradient descent algorithms (see sec. 2.3.1), normalizing flows (see sec. 2.3.2.1) or contrastive estimation (see sec. 4.3.1) to give an estimate of the parameters $\boldsymbol{\theta}$. Alternatively, we can turn statistical inference into Bayesian problem, considering parameters $\boldsymbol{\theta}$ themselves as random variables. In that case, the hypothesis H are represented by the parameters $\boldsymbol{\theta}$ with a corresponding prior distribution $p(\boldsymbol{\theta})$, such that Bayes' equation can

be written

$$p(\boldsymbol{\theta}|\mathbf{z}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})} \tag{2.2}$$

$$= \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{x}|\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{2.3}$$

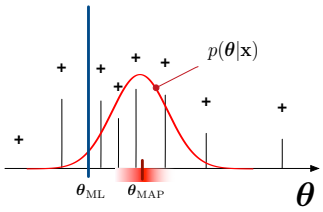


Figure 2.4: Maximum Likelihood estimation (blue) and Maximum A Posteriori estimation (red). ML gives a point-estimate of the mean, while MAP finds the maximum over a distribution of possible means.

providing a posterior distribution $p(\boldsymbol{\theta}|\mathbf{x})$ over the parameters. Modeling the estimation of $\boldsymbol{\theta}$ with a probability distribution rather than a *point-estimate* allows us to access the model *uncertainty*, from which we can get an estimate by whether selecting the $\boldsymbol{\theta}$ with highest probability, then called the *maximum a-posteriori* estimation (MAP), or to sample $\boldsymbol{\theta}l$ over a given confidence range of $p(\boldsymbol{\theta}|\mathbf{z})$. Setting a uniform prior for θ recovers the ML estimator of 2.1 up to the $p(\boldsymbol{x})$ term, that can be excluded from the optimization as it is independent from θ.

We can now see that Bayesian learning gains from frequentist approaches by formulating an a-priori knowledge on the parameters' distribution, that can be thought as a soft regularizer over the parameter estimation. The definition of prior distribution $p(\boldsymbol{\theta})$ can whether be chosen non-informative (uniform or isotropic normal) or informative, in the case of available external knowledge or iterative model updating. Furthermore, this method also allows us two different methods to *predict* new values $\hat{\mathbf{x}}$ from the inferred model

$$p_{post}(\hat{\boldsymbol{x}}|\boldsymbol{x}) = \int p(\hat{\boldsymbol{x}}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{X})d\boldsymbol{\theta} \tag{2.4}$$

$$p_{prior}(\hat{\boldsymbol{x}}) = \int p(\hat{\boldsymbol{x}}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} \tag{2.5}$$

$$\tag{2.6}$$

The first prediction method, called *posterior predictive distribution*, predict new values for $\mathbf{x}$ marginalizing out the parameters $\boldsymbol{\theta}$ using the posterior distribution while the second, called the *prior predictive distribution*, rater marginalizes the parameters using the prior information. The regularization effect brought by the probabilistic distributions $p(\boldsymbol{\theta}|\boldsymbol{X})$ and $p(\boldsymbol{\theta})$ can thus be seen by their marginalization over $\boldsymbol{\theta}$, meaning that all the possible parameters values modeled by these distributions are involved in the prediction of $\hat{\mathbf{x}}$. Bayesian learning is then more robust than its point-estimate counterpart, that would only take one possible $\boldsymbol{\theta}$ to perform the prediction.

Conjugate priors. Modeling the posterior distribution $p(\boldsymbol{\theta}|\mathbf{x})$ rather than obtaining point-estimate values $\boldsymbol{\theta}$, has then several advantages : also modeling the uncertainty about the parameters estimation, modeling both parameters' inference and data generation, and data prediction. However, computing these distributions assumes that all of the involved terms from 2.2 are tractable, even the marginal likelihood $p(\mathbf{x})$ that involves the marginalization of the likelihood $p(\hat{\boldsymbol{x}}|\boldsymbol{\theta})$ over $\boldsymbol{\theta}$. The full tractability of this expression can be ensured by formulating

a generative distribution and its corresponding prior with similar algebraic forms, i.e. taking a *conjugate prior* as initial distribution $p(\mathbf{z})$. If the two distributions are conjugate, the posterior distribution will then have a similar algebraic form, coming back to an update of the prior parameters after the observation of data samples [34]. The simplest example is the multivariate normal distribution, that is a self-conjugate distribution ; indeed, taking a likelihood model $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with a fixed covariance matrix $\boldsymbol{\Sigma}$, we can also model the prior of the parameter $\boldsymbol{\mu}$ with a normal distribution, such the corresponding prior and the posterior distributions can be written

$$p(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) = \mathcal{N}\big(p(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\big) \tag{2.7}$$

$$p(\boldsymbol{\mu}_0|\boldsymbol{\mu}) = \mathcal{N}\Big((\boldsymbol{\Sigma}_0^{-1} + n\boldsymbol{\Sigma}^{-1})^{-1}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + n\boldsymbol{\Sigma}\bar{\mathbf{x}}), \tag{2.8}$$

$$(\boldsymbol{\Sigma}_0^{-1} + n\boldsymbol{\Sigma}^{-1})^{-1}\Big) \tag{2.9}$$

where $\bar{\mathbf{x}}$ is the sample mean of the n data $\mathbf{x}_1 ... \mathbf{x}_n$. Intuitively, the posterior of the mean can the been understood as the *precision weighting* of the prior mean $\boldsymbol{\mu}_0$ and the empirical mean $\bar{\mathbf{x}}$, whose precision is the sum of both prior and observations distributions. This simple example shows how the use of conjugate priors can provide tractable posteriors with interpretable parameters. Prior distributions are available for many widely used distributions such as Bernoulli, Dirichlet, or Gamma functions, and then provide an efficient way to perform Bayesian inference of model parameters. Furthermore, using conjugate distributions allows us to perform whether *offline*, having all the observed data in a dataset $\mathbf{x}$, or *online*, where we successively update the prior information based on the previously observed data (see fig 2.5).

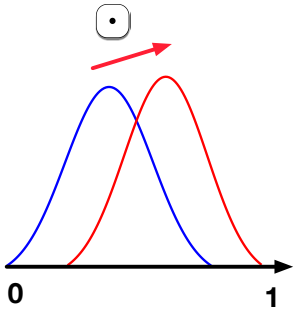


Figure 2.5: Example : a Beta(α,β) prior equally distributed between 0 and 1 shots (blue). Conjugate posterior after one more observation Beta(α,$\beta+1$). As beta is a natural mean conjugate of Bernoulli distribution, after one new "1" observation the distribution is a new Beta, with $\beta \leftarrow \beta + 1$

2.1.2 Latent models and posterior approximation methods.

Latent variables and graphs. Bayesian learning hence allows a robust and elegant way to learn generative models, gaining on likelihood-based methods by inferring posterior distributions over the model parameters. Despite the elegance of the conjugate prior approach, it significantly constrains the expressiveness of the generative model $p_\theta(\mathbf{x})$, enforcing the involved distributions to have a similar shape. Furthermore, conjugate priors can be only used to get posterior distributions for the parameters of the generative model, and can not be used for additional random variables that would increase the model expressiveness.

An alternative approach for statistical inference is to condition the observations $\mathbf{x}$ on auxiliary variables $\mathbf{z}$, then called *latent*, that are controlling the generation process of the underlying data distribution (see fig. 2.6). In this case we maximize the joint distribution $p_\theta(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})$ instead of $p_\theta(\mathbf{x}; \boldsymbol{\theta})$. Introducing latent variables to parameters model

can whether be used to ease the probabilistic formulation of a problem, such that the overall model $p(\mathbf{x}, \mathbf{z})$ is more easy to sample, or to model the influence of an external factor, observable or hidden, on the data $\mathbf{x}$. It also provides a way to *control* the generation of $p(\mathbf{x}$, by choosing a latent variable $\mathbf{z}$ and retrieving corresponding $\mathbf{x}$ samples from $p(\mathbf{x}|\mathbf{z})$. Latent variables is then a very convenient tool in statistical inference, and is now present in most machine learning methods. Similarly, latent variables can be inferred from a given generative distribution $p_\theta(\mathbf{x})$ with Bayes' rule

$$p(\boldsymbol{\theta}|\mathbf{z}) = \frac{p_\theta(\boldsymbol{x}|z)p(z)}{\int p_\theta(\boldsymbol{x}, z)dz} \tag{2.10}$$

where we explicitly condition the generative process on $\mathbf{z}$, θ being the parameters of the conditional distribution. Bayesian learning is though still probable, providing the tractability of the conditional generation $p_\theta(\boldsymbol{x}|z$. While this formulation increases the model expressiveness model, it also makes the integral $\int p_\theta(\boldsymbol{x}, z)dz$ intractable, enforcing whether to have *discrete* latent variable (so the integral becomes a summation $\sum_k p_\theta(\boldsymbol{x}, z_k)$), or to leverage optimization schemes in order to maximize the overall probability $p_\theta(\mathbf{x}, \mathbf{z})$. While the first can be very useful for some problems, such as mixture of distributions, having continuous latent variables $\mathbf{z} \in \mathcal{R}$ could allow us to extract *hidden factors* from the data, that could be used to retrieve generative factors from the data. Inference of continuous latent variables can be achieved with optimization methods, such as *expectation-maximization* (EM) or *variational inference* (VI), that we will discuss in the next section.

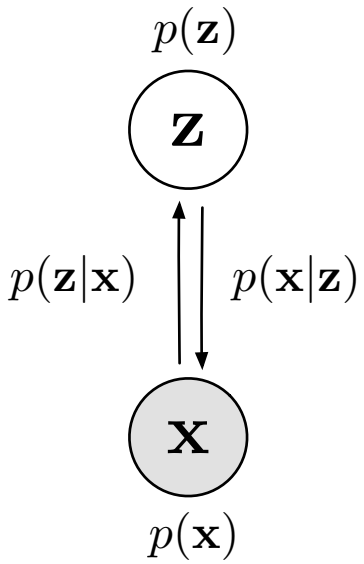


Figure 2.6: Conditional graph between observed variables ($\mathbf{x}$, in gray) and unobserved variables ($\mathbf{z}$, in gray)

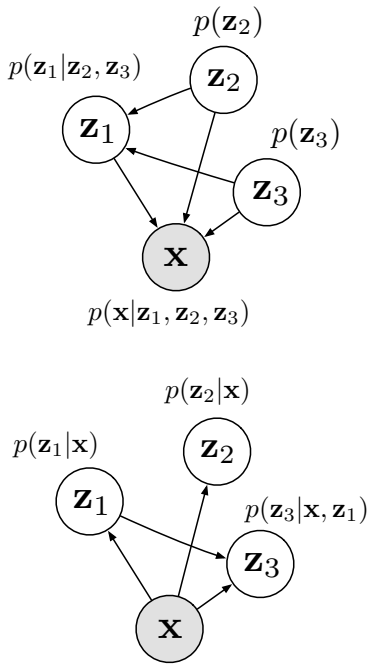


Figure 2.7: Generation (top) and inference (down) probabilistic graphs of a joint distribution $p(\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)$

Extending this idea to multiple variables $[\mathbf{x}_1, ...\mathbf{x}_n, \mathbf{z}_1..., \mathbf{z}_m]$, composed by n observed variables and m latent variables, we can define complex probabilistic structures by formulating group dependencies between observed and latent variables, as can be seen fig. 2.7 with the generative model $p(\mathbf{x}_2|\mathbf{z}_1, \mathbf{z_2})p(\mathbf{x}_1|\mathbf{x}_1, \mathbf{x}_2)$. Such conditioned structures define an underlying directed graph, such that these methods are called *probabilistic directed graphical models*. Such structure are widely used in the machine-learning field, as it allows to define intuitively model relationships between groups of observed variables that can be explained by one or several common latent factors. A first occurrence of these models can be found *Bayesian belief networks*, that are directed acyclic graphs modeling conditional dependencies between adjacent nodes. The overall probability of the observed nodes $\mathbf{x}_n$ conditioned on its *parent nodes* ($\text{pa}(\mathbf{x}_n)$, the set of nodes conditioning the node $\mathbf{x}_n$)is then [35–37]

$$p(\mathbf{x}) = \prod_{i=1}^{N} p(\mathbf{x}_i|\text{pa}[\mathbf{x}_i])$$

such that the system verifies the *Markov Blanket Property*, assuming than a node is only conditioned by their direct neighbors. Bayesian belief networks can then be easily trained with the Bayes' rule 2.2, as the marginal likelihood of discrete latent variables can be easily computed with conditional probability tables. More generally, infer-

ence in complex graphs can be possible by modeling a chain over the conditional probabilities using the graph structure, and then to train with maximum likelihood or Bayesian inference on the observed nodes. However, defining directed probabilistic graphs can be a too strong assumption on the model, as it explicitly defines the conditioning structure between the variables. Alternative approaches are rather based on *undirected graphical methods*, where the total probability of a set of connected nodes $\mathbf{x}_k$ is expressed as

$$p(\mathbf{x}) = \frac{1}{Z} \exp \underbrace{\Big(\sum_k \sum_{l \in \mathcal{N}(k)} w_{k,l} \cdot f_{k,l}(x_l) \Big)}_{-\mathcal{E}(\mathbf{x})}$$

with feature functions f_k, $\mathcal{N}_k$ denotes the neighborhood of a node k, and $w_{k,l}$ is the weight of the relation between the nodes x_k and x_l. These models, also called Markov Random Fields, are then based on the minimization of the energy $\mathcal{E}$, amounting to maximizing the overall probability of the model (see fig. 2.8). The undirectivity of the graph requires to estimate the partition function Z, such that the overall distribution $p(\mathbf{x})$ sums to 1. However, computing the partition function Z is a tedious task, forcing whether to constrain the graph architecture (such as Restricted Boltzmann Machines[38]) or to approximate the partition function using approximation schemes[39, 40]. However, the duality between directed and undirected graph motivated *message passing* approaches to Bayesian graphical models such as the *sum-product* algorithm, that uses the factor decomposition of a graph to efficiently model the free energy of adjacent nodes [41].

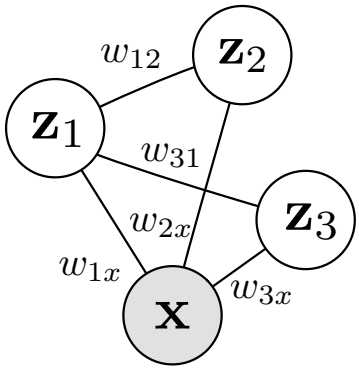


Figure 2.8: Example of Markov random field, where each node is linked by a weight w_{ij}. No direction is assumed, and the probability of the all system is maximized in turn.

Expectation-Maximization Unfortunately, the graph-based methods quickly presented above also assume the overall tractability of the model, therefore constraining the possible dependencies between involved random variables. Maximizing the model likelihood $p_\theta(\mathbf{x})$ without computing the integral term in 2.10 then involve optimization schemes, that replaces the direct computation of the posterior by iterative model updates that are perfomed until convergence of the model. A common optimization scheme, called *Expectation-Maximization* (EM), is a framework that proposes an iterative scheme both parameters and latent variables of a model $p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})$ provided the tractability of $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})$ [42]. Expectation-Maximization is based on a two-step procedure

- an *expectation step*, estimating the expectation $\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta})}[p(\boldsymbol{z}, \boldsymbol{x}|\boldsymbol{\theta})]$ over the latent variables $\mathbf{z}$
- an *maximization step*, that optimizes the parameters $\boldsymbol{\theta}$ to maximize the first expectation.

The in-turn application of these two steps models an implicit lower-bound of the data likelihood $p_\theta(\mathbf{x})$, that converges to a local maximum of $p_\theta(\mathbf{x}, \mathbf{z})$. The intuition behind EM optimization if that maximizing the joint probability $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})$ over the the distribution $p(\boldsymbol{z}, \boldsymbol{x}|\boldsymbol{\theta})$ amounts to optimize the data log-likelihood, without having to calculate the marginal in 2.10. An alternate view, proposed by Neal &

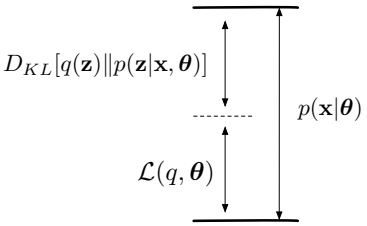


Figure 2.9: Decomposition of log-likelihood $p(\mathbf{x}|\boldsymbol{\theta})$ with any distribution $q(\mathbf{z})$. Increasing $\mathscr{L}(q,\boldsymbol{\theta})$ reduces the divergence $D_{KL}[q(\mathbf{z}) \parallel p(\mathbf{z}|\boldsymbol{x},\boldsymbol{\theta})]$, such that $q(\mathbf{z}) = p(\mathbf{z}|\boldsymbol{x},\boldsymbol{\theta})$

Hinton [43], formulates E-M optimization as an example of *coordinate ascent* of two estimators. Taking an arbitrary function $q(\mathbf{z})$, we can derive

$$\begin{aligned}
\log p(\boldsymbol{x}|\boldsymbol{\theta}) &= \log p(\boldsymbol{x}|\boldsymbol{\theta})\Big[\int q(z)dz\Big] \\
&= \int q(z)\log p(\boldsymbol{x}|\boldsymbol{\theta})dz \\
&= \int q(z)\log \frac{p(\boldsymbol{x},z|\boldsymbol{\theta})}{p(z|\boldsymbol{\theta})}dz \\
&= \int q(z)\log \left\{\frac{p(\boldsymbol{x},z|\boldsymbol{\theta})}{q(z)}\cdot\frac{q(z)}{p(z|\boldsymbol{\theta})}\right\}dz \\
&= \int q(z)\log \frac{p(\boldsymbol{x},z|\boldsymbol{\theta})}{q(z)}dz - \int q(z)\log\frac{p(z|\boldsymbol{\theta})}{q(z)}dz \\
&= \mathscr{L}(q,\boldsymbol{\theta}) + D_{KL}\big[q(z) \parallel p(z|\boldsymbol{x},\boldsymbol{\theta})\big]
\end{aligned} \tag{2.11}$$

showing the intrinsic decomposition provided by E-M between a lower-bound $\mathscr{L}(q)$, quantifying the expected density ratio between the joint distribution $p(\boldsymbol{x},z|\boldsymbol{\theta})$ and the arbitrary $q(\mathbf{z})$, and a KL-divergence between the distribution $q(z)$ and the posterior distribution $p(z|\boldsymbol{x},\boldsymbol{\theta})$ (see fig. 2.9). Thus, during the expectation step, $\mathscr{L}(q,\theta)$ is optimized with respect to q, updating the current model distribution modeled on $\mathbf{z}$. Conversely, the maximization step updates the bound $\mathscr{L}(q,\boldsymbol{\theta})$ respectively to $\boldsymbol{\theta}$, also increasing $\mathscr{L}(q,\theta)$ as both of these terms are positive. Hence, optimizing $\mathscr{L}(q,\boldsymbol{\theta})$ reduces the divergence $D_{KL}\big[q(z) \parallel p(z|\boldsymbol{x},\boldsymbol{\theta})\big]$, then matching $q(z)$ and $p(z|\boldsymbol{x},\boldsymbol{\theta})$ at convergence. EM demonstrates how we can avoid to compute the marginal log-likelihood of 2.10, while still optimizing an implicit distribution to match the posterior. This intuition provides the basis of *variational inference*, which is the core of this PhD.

2.1.3 Variational methods for posterior approximation

2.1.3.1 Mean-field variational inference

The investigation of Expectation-Maximization thus showed that the full tractability of Bayes' rule (2.10) was not necessary to perform inference on latent variables, replacing the exact computation of the posterior by a maximization problem. The relaxation of this hypothesis thus enables the formulation of a wider class of generative distributions, at the cost of maybe not retrieving the exact posterior. While the performances achieved by EM are satisfactory, this method requires the tractability of the posterior $p(z|\boldsymbol{x},\boldsymbol{\theta}$, preventing to use more complex generative models.
Variational methods alleviate this problem by rather modeling approximate posteriors $q(\mathbf{z})$, relaxing the need of a tractable posterior $p(\mathbf{z}|\mathbf{x})$. The tractability of variational methods is achieved by decomposing the approximate distribution among latent variables, allowing us to

optimize each latent variables independently [44–48]. Variational inference is based on the same decomposition than E-M (2.11), but rather *explicitly* defines the latent distribution $q(\mathbf{z}) \in \mathbb{Q}$ in a family of model $\mathbb{Q}$. A common choice for variational posteriors is the *mean-field decomposition*, that enforces the independence between each latent variables and every latent dimension [49]

$$q(\mathbf{z}) = \prod_{i=1}^{d} q(z_i) \tag{2.12}$$

in that case, the intractable calculation of the true posterior (2.10) is replaced by the in-turn optimization of each posterior component $q(z_i)$

$$q_j(z_j) = \frac{\exp^{\mathbb{E}_{ij}[\log p(\mathbf{z},\mathbf{x})]}}{\int \exp^{\mathbb{E}_{ij}[\log p(\mathbf{z},\mathbf{x})]} dz_j} \tag{2.13}$$

that is tractable, provided the conjugacy between $q_j(z_j)$ and its respective prior parameters. The approximated posterior can then be retrieved by introducing $q(\mathbf{z})$ directly in the eq. 2.11, such that we can optimize the term $\mathcal{L}(q, \boldsymbol{\theta})$ to reduce the divergence $D_{KL}\big[q(\mathbf{z}) \| p(\mathbf{z}|\boldsymbol{x}, \boldsymbol{\theta})\big]$ between approximation and true posterior. Using (2.12), we then obtain

$$\begin{aligned}
\mathcal{L}(q, \boldsymbol{\theta}) &= \int q(\mathbf{z}) \log \frac{p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} \\
&= \mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\theta})] - \mathbb{E}_{q(\mathbf{z})}\left[\prod_i \log q_i(z_i)\right] \\
&= \mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\theta})] - \sum_i \mathbb{E}_{q(\mathbf{z})}[\log q_i(z_i)]
\end{aligned} \tag{2.14}$$

the mean-field assumption allowing to factorize the distribution $q(\mathbf{z})$. As each latent dimension are optimized in turn, we can isolate the dependencies of each latent dimension z_j

$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(z_j)}[\mathbb{E}_{q(\mathbf{z}_{i\neq j})} \log p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\theta})] - \mathbb{E}_{q(z_j)}[\log q_j(z_j)] + C \tag{2.15}$$

$$= \mathbb{E}_{q(z_j)}[\log \tilde{p}_j(\boldsymbol{x}, x_j|\theta)] - \mathbb{E}_{q(z_j)}[\log q_j(z_j)] + C \tag{2.16}$$

where $C = \sum_{i\neq j} \mathbb{H}[q_i(z_j))]$ is a constant with respect to z_j. Optimization can thus been made in turn for each $\log \tilde{p}_j(z_j, \mathbf{z})$, whose expression is now tractable as the co-dependencies with other z_i have been excluded. For more examples see [50, 51].

Variational learning is thus very useful to approximate a posterior distribution over parameters or latent variables, using the mean-field property of the variational distribution to obtain tractable expressions for th different components. It also allows to define custom dependencies across parameters, and is compatible with a lot of distributions, being generalizable to exponential families [52, 53]. This versatility then motivated the use of variational inference for numerous applications such as graphs [51], bandits [54], Gaussian processes [55], and state-space models [56] (see chapter 4). Furthermore, variational methods can also be used for black-box inference, allowing us to use neural networks to model generative and inference distributions. This method, that is mandatory in our work, will be explained section

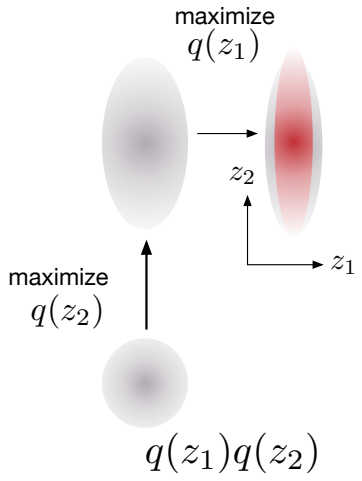


Figure 2.10: Assuming the Independence between z_1 and z_2 allows to optimize separately the components of the distribution, optimizing in turn until convergence.

2.3.1.

2.1.3.2 Variational sub-optimality.

Despite the versatility and the efficiency of variational learning, it can still face some sub-optimality issues, that can be unveiled by derivating further decompositions of the bound (2.14)

$$\mathscr{L}(q, \boldsymbol{\theta}) = -\mathbb{E}_{q(\mathbf{z})}[\log \frac{q(\mathbf{z})}{p(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\theta})}] + \mathbb{H}[p(\boldsymbol{x})] \tag{2.17}$$

this decomposition, called *joint-contrastive*, shows that maximizing the lower-bound $\mathscr{L}(q)$ maximizes the joint model $p(\boldsymbol{x}, \mathbf{z})$, up to some constant that is the entropy of the data. Alternatively, we can also derive

$$\begin{aligned}\mathscr{L}(q, \boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{x}|\mathbf{z}) + \log p(\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] \\ &= \mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\theta})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z}) - \log p(\mathbf{z})] \\ &= \mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{x}|\mathbf{z}, \boldsymbol{\theta})] - D_{KL}[q(\mathbf{z})\|p(\mathbf{z})]\end{aligned} \tag{2.18}$$

this decomposition, that we can call *prior-contrastive*, clearly shows the trade-off between the divergence term $\mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{x}|\mathbf{z})]$, that evaluates the data likelihood under the model $p_\theta(\boldsymbol{x}|\mathbf{z})$, and the divergence between approximated posterior $q(\mathbf{z})$ and the real prior $p(\mathbf{z})$. In the optimal case, the second term equals 0, such that the likelihood$p(\boldsymbol{x}|\mathbf{z}$ is maximal. This decomposition then points out two possible sources of sub-optimality, that can prevent the divergence to cancel and then lowering the bound $\mathscr{L}(q, \boldsymbol{\theta})$: formulating a distribution $q(\mathbf{z})$ is unable to match the prior $p(\mathbf{z})$, or choosing a prior distribution $p(\mathbf{z})$ that is not optimal for the overall model $p_\theta(\mathbf{x}, \mathbf{z})$). Additionally, sub-optimality can also emerge from the Kullback-Leibler divergence D_{KL} itself, leading the variational distribution $q(\mathbf{z})$ to degenerated solutions. In the following paragraphs, we will shortly detail the origin of these issues, as well as various solutions that can be taken to alleviate this degeneracy.

Mean-field approximation. We saw from eq. 2.18 that the lower-bound $\mathscr{L}(q, \boldsymbol{\theta})$ was penalized by the divergence between the approximated posterior $q(\mathbf{z})$ and the prior $p(\mathbf{z})$. Hence, if the chosen family variational $\mathbb{Q}$ for the distribution q is not able to match the prior $p(\mathbf{z})$, an unavoidable cost will be added to the lower-bound (2.14), and providing poorly approximated models. Mean-field distributions, by example, cannot express any dependency between latent variables and latent dimensions, can can then provide poor approximations of the distribution $p(\mathbf{z}|\mathbf{x})$.
Researchers then investigated solutions to make these distributions more expressive, reducing the gap to provide a tighter lower bound. Tractable variational strategies to express latent covariance was proposed by Saul & al. [57], allowing partial grouping of dimensions q_j. Some authors rather propose to replace the common Gaussian

assumption for both prior and posterior with more complex distributions, such as mean-field exponential families [58, 59], increasing the expressive power of the involved models. Dirichlet processes[60], such as Stick-Breaking Processes [61] or nested Chinese Restaurant Processes (nSCRP) [62] can be beneficially used in variational inference (see fig 2.11), empowering the flexibility of the variational approximation. Alternatively, mixture of gaussians [63], Gaussian processes [55] or graphical models with hierarchical conditioning [64] or auxiliary variables [65] (see fig. 2.12)can be leveraged to increase the expressiveness of the variational family.

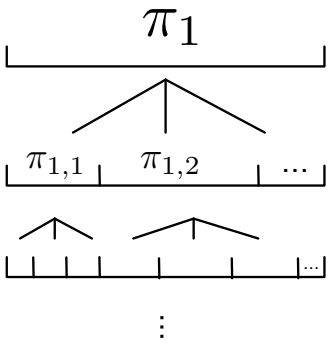


Figure 2.11: Example of hierarchical stick-breaking process, allowing to define arbitrary partitions over a probability space

While showing the flexibility of variational approaches, the *explicit* definition of a given family for the variational distribution can also be seen as a limitation. *Implicit methods* rather propose to infer variational distributions without explicitly modeling their parameters, using distribution matching methods to regularize the intractable posterior with the prior. A way to perform implicit distribution matching is to use *implicit distributions*, modelled by a function $\mathbf{z} = f(\boldsymbol{\epsilon})$ over some source of noise $\boldsymbol{\epsilon}$, that is learned during training to shape the posterior [66]. As the expression of its density function is intractable, comparing the two distributions using respective samples can be achieved with *Density Ratio Estimation* (DRE) [67], estimating the ratio between both densities $p(z)/q(z)$ to evaluate their similarity (see fig. 2.13)

$$r(\mathbf{x}) = \frac{p(\mathbf{x}))}{q(\mathbf{x})} = \frac{\hat{p}(\mathbf{x}|y=1)}{\hat{q}(\mathbf{x}|y=0)} = \frac{\mathscr{D}(\mathbf{x})}{1-\mathscr{D}(\mathbf{x})} \tag{2.19}$$

Hence, the Kullback-Leibler divergence in (2.18) can be replaced by a DRE estimator, allowing us to estimate the divergence between the two implicit distributions. The application of implicit regularization for latent inference have been proposoed by Dumoulin & al. with the Adversarially Learned Inference framework [68] (ALI), and was then proposed in variational learning with by Makhzani & al. [69] and Mescheder & al. [70] Alternatively, Laplace approximation of implicit distributions for can be used to compute their D_{KL} [71] ; however, this method do not alleviate the weaknesses brought by mean field asymmetries.

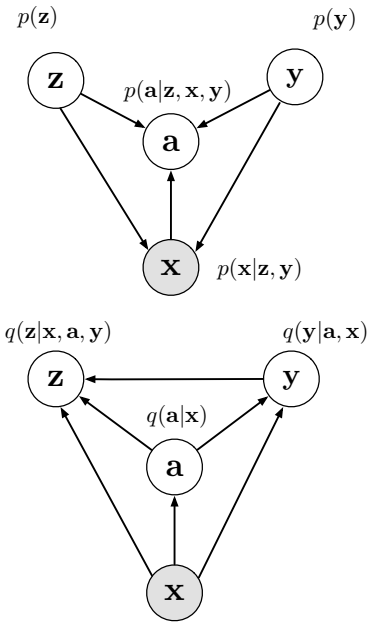


Figure 2.12: *Auxiliary variables* allow to increase the expressiveness of the inference model, bot does not modify yet the generative distribution.

While these implicit variational methods are very efficient to generate expressive distributions, they suffer from two main drawbacks. First, AVB and ALI are both based on a *discriminator*, that can is trained to accurately separate the two distributions, then introducing additional parameters and replacing the choice of variational distirbution by the choice of a discriminator. Secondly, implicit methods remove the *interpratibility* of the variational distributions, as the underlying distribution between the model $q(z)$ is not explicitly defined. Some methods, such as Semi-Implicit Variational Inference, addresses the second issue by formulating a two-stage variational distribution $q(z) = \int q(z|\boldsymbol{\psi})q(\boldsymbol{\psi})$, where the distribution $q(\boldsymbol{\psi}$ is allowed to be implicit while $q(z|\boldsymbol{\psi})$ is held explicit [72]. The same principle can also be applied to the prior $p(z)$ [73], allowing to define both both prior and variational distribution as semi-implicit. Alternatives to DRE can also be chosen, matching the

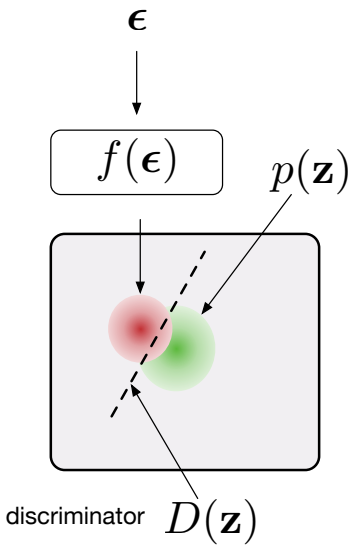


Figure 2.13: Density ratio Estimation uses a classifier to separate one distribution from the other, and replaces the original probabilities by classification rates

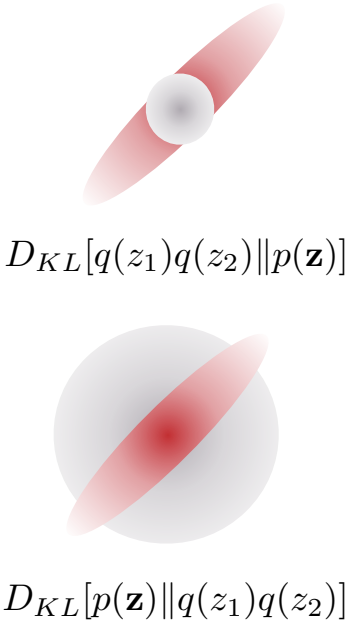


Figure 2.14: Effect of D_{KL} asymmetry on with variational distributions. As $q(\mathbf{z})$ cannot model $p(\mathbf{z})$, the effect of D_{KL} asymmetry entails a sub-optimal distribution matching, whether *mode-focusing* (above) or *mass-covering* in the other (below)

moments of the two distributions with kernel density ratio fitting [74] or maximum mean discrepancy [75] (see below).

D_{KL} sub-optimality Kullback-Leibler divergence is the most used way of computing the divergence between two distributions, as it naturally occurs in information theory (see sec. 2.1.4) and has a tractable expression for most distributions. However, some properties of the D_{KL} can also prevent the optimal maximization of the variational distribution. D_{KL} can be expressed

$$D_{KL}[p(x)||q(x)] = \int p(x) \log \frac{p(x)}{q(x)} dx \tag{2.20}$$

D_{KL} is asymmetric, and then has two different behaviors depending on the order of the involved distributions. Optimizing $D_{KL}[q(z)||p(z)]$, will enforce $q(z$ to have a *mode-seeking* behavior, heavily penalizing zones of $q(z)$ where $p(z)$ is low (hence called *zero-forcing)*. This property then rather under-evaluate the variance of the original $p(z)$, focusing on the mode but maybe missing most of the target support. Reversely, optimizing $D_{KL}[p(z)||q(z)]$ will push $q(z)$ to cover all the support of $p(z)$ to have a *mass-covering* behavior, then over-estimating the target variance (hence *zero-focusing*). While these properties are inactive if the chosen variational family $\mathbb{Q}$ is able to match the real posterior perfectly, this behavior will be decisive in case of sub-optimal variational approximations (see fig. 2.14).

Hence, while altering the original formulation of variational learning, alternative divergences have been proposed to replace the original D_{KL} of (2.18). In addition to the DRE estimation presented above for implicit distribution matching an important alternative divergence is the *Rényi Divergence* (RD), that is derived from an extension of the original notion of Shannon entropy (see section 2.1.4). This divergence depends on a parameter α, that balances between under- and over-estimation of the divergence : $\alpha \leftarrow \infty$ enforces a zero-forcing behavior, and reversely $\alpha \leftarrow 0$, enforces a zero-avoiding optimum [76]. Note that, in that case, $\mathcal{L}_\alpha$ will be a lower bound to $p(\boldsymbol{x})$ if $\alpha > 0$, and an upper-bound if $\alpha < 0$. Li & al. then propose to sandwich $p(\boldsymbol{x})$ with several α, ensuring the optimality of the solution [77]. Another divergence, χ^n-divergence, is based on the same principle, increasing the its mode-seeking behavior when increasing the power index n. The β-divergence and $\gamma-$ divergence [78] are rather adjusting the sensitiveness of the divergence to low-probability zones of $p(\boldsymbol{x})$, allowing to adjust its sensitivity to outliers. The proposed $\alpha\beta$-divergence propose to blend Renyi and β-divergences, allowing to trade between both robustness/sensitivity and mode-seeking/mass-covering behavior [79].

Alternatively, the generalization of divergences using external defined convex functions f were proposed in the literature, providing a theoretical basis for probability metrics. f-divergences, deriving a divergence for any convex function f verifying $f(0) = 1$, such that

$D_f[p\|q] = \int q(\boldsymbol{x}))f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right)d\boldsymbol{x}$. By example, choosing $f(x) = x\log x$ recovers the D_{KL} 2.20 [80]. This divergence can be approximated for any valid $f(\cdot)$ with a Taylor expansion of the function, or resorting to DRE. Furthermore, such functions f can be understood as a restricted class of cost functions used for the *optimal transport* formulation of probabilistic inference (see sec 2.1.3.3). Alternatively, *Bregman divergences* rather defines the divergence $D_f = f(p) - f(q) - < \nabla f(q), p - q >$, where $\nabla f(q)$ is the first-order Taylor expansion of $f(q)$ and $< \cdot, \cdot >$ is the scalar product. A noticeable property of Bregman divergences is that they can be used as a substitute for affine connections in the field of *information geometry*, as they define a dual potential through Legendre transform. This property of Bregman divergences allow to generalize many techniques of optimization theory to be performed in the space of probability distributions, that has been used for example in audio analysis with Music Information Geometry by Cont & al. [81].

Finally, alternative divergence methods leverage statistical testing methods to evaluate the divergence directly using samples for the distributions, allowing an implicit approach to distribution matching. Maximum Mean Discrepancy (MMD) is a two-sample test that computes distance between the means of the distributions by projecting the samples into a kernel Hilbert space (RKHS) [75]. MMD can be formulated

$$\mathrm{MMD}_f(p(\boldsymbol{z}), q(\boldsymbol{z})) = \sup_{f \in /mathcalF} \left(\mathbb{E}_{p(\boldsymbol{x})}[f(\boldsymbol{x})] - \mathbb{E}_{q(\boldsymbol{x})}[f(\boldsymbol{y})]\right) \tag{2.21}$$

$$= \| \int_{\mathcal{Z}} k(z, \cdot)dp(\boldsymbol{z}) - \int_{\mathcal{Z}} k(z, \cdot)dq(\boldsymbol{z})\| \tag{2.22}$$

where an usual choice for k is the radial-basis function kernel $k(\cdot, \cdot) = \exp(\boldsymbol{x} - \boldsymbol{x'})^2$. MMD have been successfully applied to generative model as a training criterion in the data domain [82, 83], but can also be used as a dissimilarity measure between the two probabilities $p(\boldsymbol{z})$ and $q(\boldsymbol{z})$ [84, 85] (see sec. 2.1.3.3).

2.1.3.3 Optimal transport.

Some recent approaches propose to address statistical inference under an optimal transport perspective. Optimal Transport (OT) is a mathematical problem that consists of finding the optimal coupling between two probability spaces under a give cost function [86]. This general problem has been extensively studied in various fields such as biology, fluid dynamics, or physics, even recently in audio processing [87]) but also in the domain of probabilities, then recently introduced in the machine learning field. Indeed, statistical inference of latent from observed variables can be understood as finding a transport map between $\boldsymbol{x}$ and $\boldsymbol{z}$, optimized using a given cost function c. Defining $\Lambda(p, q)$ the space of joint probabilities whose respective marginals are

$p(\boldsymbol{x}, \boldsymbol{z})$ and $q(\boldsymbol{x}, \boldsymbol{z})$, optimal transport can be formulated

$$W_c(p, q) : \inf \gamma \in \Lambda(p, q) \int c(\boldsymbol{z}, \boldsymbol{x}) d\gamma(\boldsymbol{z}, \boldsymbol{x})$$

that is a divergence between the two probabilities p and q. Generative models can then be formulated from an optimal transport point of view, finding the transport map from the empirical distribution $p*(\mathbf{x})$ to a target distribution $p(\mathbf{x})$ and minimizing the above Wasserstein divergence. While some methods are based the direct optimization of this problem, using Minimum Kantorovitch Estimators or Sinkhorn's algorithm [88, 89], we can also use OT formulate a latent inference problem by taking a distribution $p(\mathbf{x})$ that is itself obtained from the transformation of a variable z, assuming $\Lambda(p, q) = \int_{\mathcal{Z}} p_G(\hat{x}|z) q(z|x) p_X(x) dz$, such that $q(\mathbf{z}) = \int q(\mathbf{z}|\mathbf{x}) p_X(\mathbf{x}) dx = p_Z(z)$ for all $z \in \mathcal{Z}$ [90], such that the ideal OT optimization problem can be taken as (see fig. 2.15)

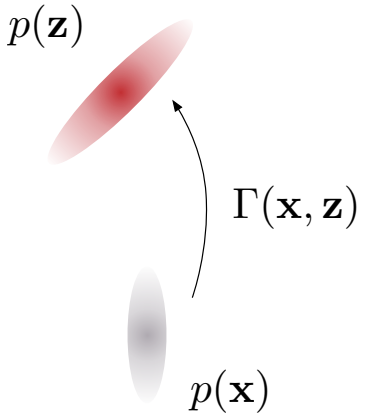


Figure 2.15: Optimal transport studies the distribution over mappings $\Gamma(\mathbf{x}, \mathbf{z})$ to transform $p(\mathbf{z})$ in $q(\mathbf{z})$ that minimizes the cost $c(\mathbf{x}, G(\mathbf{x}))$

$$W_c(P_X, P_G) \leq Wc^{\dagger}(P_X, P_G) = \inf Q : Q_Z = P_Z \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)}[c(X, G(Z))]) \tag{2.23}$$

where $W_c^{\dagger}$ is optimal if the generative function is atomic, such that $p_G(\hat{y}|z = z)$. A relaxed formulation can be written

$$W^{\beta}(P_X, P_G) = \inf_{q(z|x)} \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)}[c(X, G(z))] + \beta F(Q)$$

where F is a regularizer such that $F(Q) = 0$ when $p(\mathbf{z}) = q(\mathbf{z})$. The comparison with the ELBO 2.18 is straightforward, taking the cost $c(\mathbf{x}, G(\mathbf{z}))$ as the likelihood function. However, the main difference with the variational formulation is the regularization term, applied on the *aggregated posterior* $q(\mathbf{z})$ instead of $p(\mathbf{z})$. Furthermore, Bousquet & al. argue that the noise added to the decoder, such as in the case of VAE where the output is a factorized normal distribution, add a penalty of $\sum_{i=1}^{D} \sigma^2$ to the objective 2.23 and indubitably yields a non-optimal solution. This conclusion reinforces the propositions of implicit regularization methods, such as Adversarial Bayes or. Based on this conclusion, Wasserstein Auto-Encoders (WAE) then propose to use Maximum-Mean Discrepancy as a latent regularizer (see divergences), and propose an OT formulation of the Adversarial Auto-Encoder [91].

Alternatively, the Wassertein distance can also be used in variational inference to optimize the divergence $\mathbb{L}(,) = W_c(p(\mathbf{z}, \mathbf{x}), q(\mathbf{z}, \mathbf{x}))$. This divergence can be approximated with a Monte-Carlo estimator $\mathcal{L}_C(p_n, q_N) = \inf_{\lambda} \sum_{j,k} C^{p,q}(x_1^{(j)} z_1^{(j)}, x_2^{(k)}, z_2^{(k)})$, that generalizes f-divergences taking $C^{p,q}(x_1, x_2) = f(p(x_2)/q(x_2))$.

2.1.4 Information theory and latent models

Before moving on to dimensionality reduction and representation learning methods, we first introduce some important notions of *information theory*, that will help us to bridge the concept of *latent variables*

with the one of *representation*. Information theory, known to be initiated by Shannon in telecommunications [14], is a seminal sub-field of signal processing that leverages probabilities to quantify the amount of *information* brought by a given signal, centered on the notion of *entropy*. Information-theory based signal analysis can be seen as a probabilistic framework that aims to describe global features of the signal, estimating its probability law and quantifying the amount of *order* of the underlying generative process. Information theory then allows to formulate the notion of *latent code*, that describes the underlying structure of the signal by extracting the minimum description of the system that maximizes the information. The notion of code can thus be related to the idea of *latent variables* presented above, and will provide the essential interface between *compression* algorithms and Bayesian learning that grounds the results of this thesis. Furthermore, the *code* of a given signal can be understood as a way to describe the inner structure of a signal, discarding its non-informative content. Hence, extracting a code from a given signal can be thought as extracting its structuring laws, bridging with generative models.

2.1.4.1 Information measures

Shannon Entropy. The central notion of information theory is then the notion of *information*, that can be defined as the amount of knowledge brought by an observation to the analysed system. This quantity can the be intuited by the notion of *surprise* brought by an element of the signal, considering that an unlikely observation will bring a more substantial amount of information than an observation that would be already considered probable. Hence, it also relates the amount of *redundancy* of the signal, that is a fundamental notion of compression algorithms and signal analysis. Information theory then uses a probabilistic framework to quantify the order amount of a signal, taking inspiration from thermodynamics with the notion of *entropy*. The entropy of a signal $\mathbf{x}$ with possible values $\{x_1 ... x_N\}$ can be given by the following expression

$$\mathbb{H}[X] = -\sum_{i=0}^{N} p(x_i) \log p(x_i) \tag{2.24}$$

that can be seen as the expectation of the *information content* $-\log p(x_i)$ of each symbol. As $p(x_i) \in [0; 1]$, we can see that a very low probability event will bring an important amount of information, while an almost certain element will bring very low information. The entropy then quantifies the *amount of disorder* of a signal, and then also describes its *uncertainty*. By example, the uniform distribution $\mathcal{U}[a, b]$ is the maximal entropy distribution that can be defined on a support $]a, b]$, while the entropy of the empirical distribution of a time series $\frac{1}{N}\sum_{i=1}^{N} \delta_{x_i}$ has a minimum entropy $-\log N$, that tends to 0 when number of elements tend to 1. A similar notion can be derived in the continuous domain, called *differential entropy*)

$$\mathbb{H}[X] = \int p(x) \log p(x) dx \tag{2.25}$$

that defines a continuous entropy estimator using the probability density function $p(\mathbf{x})$, replacing the discrete values taken by the previous $p(\mathbf{x}_i)$ by infinitesimal interval $d\mathbf{x}$. This notion is thus fundamental, describing the amount of *uncertainty* of the continuous signal and then quantifying is *compressibility*. The definition of entropy can be extended to describe the uncertainty of mutual incoming signals, then quantifying their mutual amount of redundancy, even if they are defined on separate domains. The *joint entropy* can be then defined

$$\mathbb{H}[X,Y] = \int p(x,y) \log p(x,y) dxdy \tag{2.26}$$

describing the entropy of the *joint probability* of the two signals. If X and Y are independent, such that $p(x,y) = p(x)p(y)$, the entropy of both variables are summed : $\mathbb{H}[X,Y] = \mathbb{H}[X] + \mathbb{H}[Y]$. If the signals are not independent, meaning that they *share* some information, the joint entropy will be lower than their individual sum. Joint entropy then also quantifies the correlation between two signals, cancelling if they share the same probability density. Another interesting value is the *conditional entropy*, that computes the additional amount of information brought by a random variable X given a another random variable Y

$$\mathbb{H}[X|Y] = \int p(x,y) \log p(x|y) dxdy \tag{2.27}$$

such that $\mathbb{H}[X|Y] = \mathbb{H}[X,Y] - \mathbb{H}[Y]$, meaning that the joint entropy $\mathbb{H}[X,Y]$ is given by the individual entropy $\mathbb{H}[Y]$ plus the additional amount of information $H[X|Y]$. A related quantity, that will be important in the frame of variational methods, is the *mutual information*

$$\mathbb{I}[X,Y] = \int p(x,y) \log \frac{p(x,y)}{p(x),p(y)} dxdy \tag{2.28}$$

that quantifies how many information can be gained by observing a variable when the other one is observed, or equivalently how the amount of independence between X and Y. Unlike conditional entropy, mutual information is a symmetric estimator, and can be understood as being the D_{KL} (see 2.20) divergence between the joint probability $p(x,y)$ and the product $p(x)p(y)$

$$\mathbb{I}[X,Y] = D_{KL}[p(x,y) \| p(x)p(y)] \tag{2.29}$$

such that mutual information cancels if the two variables are independent, as $p(x,y) = p(x)p(y)$. Information theory provides us estimators for quantifying the uncertainty amount of signals, and provides a framework to evaluate the amount of information shared between signals. Note that while this theory was initially developed for time series analysis, there is no explicit assumption relation to time, such that these estimators can then be also used on other types of data. Information theory can then be used to describe global properties of the involved distributions, involving all examples of the target data. Connecting with the previous section, information-theoretic estimators can then also be used in Bayesian inference, especially with latent models where it can derive interesting estimators between $\mathbf{x}$ and $\mathbf{z}$.

Rényi entropy The original formulation of entropy was originally designed to ensure information extensibility, i.e. the summability of information given two independent events. $\mathbb{H}[x, y] = \mathbb{H}[x] + \mathbb{H}[y]$. Shannon's entropy is only the first solution of this constraint, the second one being the *Rényi Entropy* [92]

$$\mathbb{H}_\alpha[p(\boldsymbol{x})] = \frac{1}{1-\alpha} \log \int_{\mathbb{R}^d} p(\boldsymbol{x})^\alpha d\boldsymbol{x} \tag{2.30}$$

that is then dependent of a parameter α, tuning tune to importance of low-probability events in the divergence. Renyi entropy recovers the *collision entropy* for $\alpha = 2$, that is a well-known entropy estimator reducing the bias for short series. Rényi entropy also generalizes other estimators, such as the *Hartley entropy* $\mathbb{H}_0[p(\boldsymbol{x})] = \log |\text{supp} p(\boldsymbol{x})|$ or *min-entropy* $\mathbb{H}_{+\infty}[p(\boldsymbol{x})] = -\log \max p(\boldsymbol{x})$ [93]. Similarly to Shannon's entropy, we can derive a corresponding divergence, thus called *Rényi Divergence* [76]

$$D_r^\alpha[p(\boldsymbol{x})\|q(\boldsymbol{x}] = \frac{1}{\alpha - 1} \int_{-\infty}^{\infty} p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha} d\boldsymbol{x} \tag{2.31}$$

that is also dependent of a parameter α. Similarly to Rényi entropy, Rényi divergence is also a generalization of KL divergence (that it equals at $\alpha = 1$)) , such that $D_0^r[p\|q] = -\log q(p > 0)$, and $D_\infty^r[p\|q] \log \text{esssup} \frac{p}{q}$ is the worst-case regret in the minimum description length principle. Some noticeable values for α are also $\alpha = 0.5$, where in this case $D_{0.5}^r[p\|q] = -2 \log 1 - \text{Hel}^2[p\|q]$ and $\alpha = 2$, where $D_2^r[p\|q] = \log 1 + \chi^2[p|q]$ Another interesting property is that, R'enyi divergence is also continuous on α, with a discontinuity at $\alpha = 1$.

2.1.4.2 Latent rate-distortion trade-off

Information theory then provides global estimators describing the amount of uncertainty of given distributions. These estimators can then be used to provide a interesting understanding of Bayesian models, analyzing the dependence between latent variables $\mathbf{x}$ and $\mathbf{z}$ in terms of information. More precisely, it can quantify how much the provided latent variables act as a *code* of the data, then bridging the Bayesian concept of *latent factors* and the idea of *compressibility* of the observed data. This possible equivalence between latent factors and data compression will allow us to bridge it to the idea of *representation*, that will be seminal in this thesis. Here, we will focus on *rate-distortion* theory, that studies the possible amounts of information shared by a set of data samples and its corresponding latent code.

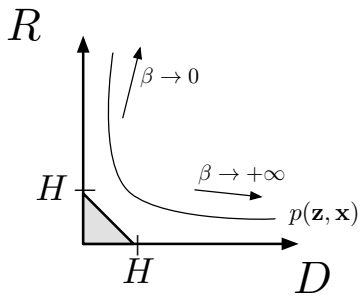


Figure 2.16: Rate-distortion trade-off : given a distribution $p(\mathbf{x}, \mathbf{z})$, there is an intrinsic trade-off between optimal rate R and optimal distortion D, threshold by the entropy H

Rate-distortion theory [94] studies the relation between the length of a compression code, called the *rate*, and *distortion* by retrieving a given signal from its corresponding code. Given a distribution $q(\mathbf{z})$ for the code and a generative distribution $p(\mathbf{x}|\mathbf{z})$ (then similar to decompression), the *distortion* can be expressed as

$$\mathbb{D}[p, q] = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), \mathbf{z} \sim q(\mathbf{z}|\mathbf{x})}[d(\mathbf{x})] \tag{2.32}$$

given a distortion measure $d(\cdot)$. In Bayesian optimization, this measure can be seen as the likelihood of the data given $\mathbf{z}$, such that $d(\mathbf{x}) = p_\theta(\mathbf{x}|\mathbf{z})$. Estimating the *rate* of the representation is more difficult with codes defined in $\mathbb{R}$, as the notion of *code length* does not really exist in the continuous domain. TIshby & Zaslavsky then propose to define the rate as the mutual information between the data and the code obtained from $q(\mathbf{z}|\mathbf{x})$ [95]

$$\mathbb{R}[p,q] = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})\mathbf{z}\sim p(\mathbf{z}|\mathbf{x})}\left[q(\mathbf{z}|\mathbf{x})\log\frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] \tag{2.33}$$

Distortion and rate can be used to evaluate the efficiency of the compression, as minimizing the first will reduce the error of the reconstruction, and minimizing the second will reducte the code length. We can define the *representation mutual information* by quantifying the mutual information using the implicit *aggregated posterior* $q(\mathbf{z})$, that is the representation obtained by marginalizing the posterior $q(\mathbf{z}|\mathbf{x})$ over $\mathbf{x}$ [96]

$$\mathbb{I}_{repr}[p,q] = \mathbb{E}_{\mathbf{x},\mathbf{z}\sim p(\mathbf{x},\mathbf{z})}\left[q(\mathbf{z}|\mathbf{x})\log\frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{x})q(\mathbf{z})}\right] \tag{2.34}$$

where $p(\mathbf{x},\mathbf{z})$ is defined as $p(\mathbf{x})q(\mathbf{z}|\mathbf{x})$. We can show that this value is lower-bounded by the distortion and upper-bounded by the rate

$$\mathbb{H}[p(\mathbf{x})] - \mathbb{D}[p,q] \leq \mathbb{I}_{repr}[p,q] \leq \mathbb{R}[p,q] \tag{2.35}$$

such that, if both $\mathbf{x}$ and $\mathbf{z}$ are independent, $\mathbb{I}_q = 0$ and then $\mathbb{R} = \mathbb{H}$ while $\mathbb{D} = 0$, meaning that the code is optimum (*auto-encoding limit*). Reversely, $\mathbb{R} = 0$ implies $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$, showing that the rate is quantifying the extra-cost we pay by not fitting the prior distribution. In that case, $\mathbb{D} = \mathbb{H}$, that is called the *auto-decoding limit*. Rate-distortion decomposition can be derived to obtain as minimization criterion to obtain an optimal representation $\mathbf{z}$

$$\begin{aligned}&\min \mathbb{D} + \beta\mathbb{R}\\ &\min \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}[-p(\mathbf{x}|\mathbf{z}) + \beta\frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}]\end{aligned} \tag{2.36}$$

that is closely relating to the ELBO (2.18), plus a factor β controlling the rate/distortion rate. Here, we show that the variational objective inherently achieves a trade-off between minimum distortion (how can we achieve maximum likelihood with a model $p(\mathbf{x},\mathbf{z})$) and minimum rate (that is the compression of the representation). The *rate* can then be defined as the distribution matching between the posterior $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z})$, that can be intuited as the average additional number of bits we have to add to $p(\mathbf{z})$ to identify $p(\mathbf{z}|\mathbf{x})$. A high rate thus means of lower sensitivity of the latent space to the data, and if the prior $p(\mathbf{z})$ is an isotropic normal distribution, an higher independence of the dimensions. This last observation is very important if we want the extracted latent factors $\mathbf{z}$ to be disentangled (see section 2.3.2.4).

2.1.4.3 Minimum description length and bits-back coding.

In information theory, *modeling* the signal is then related to *compression*, leveraging the redundancy of the signal to reduce its code length whether without altering the contained information (called *lossless compression*), or targeting an subset of information. The obtained compression code can be considered as a representation of the signal of lower dimensionality. Hence, we can presume that, given a subset of signals, the compression with the best rate is the one that manages to retrieve the underlying structure of the signal. If we consider the target data as generated by a set of factors, finding an optimal compression amounts to recovering these generative factors, retrieving the optimal representation from the signal. An interesting bridge then happens between statistical inference and variation theory : optimizing a model $p_\theta(\mathbf{x}$ with K parameters on a dataset with N data can then be considered equivalent as finding an optimal code of length K given a fixed model, defined by the probability distribution. Defining a latent model $\mathbf{z}$ can then be defined in a similar way, where $\mathbf{z}$ acts as the compressed information about $\mathbf{x}$.

A theoretic framework based on this philosophy is the *Minimum Description Length* principle (MDL), proposed by Grünwald & al. [19, 97], for a given dataset, the best model is the one that requires the less amount of bits to be communicated. MDL were developed to use the model size as a minimization criterion, imposing a code length constraint to prevent over-fitting of models. An example of such criteria is the Akaike Information Criterion (AIC), that reformulates the likelihood objective as

$$AIC = 2K - 2\log p(\mathbf{x}|z, \boldsymbol{\theta})$$

where K would be the total number of parameters (latent variables included). AIC then explicitly adds the code length of the model to the evaluation, and can be used as a criterion for model selection. An MDL approach then consists in defining a *cost* that the receiver of the message will pay to recover the original data, including the cost of the model itself. Applying MDL to Bayesian learning, we can then define three different costs : a *reconstruction cost*, that quantifies the errors made by the model, the *model cost*, that describes the number of parameters of the model, and the *code cost*, that quantifies the quality of the code [98]. An interpretation of the code cost can enlighten a common point between Bayesian learning and MDL : the more accurately a given hypothesis explains a data, the lower is the cost paid to face new incomers. This idea also provides the background of the *free energy* of Helmholtz systems, as Markov fields.

An information theoretic formulation of latent models could thus be described as follows : an emitter sends the code z to a receiver, who uses the model $p(\mathbf{x}|\mathbf{z})$ to recover the data $\mathbf{x}$ (here $\mathbf{z}$ represents both latent variables and model parameters). The total code length of this configuration is then the addition of the model code length, the reconstruction code $-p(\mathbf{x}|\mathbf{z})$, plus the code cost $-p(\mathbf{z})$, such that the

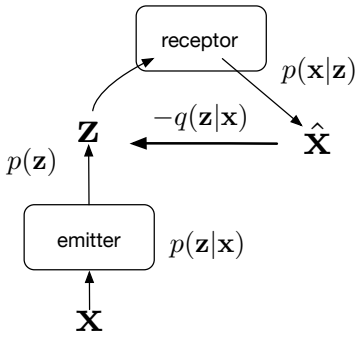


Figure 2.17: *Bits-back* in MDL formulation of variational inference

cost paid for probable event is little. However, this code definition does not take into account the stochasticity of the code, as $\mathbf{z}$ also contains random bits added during sampling. Then, if the receiver has the same encoding algorithm $q(\mathbf{z}|\mathbf{x})$ than the emitter, it can also recover these random bits to reduce the total cost, bring the *bits-back* (see fig. 2.17). The total cost length of the system is then added a term $q(\mathbf{z}|\mathbf{x})$, the additional cost of random bits that can be recovered. This learning scheme, called *bits-back coding*, then amounts to the variational formulation developed previously. The relation between bits-back coding and variational learning can helps to understand the behavior of the system from an information-theoretic point-of-view, such that the *information preference* of variational auto-encoders (see section 2.3.2) and can motivated the development of specific MDL training criteria [99] [98].

2.2 From dimensionality reduction to deep unsupervised learning

In the previous section, we investigated statistical inference methods under a Bayesian perspective, modeling a given set of observations using a parametric model. We also introduce the idea of latent models, using external variables can can be used used to *condition* the generative distribution. The concept of latent variables is seminal in Bayesian methods, as it allows to explain the data with higher-order generative parameters. However, Bayesian inference requires some tractability assumptions over its terms, and thus limit the possible choices generative models. Fortunately, we saw that the tractability of all the involved terms could be bypassed by transforming the closed-form evaluation of the Bayes' rule into an optimization problem, allowing to model more complex generative distributions. More particularly, we presented *variational inference*, that allows to maximize the likelihood by replacing the true posterior $p(\mathbf{z}|\mathbf{x})$ by an *approximated* posterior $q(\mathbf{z})$. This method relaxes the tractability of the posterior, and then allows to train more complex generative models. However, the variational methods presented still relied on the tractability of the ELBO 2.18 to get closed-forms for optimal variational distributions.

In this section, we will address the extraction of high-level features from the perspective of *representation learning*, a domain based on dimensionality reduction techniques and motivated by *automatic discovery* of features, visualization, and compression. This approach is different from statistical inference as it does not assume the invertibility of the obtained representation, and does not necesseraly resort to probabilities techniques. Formally, a *representation* can be simply described as a function $\mathbf{z} = f(\mathbf{x})$ that embed an input space into another one with additional constraints, by example on its dimensionality, organization, or enforced to reflect given properties of the data. The motivations for representation learning algorithms are then quite diverse, and highly depending on the target application. *Dimensionality*

reduction techniques are focused on the extraction of representations that aims to preserve most of the data diversity, extracting intrinsic sources of variance inside the dataset. This approach can be motivated by *manifold hypothesis*, stating that high dimensional data lies in the neighborhood of a low-dimensional sub-manifold of the input space. Recovering the sub-manifold with dimensionality reduction techniques then targets to reocver the inner structure of the data, and hopefully its natural factors of variation [20, 100].

Representation learning is also widely investigated in the field of *machine learning*, a family of statistical models based on the optimization of given set of tasks or criteria. Dimensionality reduction is indeed an efficient remedy to *curse of dimensionality*, that can make the application of machine learning methods impossible in high-dimensional input space. Indeed, the volume unit of a space $\mathbb{R}^D$ grows exponentially with D, such that Euclidean distances between two points tends to be less significant, converging to some constant [101]. As almost every machine-learning methods are based on the optimization of Euclidean metrics of some values of interest, curse of dimensionality can be critical and then prevent to directly use these techniques in the data space. Therefore, resorting to dimensionality reduction techniques can allow to alleviate this problem, first projecting the data on a space of lower dimensionality and then applying the target machine learning task . Conversely, dimension *augmentation* can also help to sparsify the data, and then help to separate problems that are not separable in the input domain.

The recent gain of interest in machine learning has then caused the development of powerful representation extraction methods, based on high-capacity function approximators (such as neural networks) that can be trained to represent a given task. However, defining specific criteria for the extraction of robust representations, i.e. reflecting underlying properties of the data, is a non-trivial task. Indeed, obtaining a representation trained on explicitly defined tasks such as classification (hence in a *supervised* setup) can provide degenerated representations, focusing too much on the target task at the cost of losing the inner data structure. An alternative framework, called *unsupervised* learning, rather focuses on automatic feature extraction without resorting to an external task. However, it still need a suitable criteria to be trained, giving rise to substantially different approaches. In our case, we are focusing on both inference and generation of audio signals, such that the invertibility of the obtained representation is mandatory. Hence, we resort to *auto-encoding*, an unsupervised framework extracting a low-dimensional representation based on *reconstruction* criteria.

In this section, we will first present dimensionality reduction methods, that are based on the extraction of statistical, geometrical or topological descriptors of the data to provide low-dimensional representations of the data. Then, we will shortly introduce *neural networks*, widely used high-capacity function approximators that are now grounding

advanced machine learning techniques. Then, we will present *auto-encoders*, and discuss how they are related to *manifold hypothesis* and how they can be used to extract meaningful representations of the data.

2.2.1 Dimensionality reduction

In this section, we summarize methods that intend to reflect structural properties of the data based on the extraction of statistical, geometrical or topological descriptors. These dimensionality reduction methods aim to describe the inner variance of a given dataset based on three different criteria : preserving the *variance* of the samples (principal component analysis), preserving the *distance* between the samples (multi-dimensional scaling), or retrieving the underlying sub-manifold of the data (*manifold learning*). In this section, we will shortly summarize these three different methods.

Principal Component Analysis *Principal component analysis* is a seminal dimensionality reduction method, first proposed by Wold & al. [102, 103]. The idea underlying PCA is to identify the projection from the data space $\mathbb{R}^D$ to a linear subplane $\mathbb{R}^d$ that retains the maximum variance of the dataset (see fig 2.18). Mathematically, PCA can be achieved using Singular Value Decomposition (SVD), that decomposes the full data matrix $\mathbf{X} \in N \times D$ in three matrices

$$\mathbf{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^\top$$

where the orthogonal vectors $\mathbf{U}$ and $\mathbf{W}$ are the right- and left- eigenvectors along dimensions N and D, and the matrix $\mathbf{\Sigma}$ is the diagonal projection matrix with singluar values $\sigma_1 ... \sigma_p$. The transformation $\mathbf{P} : \mathbb{R}^D \rightarrow \mathbb{R}^d$ that maximizes the covariance $\mathbf{X}\mathbf{X}^T$ can be shown to be the truncated $\mathbf{W}_d$, such that PCA can be obtained with $\mathbf{X}_{PCA} = \mathbf{X}\mathbf{W}_d$. PCA can thus be understood as the projection of $\mathbf{X}$ on d-principal components, then maximizing the retained variance. Despite of its simplicity, PCA is a cheap and invertible transformation that is still widely used for multivariate data analysis. PCA then motivated numerous improvements such as *Robust PCA* [104], helping the PCA resisting to outliers, *Sparse PCA* that encourages the sparsity of the projecting axes [105], or *probabilistic PCA*, that encodes the uncertainty of the projection using a linear Gaussian model [106].

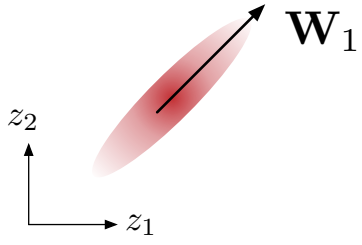


Figure 2.18: PCA will capture the axis that maximizes the variance of the original data

The main drawback of PCA is that its projection axis can be highly covariant, thus harnessing the dimensions of the projection to represent independent factors. Independant Component Analysis (ICA) reformulates the problem [107, 108] by extracting an orthogonal base from the data space by first whitening the data, and then perform PCA analysis. This method provides an interesting alternative as it enforces the independence of the dimensions, and then to recover independent variations of the data. The extension of ICA to non-linear

function spaces have been proposed by Hyvärinen, providing theoretical foundations for the uniqueness of solutions [109]. Hence, ICA is typically used in orthogonal component extraction, such as blind source separation methods.

Distance regularization. Instead of preserving the variance of the dataset, some methods rather propose to extract low-dimensional representations by preserving the pairwise *distances* of the original space (see fig. 2.19). These methods then model the distances between points rather than their absolute position, aiming to approximate the implicit metrics of a compact manifold. Multi-dimensional scaling (MDS) is a method based on optimizing the distances on a projected space $\mathcal{D}(\mathbf{z}_i, \mathbf{z}_i)$ using a mean-squared error on the original distances $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$. Given N points and the corresponding pairwise dissimilarity measures $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$ between points x_i and x_j, the objective MDS is formulated [110]

$$\min_{z_1 \dots z_N} \sum_{i \leq j} (\|x_i - x_j\| - d_{i,j})^2 \tag{2.37}$$

that minimizes the ℓ^2 norm between original and projected distances. MDS then targets to find *isometric* projections of the input space, that can be obtained by whitening the distance matrix $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$ and taking the m highest values of its SVD. However, MDS assumes an Euclidean distances, and does not model the uncertainty of the targeted distances. A probabilistic approach to MDS is provided by Stochastic Neighbour Embedding (SNE), that expresses the probability of the point $\mathbf{x}_j$ to belong to a given neighbourhood using the conditional distribution [111]

$$p(\mathcal{D}^{\mathbf{x}}_{ij}) = \frac{\exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2\right)}{\sum_{k \neq i} \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2\right)} \tag{2.38}$$

where σ^2 is the Gaussian variance of the corresponding kernel, and $\mathcal{D}^{\mathbf{x}}_{ij}$ represent the distance between the points $\mathbf{x}_i$ and $\mathbf{x}_j$. The probability $p(\mathbf{x}_j|\mathbf{x}_i)$ can then be understood as a multinomial distribution over the most probable neighbor $\mathbf{x}_j$. SNE preserve this conditional distribution in the target representation $\mathbf{z}$, where a similar density can be defined

$$q(\mathcal{D}^{\mathbf{z}}_{ij}) = \frac{\exp\left(-\|\mathbf{z}_i - \mathbf{z}_j\|^2 / 2\sigma^2\right)}{\sum_{k \neq i} \exp\left(-\|\mathbf{z}_i - \mathbf{z}_j\|^2 / 2\sigma^2\right)} \tag{2.39}$$

We can then match these two tractable distributions by minimizing the D_{KL} between $p(\mathcal{D}^{\mathbf{x}}_{ij})$ and $q(\mathcal{D}^{\mathbf{z}}_{ij})$

$$\mathbf{L}_{SNE} = \sum_i D_{KL}[p_i \| q_i] = \sum_{i,j} p(\mathcal{D}^{\mathbf{x}}_{\mathbf{ij}}) \frac{p(\mathcal{D}^{\mathbf{x}}_{ij})}{p(\mathcal{D}^{\mathbf{z}}_{ij})} \tag{2.40}$$

whose optimization is tractable using gradient descent. SNE then performs smooth dimensionality reduction method using a probabilistic model of the input pairwise distances, then providing more robust representations than MDS. Moreover, MSE intrinsically performs unsupervised clustering by minimizing the distances between close points

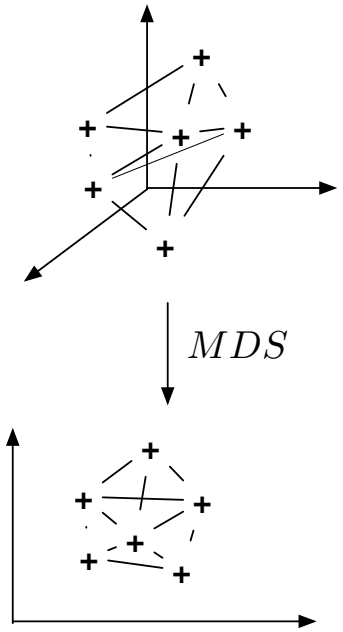


Figure 2.19: MDS tries to matches the distances of the original space in the embedded space

while pushing the others away, sparsifying the obtained representation. However, SNE is particularly subject to curse of dimensionality because of the *crowding effect*, as the Euclidean distance between original points, that exponentially grow with the number of dimensions. This undesirable effect thus tends to cancel out the saliency of the inter-point distances, and provide degenerated projected representations where the distances are preserved by projecting every points on a small support around 0. This undesirable effect can be alleviated using t-SNE, that rather models the distances using a Student-t distribution

$$p(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma\frac{\nu}{2}}\left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \tag{2.41}$$

where $\Gamma(x)$ is the Gamma function, and ν is the degree of liberty of the distribution. Student-t distribution is a common conjugate distribution of the normal distribution (see sec 2.3), modeling the uncertainty of its mean vector once the variance has been marginalized out

$$p(\mu|\mathcal{D}) = \int p(\mu|\mathcal{D}, \sigma^2)p(\sigma^2|\mathcal{D})d\sigma^2 \tag{2.42}$$

where $p(\mu|\mathcal{D}, \sigma^2)$ is a normal distribution centred on the approximated mean of the observations $\mathcal{D}$, and $p(\sigma^2|\mathcal{D})$ is a scaled inverse chi-squared distribution. The marginalization of the variance provides an heavier tail than a standard normal distribution, hence giving a higher probability to distant data points. Taking $\nu = 1$, we can thus express the probability density of the distances in 2.39 with [112]

$$q(\mathcal{D}_{ij}^{\mathbf{z}}) = \frac{(1 + \|\mathbf{z}_i - \mathbf{z}_j\|^2)^{-1}}{\sum_{k\neq i}(1 + \|\mathbf{z}_i - \mathbf{z}_j\|^2)^{-1}} \tag{2.43}$$

This formulation of SNE, then called t-SNE, is much more robust to high-dimension input spaces, and is still provides one of the most used techniques for dimensionality visualization. t-SNE will provide the inspiration of the Student-t regularization method, that we will develop section 3.4.

Manifold Learning methods The main flaw of PCA is that it projects the data onto a *linear* sub-space, hence unable to accurately represent dataset that lie on non-linear sub-manifold of the original space. For example, if the data is organized around a curvated surface (such as a sphere), the representation provided by the PCA will project points placed on opposite locations on the same position. Non-linear approaches to PCAs have also been developed, such as Kernel PCA [113], but non-linearities often makes the projection basis non-unique and then intractable with formal calculus.

Some methods propose instead to extract non-linear subspaces using the concept of a *manifold*, that is an embedded surface that is *locally* diffeomorphic to an Euclidean plane of dimensions $d < D$. This manifold can then be retrieved by taking local linear maps taken from the neighbourhood of each point, and then gluing the maps to infer a system

of coordinates. This idea was first investigated by Roweis al &. [114] using local PCAs, and then improved with Laplacian Eigenmaps [115] and Hessian eigenmaps [116], rather using local Eigenmaps to perform linear approximation. Alternatively, Local Tangent Space Alignment aims to find projections that aligns the approximated tangent spaces of each data point, hence flattening the manifold [117].

Another approach rather consists in estimating the intrinsic metrics of the underlying manifold by computing pairwise distances along the manifold's surface. This can been done by constructing a graph representation of the manifold by linking neighbouring data points, such that each vertex is a data point and each edge is the distance between the points. The *geodesics* between two points can then be estimated finding the shortest path linking them, summing the lengths of the crossed edges. This construction is used by the ISOMAP method [118], that proposes to perform a MDS with the estimated geodesics rather than the Euclidan distances, hence preserving the local geometry of the underlying manifold. A similar method is used by Uniform Manifold Approximation and Projection (UMAP) [119], that uses fuzzy topological representations to merge the local projections extracted from the data.

Manifold learning then provides very interesting methods to extract representations from an underlying manifold, helping to capture non-linear axis of variations that would not be represented in linear dimensionality reduction techniques. Furthermore, these methods can provide interesting information about the local structure of the data, finding more natural interpolation between points. However, these methods are often heavy to compute, making them hardly applicable to large and high-dimensional datasets. Additionaly, these methods are non-invertible, and are then unsuitable for generative purposes. However, the underlying intuitions behind these methods still nourishes modern advances of machine learning, as Riemannian approaches to auto-encoding (see section 2.3.2.4).

2.2.2 Neural networks and back-propagation

While performing well with small datasets, the methods described above suffers from two major drawbacks. First, most of them are hardly scalable to lard high-dimensional datasets, as they operate directly on the entire data matrix $\mathbf{x}$. Secondly, they are based on simple geometric constraints that may fail to capture the complexity of the data, limiting the expressiveness of provided representations. These challenges motivated the development of alternative approaches based on *stochastic optimization*, allowing to iteratively train systems on random data subsets. A seminal optimization method, called *stochastic gradient descent* (SGD), allows to train arbitrarily complex models by only requiring their derivability, provided a well-defined training criterion. The development of SGD, simultaneously with the ever-growing computational power of modern computers, enabled the efficient integration of *neural*

networks in machine-learning methods, increasing significantly their performance. In this subsection, we will shortly describe neural networks and how to train them with SGD.

Neural networks. Neural networks (NN), also called *Artificial Neural Networks* (ANN), can be described as trainable computation blocks composed by small units, called *neurons*, that can be linked to model complex functions of arbitrary complexity. Formerly inspired from the cybernetics and cognitive science fields, neural networks now provides the basis for black-box optimization methods used in modern machine learning techniques, from the simplest (simple classification or regression systems) to most complex applications (such as deep learning). As these systems are now very well documented, we will try to give a brief but intuitive explanation of these systems.

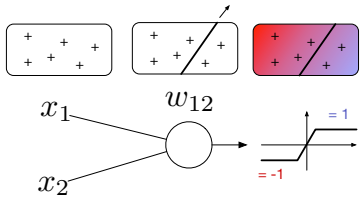


Figure 2.20: A single neuron, by drawing a straight line in a multi-dimensional space, is able to split it two parts. Adding a non-linearity allows, a hard tanh in the above example, to define a function that equals -1 if far above the line, and +1 far below.

A *neuron* is a unit composed by a linear or affine function $\mathbf{y} = f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$, then corresponding to a linear subplane of the input space. A neuron is also generally followed by a non-linearity, that will help him to use this segmentation to model non-linear functions of the input space. For example, taking a sigmoid activation $\sigma(x) = 1/(1 + e^x)$ (also called logistic as it equals 0 when $x \to -\infty$ and 1 when $x \to \infty$), we are able to split the space in two, where the function equals 1 on the left of the line and -1 on right of the line, such that the line drawn act as a *decision boundary* (fig 2.20).

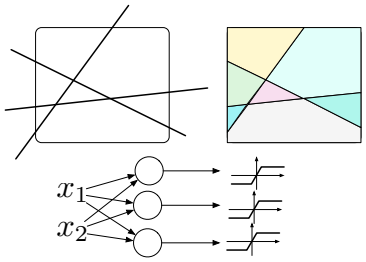


Figure 2.21: Several neurons with classification units can perform complex segmentation of the input space, and can then perform multi-class classification with sigmoid units.

To give an intuition about the inner working of neural networks, we will present a modern formulation of the *perceptron*, one of the oldest supervised machine learning system. Taking d neurons, we are then able to simulate functions $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^d$. Using sigmoid non-linearities, the system is then able to split the input space in several regions, and then to discriminate up to d classes (see fig. 2.21). However, such classification can only work if the target classes are linearly separable. We can alleviate this problem by adding a second layer of neurons, connected to the first, that can increase the modeling capacity of the network. Indeed, if the number of neurons of the hidden layer H is lower than the input dimension, the neural network is called a *bottleneck*, as it reduces the dimensionality of the input. Reversely, if the number of hidden neurons is higher, the network is able to discriminate non-linearly separable regions of the input space, as it is embedded in a space of higher dimension (see fig 2.22) for an example with the well-known XOR example). Such networks are then called *fully-connected networks*, or *multi-layer perceptron* (MLP). However, the choice of a sigmoid activation is not optimal for the hidden neurons, as it bounds the output between $\sigma(x) \in [0; 1]$, and then can weaken the expressiveness of the hidden representation. A common alternative choice is then the *Rectifier Linear Unit* (ReLU), that can be understood as a way to bypass the left part of the neuron's subplane while not affecting the other. Hence, multi-layer perceptrons can be understood as systems that embedding the input space to a hidden representations, that can be trained as an intermediary space used for binary-,

multi-classification, or regression.

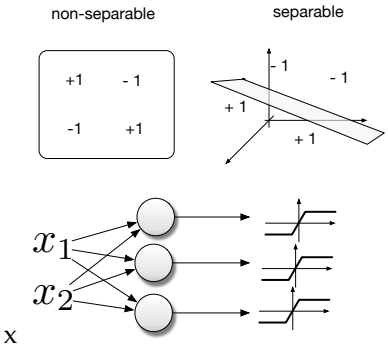


Figure 2.22: The well-known XOR example : a problem that is non-separable in an input space can become separable with additional dimensions.

Backpropagation and gradient descent A neural network can thus be understood as a parametric non-linear function $\mathbf{y} = \mathtt{NN}_\theta(\mathbf{x})$ with an arbitrary number of inputs and outputs, that has to be trained on given pairs $(\mathbf{x}, \mathbf{y})$ to model the targeted function. Neural networks then requires the definition of a *loss function*, that has to be accurately designed to fulfill a given task. By example, in a supervised classification setting, the training dataset is composed by pairs $\{\boldsymbol{x}, \boldsymbol{y}\}_{i=1}^{N}$, where $\boldsymbol{x}$ is the input of the network and $\boldsymbol{y}$ is the corresponding label information. The network is then typically trained on a classification loss function $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ between the network prediction $\hat{\boldsymbol{y}}_i$ and the true label $\boldsymbol{y}_i$, as a cross-entropy loss.

However, given a neural network with parameters w_{ij} and a suitable loss function $\mathcal{L}(\mathbf{y})$, we still need an optimization method to train the network towards the desired output. *Back-propagation* algorithms allow us to derivate the gradient $\frac{\partial \mathcal{L}}{\partial w_{ij}}$ of each weight of the network with respect to the loss function $\mathcal{L}(\mathbf{y})$, thus assuming the full derivability of the system *. Back-propagation is based on a recursive *chain rule* that is applied on the full network, as can be shown fig. 2.23 where a few neurons are shown for simplicity. Once obtained the derivatives $\frac{\partial \mathcal{L}}{\partial w_{ij}}$ for each weight w_{ij}, we can optimize the network using *gradient descent*, an optimization scheme that performs iterative updates of the weights using the corresponding derivative

$$w_{ij} \leftarrow w_{ij} - \nu \frac{\partial \mathcal{L}}{\partial w_{ij}} \tag{2.44}$$

where the *step size* ν is generally shrunk during training. Indeed, if $\frac{\partial \mathcal{L}}{\partial w_{ij}} > 0$, an increase of w_{ij} will lower the loss $\mathcal{L}$, and reversely, if $\frac{\partial \mathcal{L}}{\partial w_{ij}} < 0$, decreasing w_{ij} will decrease $\mathcal{L}$. This procedure is ithen terated until convergence $\mathcal{L} \to 0$, or according to a diverse stopping policies. While the optimization was initially performed on the overall dataset, this procedure can be very time- and memory-consuming for large amounts of examples, then limiting the scalability of the method. An alternative optimization scheme, called *stochastic gradient descent* (SGD), rather performs iterations over a small batch of examples chosen randomly in the dataset [120]. This optimization scheme has the same convergence guarantees as the original gradient descent but is much more scalable, as it does not imply to perform back-propagation over the full dataset. Furthermore, it has been shown that the stochastic nature of SGD increased the robustness of the trained model, preventing the system to suffer severe over-fitting. Since the creation of SGD, gradient descent has been a main topic in the optimization field, and still provides the basis of advanced methods such as Adagrad [121], RMSProp[122], or ADAM [123], just to cite the most recent ones.

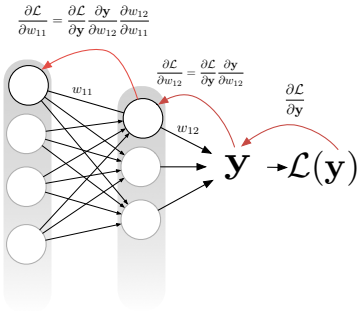


Figure 2.23: Schema of gradient back-propagation through two neuron layers, using the chain rule to obtain the derivative of the loss function with respect to a single neuron weight.

* this hypothesis is not always respected, as for example with the systems using ReLU activation. To this end, the recently proposed ELU activation is often used instead of ReLU, because of its derivability.

Theoretic guarantees of neural networks. Despite of the conceptual simplicity and the efficiency of such *connectionist* approaches, the lack of interpretability of the network's parameters and their high over-fitting potential, due to their capacity, often provoked the mistrust of statisticians. Indeed, neural networks can be dauntingly hard to approach with analytical methods, especially with the latter raise of *deep learning* that could use up to hundreds of stacked hidden layers (hence called *Deep Neural Networks*, or DNN). Furthermore, some experiments showed that such models can also trivially record the data, rather than extracting robust features from it. The over-fitting tendency of deep neural networks is generally alleviated by learning on very large datasets, providing efficient yet greedy models. Therefore, the expressiveness provided by these methods can also be part of their weaknesses, and then requires a particular attention on the obtained results.

Studying the intrinsic behavior of neural networks is thus very challenging. First theoretical guarantees investigated by the researchers focused on the approximation capacity of these systems, and have been widely studied since the 90s. A seminal universal approximation theorem, proposed by Hornik, proved the existence of a single hidden layer MLP that can approximate arbitrary well any function defined on some compact set [124–126], independently to the chosen activation and input dimension. However, this theorem does not state about the number of hidden units required, or about the training procedure. Recently, Lu & al. proved that any function could be approximated with a width $d \neq n+4$, regardless of the depth [127]. Hanin showed that it rather required $d \leq n+1$, and provided a bound on the approximation error depending on the network depth's [128]. While these theorems evaluate the approximation abilities based on an infinite amount of input examples, Zhang & al. rather study the approximation capacity of neural networks on finite samples, showing the existence of a two-layer neural network with ReLU activations and $2n+d$ weights that can represent any function on a sample of size n in d dimensions.
Other methods rather foster the expressiveness, i.e. is the diversity of functions a network can model, and how this diversity evolves with the network's depth. By example, Eldan & al. showed that 3-layers neural networks were able to express functions that a 2-layer network could not model [129]. Topological analysis of NNs can also provide interesting results, as shown by Bianchini & al. who proved that topological capacity of a network increased with its depth [130]., or by Montufar & al. expressing the number of maximum linear boundaries of a MLP with arbitrary depth and ReLU non-linearities [131].

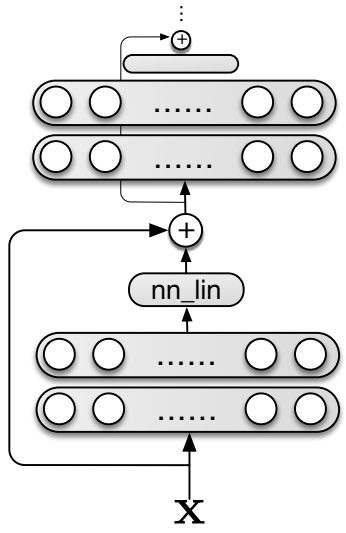


Figure 2.24: Example of *residual* network, adding skip connections to inject gradient information in deeper layers

While approximation theorems aim to provide theoretical basis on the modeling capacity of neural networks, some study rather focus on their learning dynamics. By example, the analytical study of information propagation in NN underlines a duality between forward and backward passes [132], preventing to perform SGD in very deep fully-connected networks (then justifying *residual networks*, that add skip-connections between layers [133] as shown fig. 2.24). DNNs have

also been demonstrated to fail learning simple low cross-predictability data-label pairs with SGD, underlining the impact of the chosen optimization method [134].
Due to the intractability of neural networks, some authors proposed to develop black-box analysis methods to gain some insights on the neural networks' internal behavior. *Saliency maps*, for example, estimate the sensibility of each network's weight with respect to a given input dimension based on its derivative [135–137]. Alternatively, some information-theory grounded methods were proposed to study information propagation in DNNs across layers. However, as neural networks are deterministic, mutual information between layers can be whether infinite or constant. Some estimators can ye bet derived by injecting little amount of noise in the layers' output, and allows to demonstrate interesting compression abilities as the number of layers increase [138]. The information bottleneck framework can also be leveraged to study neural networks in the case of supervised learning, showing that during the train process the system first reduces the empirical error of the task (empirical error minimization), then compresses its representation [139]. However, such effect has been mitigated by Saxe & al., arguing that this effect is dependent of the used non-linearity [140].

Regarding the generalization abilities of DNNs, most validation procedures are based the evaluation of the network's performance on test data. However, while over-fitting is an admitted tendency of DNNs, some authors investigate why DNNs are not over-fitting much more, unveiling an implicit regularization property pushing them to non-degenerated solutions [141]. While the compression effect demonstrated by Tishby & al. could be held responsible, compression can not be generally held as being related to generalization [140]. This regularization is actually performed by the SGD algorithm, as averaging gradient among batches of data naturally flattens the error landscape [142]. Furthermore, it can be shown that with SGD the number of training examples has much more impact on the generalization than the model size, explaining why taking very large architectures does not automatically yield to pathological over-fitting. Despite these theoretical guarantees generalization of DNNs may be enforced by external methods, preventing the networks' to converge on degenerated solution. Similarly to classical regression approaches, weight-normalization of the weights can be used to prevent over-fitting. Alternatively, Dropout is a widely used method for improving generalization, consisting in randomly zeroing hidden dimensions during training to push covariances between networks' weight [143]. However, recent analysis show that such techniques decrease the capacity of the model, while gaining too little when compared to data augmentation [144]. Finally, weights of neural networks can also be considered as random variables, and then be trained with Bayesian inference [145], or even be trained as Gaussian Processes [146]. Despite the real generalization gain brought by Bayesian techniques for neural networks, they are barely used in the actual machine-learning field because of their low scalability and their high computational cost.

2.2.3 Auto-encoders as invertible representation extractors

Most of machine learning techniques using neural networks are trained on explicit tasks, providing efficient methods for regression, classification, or reinforcement learning. As we saw in the previous sections, deep neural networks can be understood as feature extractors whose layers are progressively transforming the input into a suitable space, whose emerging topology is trained to represent the targeted task. However, the retrieved representations are generally task-dependent, and are not enforced to retrieve structural information about the data itself. Auto-encoders provide an alternative to such supervised methods, extracting a representation from the data without the definition of any external task, but rather trained with a *reconstruction* objective. Auto-encoders can then be understood as invertible representation extractors, based on the joint learning of two processes : an *encoding* process, projecting the data in the representation, and the *decoding* process, that projects back the representation in the data space. As we will see, representations provided by standard auto-encoders are often quite close to the ones obtained with PCAs ; however, the development probabilistic formulations of auto-encoding allowed the formulation perform black-box manifold learning techniques, and opens the way to the Auto-Encoding Variational Bayes that grounds this PhD.

Auto-encoders. The original auto-encoders, proposed by Rumelhart & al., are built from two distinct 1-layer sigmoid networks : an *encoder* that transforms the input $\mathbf{x} \in \mathbb{R}^D$ into a code $\mathbf{z} \in \mathbb{R}^d$, and a *decoder*, that transforms the output $\mathbf{z}$ back to the data $\mathbf{x}$ [147] (see fig. 2.25)

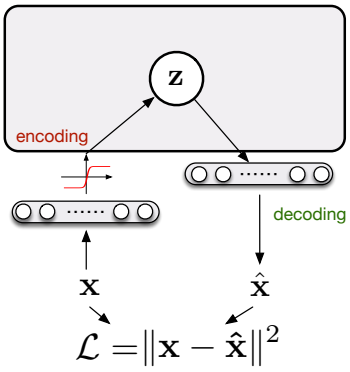


Figure 2.25: Graph of an auto-encoder, that is built from two parts : an *encoding* function, generally a neural network followed by a sigmoid non-linearity, and a *decoding* function, that inverts the representation to recover the corresponding data. The system is then trained using an ℓ^2 loss between the original example and its reconstruction.

$$\mathbf{z} = \sigma(\mathbf{A}\mathbf{x} + \mathbf{c}) \tag{2.45}$$

$$\mathbf{x} = \sigma(\mathbf{B}\mathbf{x} + \mathbf{d}) \tag{2.46}$$

where the matrices $\mathbf{A}$ and $\mathbf{B}$ are trained with gradient descent on the reconstruction error $\|\mathbf{x}, \mathbf{x}'\|^2$. Without the sigmoid non-linearity, and provided that $d < D$, auto-encoders converge to PCA, as shown by Baldi & al. [148]. Auto-encoders can then be assumed to perform somehow non-linear PCA, yet being trained by gradient descent. Moreover, the system can be trained using alternative objective functions such as cross-entropy or $\|\cdot\|_1$ to provide different solutions, showing the flexibility of the method.

Auto-encoders thus allow *invertible* representation leaning, both for encoding an decoding functions $f(\mathbf{x})$ and $g(\mathbf{x})$ are trained by gradient descent. As the choice of d is arbitrary, we can distinguish two separate cases. Taking $d \gg D$ provides an *over-complete* representation, embedding the data in a larger space where projections can be sparse. Sparsity can be used for example to linearly separate data points that would not be separable in the input space [149]. Reversely, when $d \ll D$ the representation is said *under-complete*, and the latent code $\mathbf{z}$ can then be seen as a *compressed* representation of $\mathbf{x}$. In this

case, the auto-encoders weights are forced to share information among several inputs, entangling the representation. Though, as we are here interested by low-dimensional representations, we will only consider the second case.

Manifold learning with auto-encoders While auto-encoding is an efficient method to learn representations from an unsupervised manner, the extracted features are not very robust and can be very sensitive to small changes of data, providing degenerated reconstructions [150]. Furthermore, as the representation is only trained on the projection points of the input data, a direct exploration of the space often produces unrealistic samples, such that the obtained representation is unusable for generation purposes. Hence, Vincent & al. proposed to reinforce the extracted features by adding noise to the input data, learning the auto-encoder to denoise the input data and then to be less sensitive to small perturbations [151]. *Denoising auto-encoders* (DAE) may then be understood as a stochastic operator $p(\mathbf{x}|\tilde{\mathbf{x}})$, that learns to project low-probability noisy observations $\mathbf{x}_\epsilon = \mathbf{x} + \epsilon$ to the closer high-probability point (see figure 2.26). Therefore, DAEs is inherently extracting the true underlying manifold $p(\mathbf{x})$ by learning the transition operator $\mathbf{x} + \epsilon \rightarrow \mathbf{x}$ [152]. Alain & al. showed that if $\epsilon \sim \mathcal{N}(0, \mathbf{œ}^2\mathbb{1})$, the total loss amounts [153]

$$\mathcal{L}_{DAE}(r_\sigma(\mathbf{x})) = \mathbb{E}\Big[\|r_\sigma(\mathbf{x}) - \mathbf{x}\|_2^2 + \sigma^2 \|\frac{r_\sigma(\mathbf{x})}{\partial \mathbf{x}}\|_F^2\Big] + o(\sigma^2)$$

showing that adding a noise of variance σ^2 penalizes the first-order derivative of the reconstruction function $r_\sigma(\mathbf{x})$ with respect to x, then also matching also the tangent space of the manifold $p(\mathbf{x})$. This result caused a surge of interest in auto-encoding systems, as the DAEs demonstrated some abilities to catch the inner data manifold $p(\mathbf{x})$. However, DAEs were shown to be robust facing only one type of noise, then limiting the efficiency of the approach. Similarly, *contractive auto-encoders* propose to penalize the input sensitivity of the representation by penalizing the Jacobian of the encoder's output [154]

$$\mathcal{L}_{CAE} = \|r(\mathbf{x}) - \mathbf{x}\|_2^2 + \sum_{i=0,j=0}^{D,d} \frac{\partial \mathbf{z}_j(\mathbf{x})}{\partial \mathbf{x}_i}$$

where the regularization of the Jacobian enforces the representation to be contracted, and to "whiten" its axis of variations to provide parsimonious projections. The emerging properties of DAEs and CAEs provoked a regain of popularity in auto-encoders, showing that these methods could be used for black-box manifold learning. While auto-encoders were initially deterministic problems, probabilistic formulations first raised by Renzato & al. and then by Vincent & al. allowed to turn latent *codes* to latent *spaces*, where the full span of the latent space $\mathbf{z}$ is used. Furthermore, as the extracted representation were shown being closer to the real underlying manifold of $p(\mathbf{x})$, and then significantly increasing its consistency, the invertibility of the representation

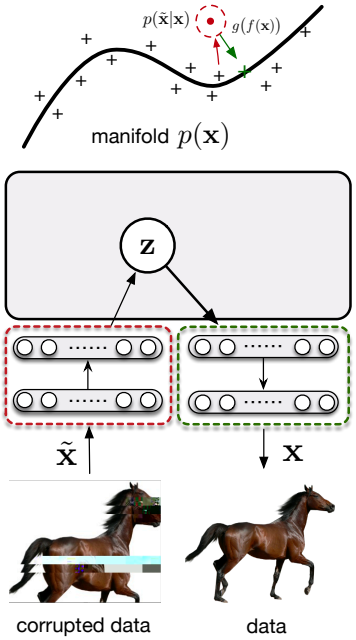


Figure 2.26: The denoising auto-encoder, by adding a noise with fixed variance to the input, enforces the auto-encoder to learn how to project a neighboring zone of the data manifold to its correct reconstruction.

allowed to directly explore the extracted manifold and then turn the DAE into a proper *generative model*.

2.3 Bridging representation learning and variational inference

In the previous sections, we summarized two different trends for extracting useful representations from signals. The first, grounded on Bayesian learning, formulates a generative distribution $p_\theta(\mathbf{x}|\mathbf{z})$ and, based on some prior information $p(\mathbf{z})$, is able to extract a posterior distribution $p(\mathbf{z}|\mathbf{x})$ in order to infer the latent variables $\mathbf{z}$ corresponding to a given data $\mathbf{x}$. The second, grounded on representation learning, proposes to model functions $\mathbf{z} = f(\mathbf{x})$ whether by extracting some structural information from the data, or based on an explicitly defined loss. While the first provide robust and interpretable representations, it requires the tractability of involved terms and then strongly limits the expressiveness of modeled distributions. Conversely, machine-learning methods using neural networks are able to model functions or arbitrary complexity, and can be trained on any derivable loss functions ; however, they do not ensure any representational consistency except the one indicated by the task, and can suffer from strong overfitting. Furthermore, the obtained representations are generally not invertible, preventing to use the obtained systems as generative models.
Auto-encoders, presented above, seemed to hint at an interesting way to combine both of best worlds. Indeed, this system jointly trains encoding and decoding functions, allowing to train invertible representations using neural networks, and then enabling their use as generative models. We could then take inspiration from auto-encoding systems but rather use neural networks to model a generative distribution $p(\mathbf{z}|\mathbf{x})$ as an decoding process, and another network to model the variational distribution $q(\mathbf{z}|\mathbf{x})$ as an encoding process. This would allow us to learn expressive relations between the data $\mathbf{x}$ and the latent variables $\mathbf{z}$, while preserving the robustness and the invertibility of Bayesian inference. This framework, called Auto-Encoding Variational Bayes (AEVB), will then be introduced in the next sections.

2.3.1 Auto-Encoding Variational Bayes

Stochastic optimization of ELBO Reminding the ELBO (2.18), the variational lower-bound for the data evidence $p(\mathbf{x})$ can be written

$$\mathcal{L}(q) = \int q(z) \log \frac{p(\boldsymbol{x}, z)}{q(z)} dz \tag{2.47}$$

$$= \mathbb{E}_q[\log p(\boldsymbol{x}, z) - \log q(z|\boldsymbol{\lambda})] \tag{2.48}$$

where each latent parameters z_j is optimized in turn to maximize the term $\mathcal{L}(q)$. However, this approach is still assuming the tractability of (2.18) for each $q(z_i)$, still preventing to optimize the ELBO with

intractable variational distributions. In the previous section, we introduced SGD, that allows to train a parametric model with respect to a defined criterion using iterative gradient descent optimization. As the ELBO (2.18) provides a lower-bound of the model evidence $p(\mathbf{x})$, it could then be used as a training criterion for SGD optimization. However, contrary to the non-stochastic cases introduced above, here the involved quantities are distributions, such that the derivation must be obtained respectively to the distribution parameters. This framework, called *Stochastic Variational Inference* (SVI), were first proposed by Hoffman & al. for exponential conjugate priors, successively updating the ELBO gradient with respect to the *natural parameters* of exponential families [155]. Another approach is black-box variational inference [156], that allows to train a variational distribution $q(\mathbf{z}|\boldsymbol{\lambda})$ given an arbitrary generative model $p(\mathbf{x}|\mathbf{z})$

$$\nabla_\theta \mathcal{L}(q) = \nabla_\theta \mathbb{E}_q[(\log p(\boldsymbol{x}, z) - \log q(z|\boldsymbol{\lambda}))] \tag{2.49}$$

where λ are the parameters of the distribution $q(z; \boldsymbol{\lambda})$. As the involved expectation is generally intractable, we can estimate it with *Monte-Carlo approximation*, a simple sampling method drawing L samples from a distribution and then averaging the outputs obtained with a function $f(\cdot)$

$$\nabla_\theta \mathbb{E}_{q_\theta(z)}[f(z)] = \frac{1}{L}\sum_{l=1}^{L} f(z)\nabla_\theta q_\theta^l(z^l) \tag{2.50}$$

this estimator, called REINFORCE, is unbiased as $l \to +\infty$. However, while its generality makes it applicable to a wide variety of models, its efficiency is awfully reduced by its high variance, dauntingly slowing down the training convergence. Several methods can then be used to reduce this slowdown, such as *Rao-Blackwellization* [157] and *control-variates* [158]. However, these methods are not sufficient to perform SVI with high-capacity neural networks, and then limits the efficiency of the variational approximation.

Reparametrization trick & Variational Auto-Encoder Despite the advances brought by SVI, that allows to perform variational inference with black-box approximation, the variance of the ELBO estimator (2.49) makes it inapplicable for complex cases. This problem can be alleviated under some mild assumptions for the chosen variational family $\mathbb{Q}$ thanks to the *reparametrization* of the network's gradient. Indeed, provided that sampling from $q_\phi(z|\boldsymbol{x})$ can be split between a deterministic part and stochastic component $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$, such that $g(\boldsymbol{x}, \boldsymbol{\epsilon})$, the expectation in (2.49) can performed with respect to $p(\boldsymbol{\epsilon})$ instead of $\mathbb{E}[q_\phi(\boldsymbol{z}|\boldsymbol{x})]$ thanks to the *Law Of Unconscious Statistician* (LOTUS)

$$\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[f(z))] = \mathbb{E}_{p(\boldsymbol{\epsilon})}[f(g_\phi(\boldsymbol{\epsilon}, \boldsymbol{x}))]$$

so the gradient of the deterministic part is separated from the gradient of the stochastic part (see fig. 2.27). This method significantly reduces the variance of the ELBO, and then fastens its convergence rate. This

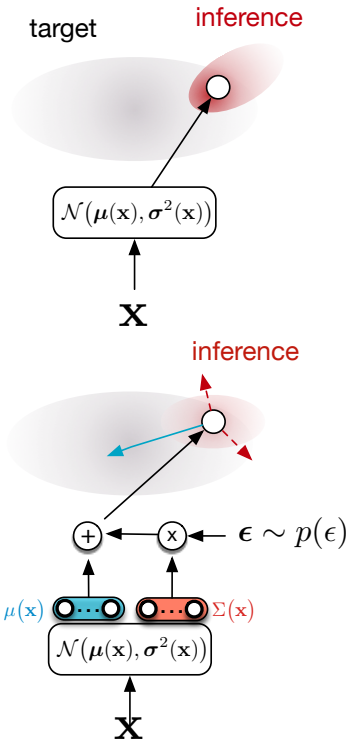


Figure 2.27: The reparametrization of samples allows to separate the gradient coming from the stochastic and deterministic components.

reparametrization trick available for many distributions such as Normal, Logistic, Uniform, and Laplace distributions, but also to composition of these, such as Log-Normal, Dirichlet, Beta, and many others. More generally, it can also be applied to distributions with tractable inverse Cumulative Distribution Function, extending this trick to Exponential, Cauchy, and Gumbel distributions. This application of this trick to ELBO estimation was first proposed by Kingma & al., allowing a scalable and fast method for black-box amortized Variational Bayes with arbitrary encoding / decoding functions [22]. A seminal example of this method, called the *Variational Auto-Encoder* (VAE), is a particular parametrization of this system where both generative distributions $p(\boldsymbol{x}|z)$ and $q(z|\boldsymbol{x})$ are defined as mean-field normal distributions (see fig. 2.28)

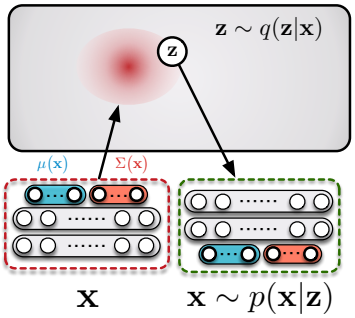


Figure 2.28: The *variational auto-encoder*, instead of defining a point-estimate of data $\mathbf{x}$ and latent code $\mathbf{z}$, rather define the parameter of the generative and variational distributions.

$$p_\theta(\boldsymbol{x}|z) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_\theta(z), \boldsymbol{\sigma}^2_\theta(z)) \tag{2.51}$$
$$q_\phi(z|\boldsymbol{x}) = \mathcal{N}(z|\boldsymbol{\mu}_\phi(\boldsymbol{x}), \boldsymbol{\sigma}^2_\phi(\boldsymbol{x})) \tag{2.52}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are respectively the parameters of generative and variational models, both defined with neural networks. As outlined by the name, VAE then bridges Bayesian inference with auto-encoders, and are optimized with SGD on the ELBO (2.18), that can be re-written from an auto-encoder perspective

$$\mathcal{L}[q] = \underbrace{\mathbb{E}_{q(z)}[\log p(\boldsymbol{x}|z)]}_{\text{reconstruction error}} - \underbrace{D_{KL}[q(z|\boldsymbol{\lambda})\|p(z)]}_{\text{regularization error}}$$

The first term can be understood as a reconstruction error, where the log-likelihood replaces the ℓ^2 loss, while the second is a regularization term, preventing the variational model to move away from the prior $p(\mathbf{z})$. VAEs can then be formulated as a stochastic version of the auto-encoder, with an additional regularization term that enforces the approximated posterior distribution to match the prior. Variational auto-encoders raised a significant enthusiasm in the machine learning community, because of its efficient trade between simplicity, scalability, and flexibility. Indeed, the possibility of modeling the dependencies between data $\mathbf{x}$ and latent variables $\mathbf{z}$ with neural networks greatly enhance the expressiveness of Bayesian latent models, relaxing almost all of the previous tractability assumptions. Also, VAEs gain from auto-encoders by inferring its latent representation with a Bayesian learning process, increasing the generalization of the model and the robustness of the representation. Furthermore, representations obtained with variational auto-encoders on simple datasets were surprisingly coherent, pointing out that the VAE successfully retrieved the data underlying manifold (see fig. for an example with the MNIST dataset [159]). These appealing properties turned VAE as one of the most popular recent machine-learning generative model, and were then consequently widely investigated by the community.

2.3.2 Emerging properties of variational auto-encoders

Since the seminal works of Kingma & Welling [22] and Rezende & al.[160], emergent properties of AEVB have then been extensively studied in the domain of probabilistic generative models field. Despite the surprising consistency of the extracted representations, researchers unveiled two main drawbacks of VAE, preventing these models to be extended to datasets of higher complexity. The first issue was the blurriness of generated samples, pointing out the sub-optimality of the generation process. A second issue was that, in some cases, the latent spaces extracted by VAEs were degenerated, indicating that the latent space was bypassed by the model [161]. In this section we will review the main sources of degeneracy that occur when training variational auto-encoders, and provide a state of the art of current solutions and improvements brought so far by the research community.

Approximation vs amortization gaps. Despite the gain of expressiveness brought by parametrizing $q(\boldsymbol{x}|z)$ with neural networks, the variational model distribution is still defined as normal distribution with diagonal covariance, then still belonging to the mean-field family. Choosing a given family of distributions $\mathbb{Q}$ for our variational model, we can derivate the ELBO 2.18 to outline two possible sources of sub-optimality [162] (see fig.2.29)

$$\begin{aligned}\mathcal{L} &= \log p(\boldsymbol{x}) - \mathcal{L}[q] \\ &= \log p(\boldsymbol{x}) - \mathcal{L}[q^\star] + \mathcal{L}[q^\star] - \mathcal{L}[q] \\ &= \underbrace{D_{KL}[q^\star(z|\boldsymbol{x})||[p(z|\boldsymbol{x})]}_{\text{approximation gap}} + \underbrace{D_{KL}[q(z|\boldsymbol{x})||[p^\star(z|\boldsymbol{x})] - D_{KL}[q^\star(z|\boldsymbol{x})||p(z|\boldsymbol{x})]}_{\text{amortization gap}}\end{aligned} \tag{2.53}$$

where $q^\star(z|\boldsymbol{x})\mathbb{Q}$ is the optimal distribution minimizing the ELBO $\mathcal{L}$. This reformulation allows to separate the inference gap in two different parts : an *approximation gap*, that quantifies the loss obtained from choosing a given a non-optimal family of distributions $\mathbb{Q}$, and the *amortization gap*, that quantifies the loss obtained between the actual inference and optimal distributions (respectively q and $q^\star$).
The first term quantifies the loss that occurs when the chosen variational family $\mathbb{Q}$ is unable to approximate the real posterior $p(z|\boldsymbol{x})$ distribution. As mean-field normal distributions are unable to match complex posteriors, the system will not be able to learn the corresponding generative distribution, explaining the blurriness of generated samples. Approximation gap can be reduced by choosing more expressive variational families, or normalizing flows (see sec 2.3.2.1), or rather using implicit variational distribution to match the aggregated posterior $q(\mathbf{z}) = \int q(\mathbf{z}|\mathbf{x})d\mathbf{x}$ (see sec. 2.3.2.3). The second term rather quantifies the optimization error, that can be caused by the Monte-Carlo approximation used to estimate the expected likelihood in (2.18), or imperfect convergence. This gap can be reduced by using better ELBO estimators, such as importance-weighting approximations that provide tighter bounds than AEVB (2.3.2.2).

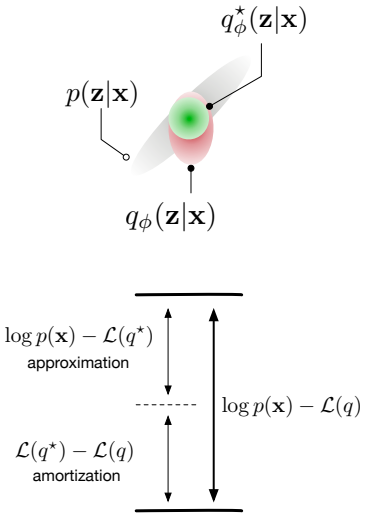


Figure 2.29: The KL-divergence between true and approximated posteriors can divided in two parts : an amortization gap, that represent the gap between the current approximated posterior and the optimal one, and the gap between the optimal approximation and the true posterior.

Latent over-pruning. Another surprising property of AEVB is *latent over-pruning*, an phenomenon observed quite early in the AEVB literature [161, 163, 164]. Over-pruning describes the tendency of VAEs to bypass some latent dimensions during training, over-regularizing the representation by automatically shrinking the number dimensions used by the decoder. While this effect could be thought desirable, pointing out that VAE naturally retrieves the true manifold underlying the data and getting rid of unnecessary dimensions, it was also noticed in degenerated cases, such that the system was not able to generate convincing samples. While the exact origins of latent over-pruning are still unclear, several solutions have been developed to control this effect, allowing diagnosis and efficient training of VAEs.

2.3.2.1 Approximation gap : towards more expressive inference

Alternative inference distributions. A way to reduce the approximation gap is then to choose more flexible variational families $\mathbb{Q}$, that would be able to match the true posteriors more accurately. Even if, theoretically, any inference distribution can be chosen, these distributions must be reparametrizable, samplable, and preferably have a derivable D_{KL}, to be correctly trained . Normal distributions with full-rank covariance matrices $\mathcal{N}(z|\boldsymbol{\mu}, \Sigma)$ can provide an alternative, allowing to model dependencies between latent dimensions. However, low-rank approximation methods have to be used, as sampling from full-rank covariance normal distributions has $\mathcal{O}(d^3)$ complexity. Rezende & al. propose a 1-rank matrix with diagonal correction $\boldsymbol{\Sigma}^{-1} = \boldsymbol{\sigma}^2\mathbb{1} + \boldsymbol{u}\boldsymbol{u}^T$, allowing arbitrary rotations along one principal direction $\boldsymbol{u}$ [32]. Alternatively, the Householder flow was proposed by Tomczak & al. to encode full-rank covariance matrices with a lower computational cost [165]. However, normal distributions with full-rank covariance could not be sufficient, as it only allows to fit the first two moment of the true posterior. Alternatively, the *exponential family* is a generalization of many commonly-used distributions that is specified by its *base measure* $h(\boldsymbol{x})$, *sufficient statistics* $T_i(\boldsymbol{x})$ and natural parameters $\boldsymbol{\nu}$ [59, 166],

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = h(\boldsymbol{x}) \exp \sum_{i=1}^{s} \nu_i(\boldsymbol{\theta}) T_i(\boldsymbol{x}) - Z(\boldsymbol{\theta}) \tag{2.54}$$

where $Z(\boldsymbol{\theta})$ is the partition function of $p(\boldsymbol{x}|\boldsymbol{\theta})$ that normalizes the distribution. The exponential family generalizes a lot of well-known distributions : by example taking the first two moments as sufficient statistics, $T(\boldsymbol{x}) = [\boldsymbol{x}, \boldsymbol{x}\boldsymbol{x}^\top]$, $(2\pi)^{-k/2}$ as base measure, and $[\boldsymbol{\Sigma^{-1}\mu}, -\frac{1}{2}\Sigma^{-1}]$ as natural parameters, provide a reparamterization of the multivariate normal distribution. The exponential family also extend such as Bernoulli, Poisson, Multinomial, Gamma, and all conjugate priors and combinations of these. Exponential families can then provide an interesting framework for generalized inference families, as proposed by Ranganath & al. [59]. Furthermore, exponential families also have a natural connection to *information geometry* [167], with a direct equivalence between its natural parameters and its Fisher information matrix [166].

Another weakness of normal distributions is their unimodality, that is hence unable to fit multi-modal posteriors. In this case, modeling multi-modal posteriors with diagonal normals can provide faulty inference results, worsened by the D_{KL} properties(see sec. 2.20). Some authors thus investigate Bayesian non-parametric methods, that can allow arbitrary information capacity for flexible posterior matching. Serban & al. proposes to define a piecewise parametrization of the latent space, that is divided in equalparts and modeled using a Piecewise Constant Distribution, for both inference and prior distributions. This approach, inspired by non-parameteric Bayesian inference, is then able to shape very generic posteriors, but unfortunatelly grows exponentially with the number of dimensions. Another way to use non-parametric distributions is to model a *Dirichlet process*, that is sampled to obtain the posterior distribution. Dirichlet processes can be built with stick-breaking processes, a sequence of Beta distributions whose parameters can be inferred by the variational model from the data [168], or defined hierarchically using nested Chinese Restaurant Processes, that can allow interesting unsupervised extraction of hierarchical features as shown in some studies in video classification [168].

Normalizing Flows. Choosing an approximated posterior among an extended variational family $\mathbb{Q}$ can indeed provide more expressive inference, but also add a significant computational cost to the training. Alternatively, *Normalizing Flows* (NF) allow to obtain expressive implicit distributions by transforming samples from a simple distribution, enriching the expressiveness of posterior with a little computational cost. Proposed by Tabak-Turner & al., a normalizing flow is a sequence of local invertible transformations $\mathbf{x}_K = F(\mathbf{x}_0) = (f_K \circ ... \circ f_2 \circ f_1)(\mathbf{x})$ that is trained to transform a set of input $\mathbf{x}$ into an output $\mathbf{x}_K$, that we assume drawn from a simple distribution (such as uniform or isotropic normal). The key point of NFs is that, provided the invertibility of maps f_i, the log-likelihood still tractable, so we can train the flow parameters directly on the output space [169, 170]. As normalizing flows do not perform any dimensionality reduction they can be very tedious with high-dimensional data, then outshined by deep learning techniques. However, a recent surge of interest in normalizing flows has been allowed by the increasing capacity of parallel computing, providing very convincing generation results [171].

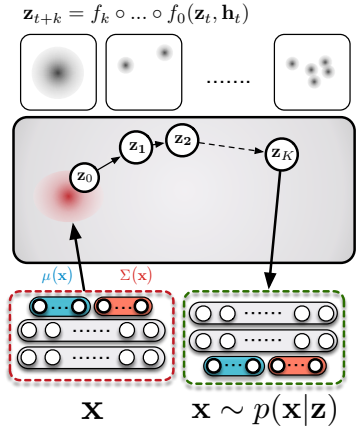


Figure 2.30: Normalizing flows allow to mode complex posterior distributions with a transformation chain from $\mathbf{z}_0$

Turning the original idea upside-down, Rezende & al. propose to take samples drawn from simple distributions, and to transform them into complex ones with normalizing flows [172]. Hence, taking the output of a variational model $\mathbf{z}_0 \sim \mathcal{N}(boldsymbol\mu, boldsymbol\sigma^2)$, we can then use a NF to transform the input latent vector to obtained an implicit posterior distribution, such that $\mathbf{z}_K = F(\mathbf{z}_0)$. As each component of the flow is invertible, the expression of the ELBO 2.18 is still tractable (see fig. 2.30)

$$\log q(\mathbf{z}_K|\mathbf{x}) = \log q(\mathbf{z}_0|\mathbf{x}) - \sum_{k=1}^{K} \log |\det \frac{\partial f_k}{\delta \mathbf{z}_{k-1}}| \tag{2.55}$$

such that NF can be naturally integrated in the ELBO formulation. In addition to its tractability, the significant strength of this approach resides in its simplicity. Indeed, taking a simple planar flow and a isotropic normal prior, we are able to obtain a bi-modal distribution that can be trained to approximate any mixture of two Gaussian. Then, even with little chains of flows $f(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^\top \mathbf{z} + b)$ and simple flow blocks, we are able to approximate complex posterior distributions with a very low computation cost. Since the seminal paper of Rezende, numerous alternative flow blocks have been proposed :*Inverse Auto-Regressive Flows*, proposed by Kingma & al., modeling auto-regressive distributions [173], masked auto-regressive flows [174] , conditional flows [175], or riemannian normalizing flows, using planar flows to regularize the curvature of the latent space with respect to the input [176].

2.3.2.2 Amortization gap : improving variational approximations

As shown by the decomposition (2.53), another source of sub-optimality in AEVB is the *amortization gap*, the loss between the optimal distribution for a given variational family $q^\star(\mathbf{z}|\mathbf{x}) \in \mathbb{Q}$ and the distribution $q(\mathbf{z}|\mathbf{x})$ fixed after training. While traditional variational methods were based on tractable solutions, black-box VI is based on gradient-descent based optimization, such that the obtained solution at convergence is not necessarily optimal. We leave optimization issues due to gradient descent out of this work, as such problems are not specific to AEVB. However, some contributions to the amortization gap can be found in the ELBO itself.r source of sub-optimality in AEVB is the *amortization gap*, the loss between the optimal distribution for a given variational family $q^\star(\mathbf{z}|\mathbf{x}) \in \mathbb{Q}$ and the distribution $q(\mathbf{z}|\mathbf{x})$ fixed after training. While traditional variational methods were based on tractable solutions, black-box VI is based on gradient-descent based optimization, such that the obtained solution at convergence is not necessarily optimal. We leave optimization issues due to gradient descent out of this work, as such problems are not specific to AEVB. However, some contributions to the amortization gap can be found in the ELBO itself. One of the main source of amortization error in AEVB is due to the Monte-Carlo approximation of the expectations over $q(\mathbf{z}|\mathbf{x})$ involved in the ELBO. Remembering the expression of black-box VI gradient estimator (ee sec. 2.49)

$$\begin{aligned}\nabla_\theta \mathcal{L}(q) &= \nabla_\theta \mathbb{E}_q[(\log p(\boldsymbol{x}, z) - \log q(z|\boldsymbol{x}))] \\ &= \nabla_\theta \mathbb{E}_q[\log p(z|\boldsymbol{x})] - D_{KL}[q(z|\boldsymbol{x})||p(z)]\end{aligned}$$

the exact formulation of the gradient estimator is using expectations over $q(\mathbf{z}|\mathbf{x})$, that are generally intractable in the black-box VI framework. We then estimated these expectations with Monte-Carlo approximation

$$\nabla_\theta \mathbb{E}_{q_\theta(\mathbf{z})}[f(\mathbf{z})] = \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{z})\nabla_\theta q_\theta^l(\mathbf{z}^l) \tag{2.56}$$

In practice L is chosen very low, generally taking $L = 1$, such that this approximation is computed using only one sample from the recogni-

tion network. Amortization gap can then theoretically be reduced by taking more samples for the expectation approximation, its precision increasing with L. However, Monte-Carlo approximation performs poorly, and the precision gained by increasing L does not worth the computational cost of decoding L more values.

Importance-Weighted Auto-Encoders A well-known alternative to VAEs are *Importance Weighting Auto-Encoders* (IWAE), that replace the single Monte-Carlo approximation described in previous section by a importance-weighted estimate of the ELBO, taken over several samples from the recognition distribution. The new gradient estimate of the lower-bound $\mathscr{L}_k(q)$ then corresponds to the k-sampled importance weighting of the log-likelihood

$$\begin{aligned}\nabla_{\theta,\phi}\mathscr{L}_k[q] &= \mathbb{E}_{\mathbf{z}_1\dots\mathbf{z}_n\sim q(\mathbf{z}|\mathbf{x})}\Big[\log\frac{1}{k}\sum_{i=1}^{k}\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}\Big] \\ &= \nabla_{\theta,\phi}\mathbb{E}_{\epsilon_1\dots\epsilon_k}\Big[\sum_{i=1}^{k}\tilde{w}_k\nabla_{\theta,\phi}\log w(\boldsymbol{x},\boldsymbol{z}(\boldsymbol{x},\boldsymbol{\epsilon}))\Big] \\ w_i &= p(\boldsymbol{x},\boldsymbol{z}(\boldsymbol{x},\boldsymbol{\epsilon}_i))/q(\boldsymbol{z}(\boldsymbol{x},\boldsymbol{\epsilon}_i)|\mathbf{x})\end{aligned}$$

where we applied the reparametrization trick to obtain k stochastic components $[\boldsymbol{\epsilon}_1\dots\boldsymbol{\epsilon}_k]$. In the latter expression, w_i denotes the density ratios, and the weight $\tilde{w}_i = w_i/\sum_i^k w_k$ is the importance weight of the corresponding sample. Importance-weighting approximation of the ELBO improves the quality of the reconstructions, and at the same time can help to prevent latent over-pruning (see section 2.3.2.3). Cremer & al. argue that the true recognition model recovered by IWAE is slightly different the one recovered by the VAE, as it takes into account the real posterior $p(\mathbf{z}|\mathbf{x})$ [177]

$$q_{IW} = \mathbb{E}_{\mathbf{z}_2,\dots,\mathbf{z}_k\sim q(\mathbf{z}|\mathbf{x})}\left[\frac{p(\mathbf{x},\mathbf{z})}{\frac{1}{k}\left(\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}+\sum_{j=2}^{k}\frac{p(\mathbf{x},\mathbf{z}_j)}{q(\mathbf{z}_j|\mathbf{x})}\right)}\right]$$

where the additional term brought by IWAE is $\sum_{j=2}^{k}p(\mathbf{x},\mathbf{z}_j)/q(\mathbf{z}_j|\mathbf{x})$. Such methods have then be generalized under the name of *Monte-Carlo objectives* for global log-likelihood estimators [178], giving a series of high-precision estimators such as Filtering Variational Objectives [179] for sequential variational data.

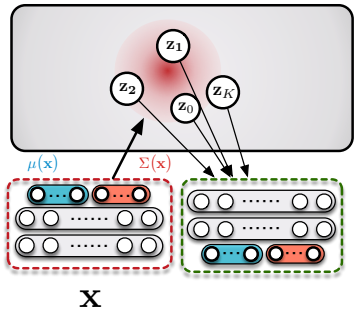


Figure 2.31: Importance weighting samples several samples from the distribution $q(\mathbf{z}|\mathbf{x})$, that are then weighted by their probability in the final loss.

2.3.2.3 Latent over-pruning and posterior collapse

One of the main reasons of the AEVB success in the machine learning literature is the emerging structural properties of the extracted representations [22]. According to manifold hypothesis, this tendency would point out that VAEs are able to retrieve the underlying manifold of a given dataset, and recovering the true factors of variation structuring the data. Furthermore, a natural *latent pruning* effect has been noticed during training, pointing out that VAEs automatically prunes latent dimensions that are considered unnecessary for data

representation. This would indicate that the VAE retrieves the intrinsic dimensionality of the data manifold, and would then structurally protected against over-fitting. However, further investigations of AEVB training dynamics tempered these observations, exhibiting undesirable effects occurring during the training process and mitigating the natural *disentanglement* nature of the extracted manifold.

Latent over-pruning While dimensional pruning can be taken as a desirable property of the system, VAEs has been showed to deactivate latent dimensions even when the reconstruction quality was insufficient, *over-pruning* the latent representation. This deficiency can be intuited recalling the formulation (2.18) of the ELBO

$$p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]$$

observing that the distribution $q_\phi(\mathbf{z}|\mathbf{x})$ is pushed to match the overall prior $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ *for each sample* $\mathbf{x}$ of the data by the regularization term. The two terms are then antagonistic, as the reconstruction term will shrink the latent projections to atomic points (similarly to regular auto-encoders), while the regularization term enforces each data point $\mathbf{x}$ to cover the whole support of the representation (see section 2.1.4.2). This behavior is due to the amortization of variational inference, that from one hand confers strong generalization properties to the system, and from the other hand can also *over-regularize* some dimensions, pushing some axis of the posterior $q(\mathbf{z}|\mathbf{x})$ to match the prior and then being deactivated during the rest of the training process. This effect, called *posterior collapse*, has been clearly highlighted, as can be shown whether by studying the decoder weights [180] or by analyzing statistics from latent projections[164, 181]. As shown by Yeung & al., *active* dimensions have typically tiny variances in average and a strong variance in the distributions' means, while *inactive* dimensions are almost perfectly fitted to the prior, projecting means near to zero and setting the variances to 1. Such regime is then called *polarized*, as the distinction between active and inactive dimensions is significant [182]. This observation clearly demonstrates the double bind of the ELBO, where the inactive units are the dimensions matching the prior.

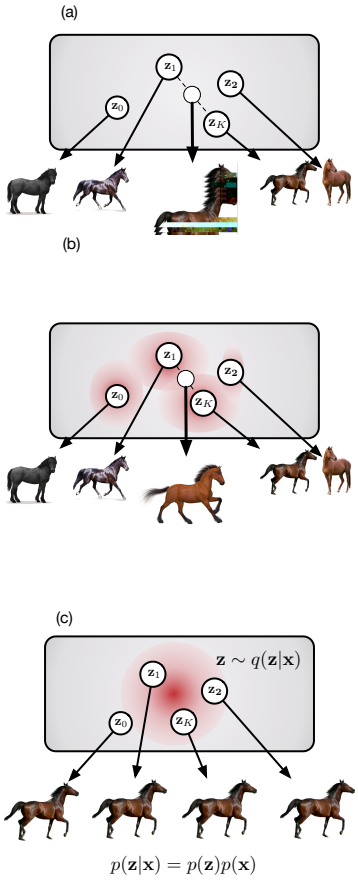


Figure 2.32: (a) an auto-encoder associates to a code $\mathbf{z}$ to a given example $\mathbf{x}$, but is not trained between the examples. (b) a well-balances VAE enforces the distributions to overlap, being able to find *mixtures* between the two distributions (c) if the recognition model $q(\mathbf{z}|\mathbf{x})$ matches the prior, the variables $\mathbf{z}$ and $\mathbf{x}$ are independent.

Bits-back coding Latent over-pruning is particularly sensible to decoder's capacity. Bowman indeed noticed that, when using an auto-regressive decoder, the information given by the latent space was totally bypassed, such that almost every dimension was matching the prior [161]. Chen & al. proposed to explain this strange effect using a *bits-back coding* interpretation of the ELBO (2.14). Remembering the formulation of Bit-Back code length (see section 2.2.3)

$$\begin{aligned}\mathscr{C}_{\text{BitsBack})} &= \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{z}) - log p(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{H}[p(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\big[D_{KL}[q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}|\mathbf{x})]\big]\end{aligned}$$

the extra-cost is then $D_{KL}[q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}|\mathbf{x})$, that can be easily avoided by the system by taking $p(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$, such that $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$. This

principle, called *information preference*, is particularly strong at the beginning of the training, as the encoder can fit much faster the prior $p(\mathbf{z})$ than the decoder can fit the data $p(\mathbf{x})$, especially if the decoder has high capacity and thus train more slowly. This *information preference* property of the VAE thus weakens the representation learning process if the decoder has too much capacity, and then somehow weakens the adaptability of the model.

D_{KL} scheduling and data selection. Different approaches have been developed to counter over-pruning in VAEs. The most common method to prevent this effect is to schedule the weight of regularization term $D_{KL}[q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}|\mathbf{x})]$, initiated at 0 and raising to 1 during training [183]. This *warm-up* procedure allows to prevent hasty posterior collapses, especially in hierarchical models where higher latent layers can converge really quickly to degenerated solutions [184]. Warm-up is a simple and efficient way of avoiding latent over-pruning at the beginning of the training, but unfortunately does not prevent it once the regularization term is weighted as usual. Another solution if to threshold the regularization term, such as the *free bits* constraint proposed by Kingma & al. [173] that replaces the original D_{KL} by a clamped regularization term $\max\{D_{KL}[q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})], \beta\}$, such that the encoding model is forced to not exactly match the posterior.

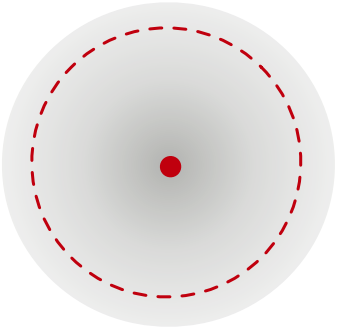
Alternatively, over-pruning can be alleviated by skewing the amount of information given to the decoder, such that the decoder cannot recover the full data $\mathbf{x}$ without $\mathbf{z}$. This idea, called by Chen & al. *explicit information placement*, can be done by limiting the target scope of the generative distribution [185], or to train the encoder on the future of a sequence rather than its past in the case of sequential data [186]. Alternatively, Yeung & al. propose a latent gating mechanism that automatically selects the latent dimensions used for each example [164], enforcing the system to use all of its latent dimensions.

Aggregated posterior matching. Over-pruning occurs mainly because of the term $D_{KL}[q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]$, that tries to fit the approximated posterior distribution to match the prior *for each data sample* $\mathbf{x}$, then over-regularizing the representation $\mathbf{z}$. This can be formally expressed by another decomposition of the ELBO, aiming to exhibit the information loss caused by regularizing $q(\mathbf{z}|\mathbf{x})$ instead of $q(\mathbf{z})$ [187]. In order to express the aggregated posterior $q(\mathbf{z})$ from $q(\mathbf{z}|\mathbf{x})$, Hoffman & al. propose to express the index n as a random variable, that according to training procedures are drawn from a uniform distribution $q(n) = p(n) = 1/N$

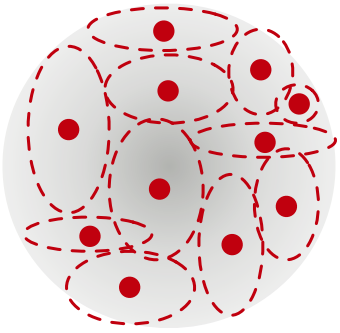
$$\mathscr{L}[q] = \underbrace{\left[\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{q(z_n|x_n)}[\log p(x_n|z_n)]\right]}_{\text{average reconstruction}} - \underbrace{(\log N - \mathbb{E}_{q(\mathbf{z})}[\mathbb{H}[q(n|\mathbf{z})]])}_{\text{index-code mutual information}} \tag{2.57}$$

$$- \underbrace{D_{KL}[(\mathbf{z}\|p(\mathbf{z}))]}_{\text{marginal divergence}} \tag{2.58}$$

where the term $(\log N - \mathbb{E}_{q(\mathbf{z})}[\mathbb{H}[q(n|\mathbf{z})]])$ acts as a regularizer, enforcing the variational distribution $q(\mathbf{z}|\mathbf{x})$ to overlap between n. While this term regularizes the representation, it also prevents the full model to maximize the ELBO. This decomposition shows an inner trade-off of amortized variational inference : while conditioning the approximating distribution q on the data x allows an explicit variational posterior $q(\mathbf{z}|\mathbf{x})$, it also prevents the aggregated posterior $q(\mathbf{z})$ to match the prior $p(\mathbf{z})$. Some authors have then been investigating methods to match the aggregated posterior $q(\mathbf{x})$ to the prior $q(\mathbf{x})$ (see fig. 2.33). Aggregated posterior regularization can also be justified from an optimal transport perspective, as overlapping projections $q(\mathbf{z}|\mathbf{x})$ inevitably results in producing blurry samples (see section 2.1.3.3).



$D_{KL}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$



$D_{KL}[q(\mathbf{z})||p(\mathbf{z})]$

Figure 2.33: Matching the aggregated prior $q(\mathbf{z})$ instead of $q(\mathbf{z}|\mathbf{x})$ allows to prevent over-pruning, as its matching the overall distribution rather than the posterior.

As recovering the aggregated posterior $p(\mathbf{x})$ is generally intractable, we have to use implicit distribution matching $\mathscr{D}[p, q]$ to replace the D_{KL} term in the ELBO 2.18. Moreover, implicit distribution matching approaches also allow arbitrary priors $p(\mathbf{z})$, providing that we can sample from it. As we saw section 2.1.3.2, MMD provides an efficient way to estimate the divergence of implicit distributions [75], as proposed by the Wasserstein Auto-Encoder (WAE) [84], or leveraging Sinkhorn distances, computing the transport cost between the two distributions [188]. Adversarial Auto-Encoders, proposed by Makhzani & al., rather propose to use density ratio estimation (see sec. 2.19) to address the classification problem

$$r(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})} = \frac{p(\mathbf{x}|y=1)}{q(\mathbf{x}|y=0)} = \frac{\mathscr{D}(\mathbf{x})}{1-\mathscr{D}(\mathbf{x})}$$

where $\mathscr{D}(\mathbf{x})$ is the output of a discriminator. Based on a similar intuition, Mescheder & al. proposed an unified Adversarial Variational Bayes (AVB) framework, using adversarial methods to rather match the full joint distributions $p(\mathbf{x}, \mathbf{z})$ and $q(\mathbf{x}, \mathbf{z})$; however, as underlined by Bousquet & al., the actual formulation of AVB does not implicitly perform aggregated posterior. While using implicit distribution methods allows us to match the marginal distribution $q(\mathbf{z})$ to the prior, and are well-defined from an optimal transport perspective, they can yet provide degenerated representation as it encourages disjoint encoding distributions $q(\mathbf{z}|\mathbf{x})$, as shown by (2.57). Thus, some approaches propose to optimize both $D_{KL}[q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]$ and $D_{KL}[q(\mathbf{z})\|p(\mathbf{z})]$, hence maximizing the mutual information $\mathbb{I}[\mathbf{x}, \mathbf{z}]$ [189, 190].

Prior learning. Alternative approaches rather suggest to provide richer priors, assuming that the prior $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$ unavoidably yield to over-regularized representations. Tomczak & al. propose *variational mixtures of posterior priors* (VAMP), formulating the prior as a mixture of normal distributions inferred from pseudo-inputs learned with an auxiliary distribution. Interestingly, the pseudo-inputs inferred by VAMP priors implicitly represent existing patterns in the dataset, helping to perform unsupervised clustering in the latent representation. Multi-layer latent approaches, such as Deep Latent Gaussian Models [160], an also be thought as learning priors on the lower levels [32], while keeping the regularization term on the top-level. Similarly, Klushyn & al. propose to use a hierarchical prior model learning to fit the aggregated posterior $q(\mathbf{z})$. Some methods also propose to automatically adjust the number of latent dimensions, keeping the optimal dimensionality that is needed to maximize the reconstructions. Finally, the infinite VAEs proposed by Abbasnejad & al. propose to dynamically create single dimension VAEs once a sufficient number of data instances is not represented, using random affectation variables drawn from a Dirichlet process [191].

2.3.2.4 Code efficiency and disentanglement

Disentangling and D_{KL} weighting The natural disentangling properties of VAE raised an important interest in the machine-learning community, explaining the popularity of this model. The concept of *disentangling* relates to the amount of correlation between the axis of the representation, assuming that having independent axis means that the we retrieve independent factors of variation. Disentangling can thus be related to *interpretability*, as an interpretable representation would have dimensions corresponding to independent human-understandable factors of variation [192]. However, the actual origin of the witnessed tendency is hard to identify precisely. From a geometric perspective, if we want two axis of variation to be independent, these axis should be *orthogonal* in the representation, and even correspond to its axis (see fig. 2.34).

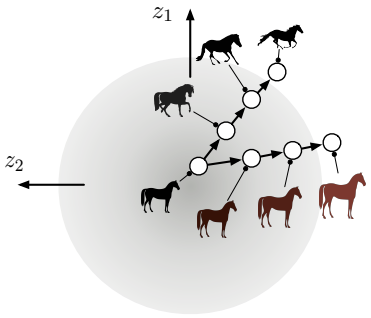


Figure 2.34: Disentangling : given two factors of variation (*running* and *color*), here neither axis are *covariant* with axis of latent space, and are not *orthogonal* between them.

The disentangling properties of existing methods such as probabilistic PCA, robust PCA [180] or non-linear ICA [193] have been investigated, all of these methods enforcing the orthogonality of representation. Rolinek & al. recently demonstrated the role of the stochastic part of the reconstruction loss, promoting local orthogonality of the decoder. More specifically, Higgins & al., underlined the role of the regularization term $D_{KL}[q(\mathbf{z}|\boldsymbol{x})\|p(\mathbf{z})]$ in disentangling, investigating the alternative objective [194]

$$\mathcal{L}_\beta \mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{x}|\mathbf{z})] - \beta D_{KL}[q(\mathbf{z}|\boldsymbol{x})\|p(\mathbf{z})] \tag{2.59}$$

and showing that increasing the β parameter was enhancing the *disentangling* properties of the representation, recovering factors of variation from image such as translation, rotation, and shape. Indeed, in VAEs, the prior is defined as a mean-field normal distribution $p(\mathbf{z})$, such that each dimension z_i of the distribution is independent. Increasing the weight of $D_{KL}[q(\mathbf{z}|\boldsymbol{x}])\|p(\mathbf{z})]$ then enforces the variational distributions

to match the prior, and then to be factorized among these dimensions. However, as shown in previous sections, increasing β also favors posterior-collapse and brings more distortion to the data, then blurring the reconstructions. Indeed, the objective 2.59 is a reformulation of objective (2.36), such that increasing β lowers the representation rate, but increases the distortion. Alternative methods then propose to rather regularize the total covariance of the aggregated posterior [150]

$$TC(z) = KL\Big[q(\mathbf{z})\| \prod_i q_i(z_i))\Big] \tag{2.60}$$

that enforces the factorization of the aggregated posterior $q(\mathbf{z})$ without increasing over-pruning. We can also increase mutual information between $\mathbf{x}$ and $\mathbf{z}$ by reducing the pair-wise distances between samples in the encoding space, as proposed in [195], or directly approximating the mutual information with optimal transport methods [196]. However, retrieving independent axis does not ensure their interpretability, as independence is rotation-invariant and may fail to extract accurate axis of variation from the data.

Disentangling and Riemannian geometry. Further insights on the disentangling properties of AEVB can be given under the scope of *Riemannian geometry*, bridging Bayesian learning and differential geometry. As summarized previously, *disentanglement* is closely linked to the idea of orthogonality, providing that the axis of the representation correspond to independent factor of variations underlying the data. In other words, an *interpolation* in the latent space should then correspond to a *flat* trajectory on the underlying data manifold. The quality of interpolations in the latent space are thus related to how much the representation has caught the inner manifold metrics, that in practice is rather unsatisfying for complex datasets.

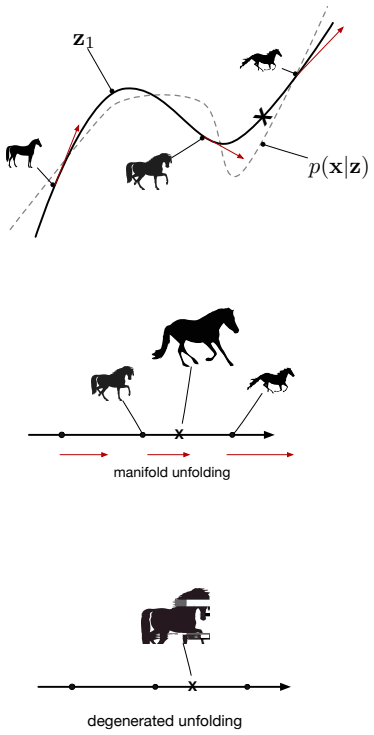


Figure 2.35: Evaluating the true underlying factors of a given dataset implies not only to extract the exact sub-manifold from the original space, but also to infer its underlying metrics.

Riemannian geometry can explain why the orthogonality of latent axis not a sufficient criterion for extracting disentangled representations. Remembering the definition of a manifold (see sec. 2.2.1), each point lying on the manifold defines a tangent space, with an infinity of possible orthogonal coordinates. According to the manifold a given metrics, the optimal or *parallel* translation between these two points is given by its geodesics, that is a property of the manifold. These inner metrics can be obtained with the intrinsic riemannian metrics, that describe the manifold curvature. Then, retrieving the ideal translation between two points in the latent space by making a linear interpolation imply to have extracted *flat* coordinates of manifold, providing an additional criterion for latenet disentanglement (see fig. 2.35). Formulating the data output $\mathbf{x} = f(\mathbf{x}) + \boldsymbol{\sigma}(\mathbf{z}) \odot \boldsymbol{\epsilon}$, the corresponding Riemannian metrics tensor can be obtained by [197]

$$\mathbf{M}_\mathbf{z} = (\mathbf{J}_z^\mu)^\top(\mathbf{J}_z^\mu) + (\mathbf{J}_z^\sigma)^\top(\mathbf{J}_z^\sigma)$$

that allows us to get an estimation of the Riemannian metrics defined by the latent representation. Geodesic interpolation between two

points can then be obtained by minimizing the curve length in the manifold, as proposed by Shao & al. [198] [199] using finite-time differences and arc length optimization. Geodesic interpolations have been shown to qualitatively improving translations between two points, indicating that the latent space coordinates are not flat respectively to the Riemannian metrics of the underlying manifold. To our knowledge, adding a flatness criterion to the ELBO has not been yet investigated in the literature, even if this idea is closely related to contrastive auto-encoders 2.45.

Riemannian approaches to variational auto-encoders then give consequent conclusions on the inner behavior of these algorithms, underlining the role of differential geometry on the emerging properties of the latent representation. Recent approaches then investigate this topic further to define non-Euclidean latent spaces such as spheres or toruses [200] , or investigating hyperbolic geometries [201, 202]. Riemannian metrics have also been investigated to regularize the latent space in a supervised way by Hadjeres & al., enforcing latent geometrical properties to match with user-defined attribute functions [203].

Multi-object approaches. Disentangling in VAEs has been mostly studied in multi-object recognition, where several objects are depicted in a single image. Multi-object recognition is a seminal machine-learning task, challenging the model abilities to understand *compositionality* of images. While very efficient methods have been developed using deep convolutional networks architectures [204] [205], these methods still do not provide interpretable representations such as the one provided by AEVB. Chazan & al. proposed *deep clustering* algorithm for object recognition by pre-training multiple variational auto-encoders on every object, and then use these models as a mixture of experts based on the reconstruction score [206]. However, pre-training specific auto-encoders for each object can be tedious, and for many objects the reconstruction procedure can be too slow. Alternatively, Greff & al. propose a multi-object recognition model using VAEs by splitting the latent spaces in subsets that are fed to multiple decoders, recomposing the original image by generating a mask in addition to the image [207]. Multi-object recognition can also be processed in a sequential way, generating the salient objects one after the other on a canvas, as done by the DRAW model proposed by Gregor & al. [208]. Similarly, the *Attend, Infer, Repeat* procedure proposed by Eslami & al is based on a single structured decoder, that is used sequentially to recompose all the objects present in a scene until the inference model stops the procedure. Latent space is then explicitly disentangled to infer separately spatial and object properties, then enforcing its interpretability [209]. Explicit latent disentanglement can then be enforced by designing the generation process, and is certainly an interesting perspective for variational auto-encoding systems.

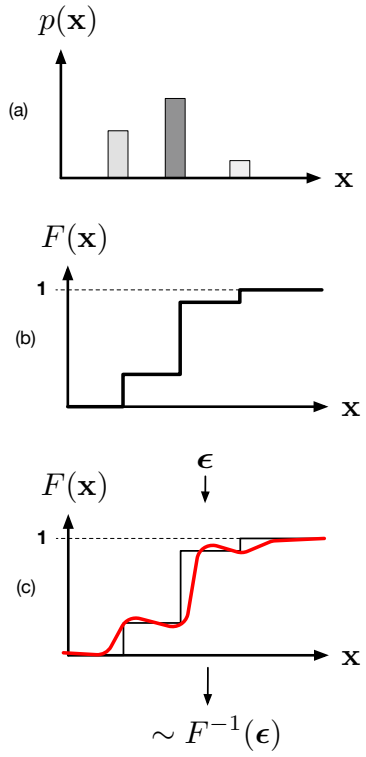


Figure 2.36: Relaxation of discrete probability function : (a) discrete probability functiion $p(\mathbf{x})$ (b) obtain cumulative distribution function $\int p(\mathbf{x})d\mathbf{x}$ (c) *relaxation* by approximating the CDF with a continuous distribution. Provided its tractable reciprocal, this distribution can be sampled.

2.3.3 Discrete latent spaces

Finally, we will address the problem of *discrete latent spaces*, that is a challenging objective as back-propagation through discrete variables is generally not possible [210]. Indeed, discrete variational distributions $q_\phi(\mathbf{z}|\mathbf{x})$ are not re-parametrizable, and approximating the gradient with a Monte-Carlo estimator is often too noisy to ensure convergence. Yet, extracting discrete representations could be useful for several applications such as unsupervised classification or topic detection [211]. Discrete representations can also be a very efficient method to reduce over-pruning when using auto-regressive decoders, as shown by van den Oord & al [212], or to achieve latent input selection by inferring a mask [213].

Continuous relaxation. Discrete latent variables are usually modeled using Categorical distributions, that can be thought as a generalization of Bernoulli distributions with $k > 2$ possible issues. Categorical distributions are typically used at the end of a classification network, jointly with a final Softmax non-linearity layer, and directly back-propagated with cross-entropy estimators. However, categorical distribution do not allow re-parametrization trick and estimation of its gradient with REINFORCE converge slowly. A solution to ease the back-propagation through discrete random variables is *continuous relaxation*, that consists in projecting the discrete representation in a continuous space using a smoothing transformation. Categorical distribution can be smoothed using *Gumbel distribution*, such that sampling from Categorical distribution can be done using samples from $g_i \sim \text{Gumbel}(0, 1)$ [214, 215]

$$\begin{aligned}\mathbf{z} &= \text{onehot}\Big(\text{argmax}_i[g_i + \pi_i]\Big) \\ &= \frac{\exp\left(\log \pi_i + g_i\right)/\tau)}{\sum_{i=1}^{k} \exp\left(\log \pi_j + g_j\right)/\tau)}\end{aligned}$$

by approximating the argmax function with a Softmax. Such variables are distributed as *Gumbel-Softmax* distribution (also called *Concrete*), whose probability density function is tractable and can be used to approximate Categorical variables. The τ parameter is called the *temperature* of the distribution, such as it tends to be atomic when $\tau \to 0$, but equals a uniform distribution as $\tau \to +\infty$. As the Gumbel-Softmax distribution is obtained from a tractable inverse cumulative distribution, it can the applied the reparametrziation trick, and then be efficiently integrated into the AEVB framework. The gradient of a Categorical distribution can then be approximated by using the one from a Gumbel-Softmax, a method called *straight-through Gumbel estimator*, that tends to the real gradient when $\tau \to 0$. While the Gumbel-Softmax trick can provide efficient approximations to Categorical distributions, they can be too smooth to accurately perform discrete sampling. Rolfe & al., followed by Vahdat & al., rather propose *spike-and-exponential* smoothing transformations, that only activates the unit if the real threshold of the corresponding latent unit is passed. Xu & al. also propose to define the simplicial probabilities by taking planar projections

of samples obtained from a Von-Mises distribution, whose support is defined on an hyper-sphere $\mathbb{S}^K$ [216].

Unsupervised and semi-supervised learning Some approaches rather extract discrete information from the continuous latent space by *quantifying* it, then labellng different zones of the latent spaces in a similar way to clustering. By example, the GM-VAE proposed by Dilokthanakul & al. [217] uses a Gaussian-Mixture prior, where the index of the corresponding mixture is inferred from the latent space. Similarly, van den Oord & al. propose to use vector quantization on the latent space to extract a discrete dictionary of latent centroids, using *straight-through gradient estimation* to back-propagate in the encoder. In that case the regularization constraint then become a ℓ^2 norm between latent projections and the closest embedding vector, then preventing over-pruning but enforcing the sparsity of the representation.

Extracting discrete information from continuous latent-space can also be performed in a supervised manner, adding a discriminator on the top of the latent space and trained on corresponding symbols $\mathbf{y}$. Classification can then be processed afterwards, such that the latent space provide a compressed representation to the discriminator. More interestingly, the auto-encoder and the discriminator can be jointly processed, such that the representation is also influenced by the gradient information coming from the discriminator. An advantage over traditional supervised approach is that, in case of incomplete label information, the auto-encoder can still be trained by inferring the corresponding label. Such methods have been proposed by Kingma & al. as *semi-supervised*, and provide a flexible method for task-oriented representation learning (see fig. 2.37). Probability densities of data $\mathbf{x}$, labels $\mathbf{y}$ and latent variables $\mathbf{z}$ can the be described

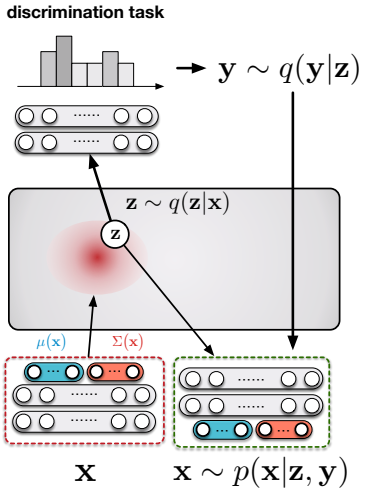


Figure 2.37: *Semi-supervised learning* : a supervised task is perform in the latent space, and the extracted label is given to the decoder for class-dependant generation.

$$p(y) = \text{Cat}(y|\boldsymbol{\pi}(\mathbf{z})); \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbb{I}); \quad p(\mathbf{x}|y, \mathbf{z}, \theta) = f(\mathbf{x}; y, \mathbf{z}, \theta)$$

where $p(\mathbf{y})$ is a prior over discrete variables. Two different bounds are then derived, depending on if the availability of the label information. If the label is available, the lower-bound is the direct application of (2.14) to the generative model :

$$\mathcal{L}(\mathbf{x}, y) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},y)}[\log p_\theta(\mathbf{x}|y, \mathbf{z}) + \log p_\theta(y) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})]$$

that is generally added an explicit classification cost $\alpha \cdot \mathbb{E}_{p_l(\mathbf{x},y)}[-\log q_\phi(y, \mathbf{x})]$. If the label information is missing, we compute the expectation over the full inference distribution $q_\phi(y, \mathbf{z}|\mathbf{x}, y)$

$$\mathcal{U}(\mathbf{x}) = \mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{x},y)}[\log p_\theta(\mathbf{x}|y, \mathbf{z}) + \log p_\theta(y) + \log p(\mathbf{z}) - \log q_\phi(y, \mathbf{z}|\mathbf{x})]$$

where the expectation respectively to $\mathbf{y}$ can be used by summing the possible values of $\mathbf{y}$, $\mathbf{y}$ being discrete. As the influence of the discriminator can be important and remove the unsupervised features of the latent space, the discrimination step can be proceeded on the second layer of a multi-layer VAE as proposed by Kingma & al., or using auxiliary variables [218]. Bouchacourt also propose a disentangling

method by grouping observations, enforcing the same support for data items with common label properties [219].

Learning spectral representations and audio regularization strategies

3

In the previous chapter, we introduced two different frameworks for the extraction of invertible representations : Bayesian latent models, based on modeling the data with a generative distribution $p_\theta(\mathbf{x}|\mathbf{z})$ that is conditioned on inferred latent variables $\mathbf{z}$, and methods coming from representation learning, leveraging high-capacity function approximators to extract features from the data according to a suitable loss function. We saw that these two frameworks could be beneficially combined to obtain an hybrid method for generative models, called Auto-Encoding Variational Bayes (AEVB), that is based on both Bayesian statistical inference and neural networks to model both the generative model $p_\theta(\mathbf{x}|\mathbf{z})$ and the approximated posterior $q_\theta(\mathbf{z}|\mathbf{x})$.

In this chapter, we will invest AEVB as an generative model for audio signals. We focus on AEVB because it combines two different processes : the **generation process** $p_\theta(\mathbf{x}|\mathbf{z})$, that generates new samples from the analysed dataset conditioned on a latent representation $\mathbf{z}$, and an **inference process** $q_\theta(\mathbf{z}|\mathbf{x})$, that extracts this latent representation in an unsupervised way. This PhD thesis focuses on this idea of using this framework to mirror audio *analysis-synthesis* methods, that are seminal DSP techniques based on two similar steps : a *synthesis* process, generating a signal with respect to a set of parameters, and an *analysis* process, that extracts the generative parameters from an incoming audio signal (see 3.1). As an introduction, we will shortly summarize some analysis and synthesis methods in the DSP domain, linking them to machine-learning methods, and then reviewing the actual analysis-synthesis techniques and see how our work takes place in the current state of the art.

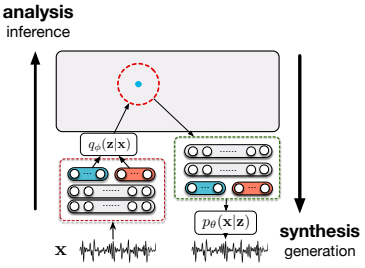


Figure 3.1: The inference/generation processes of AEVB can be thought as mirroring a DSP analysis/synthesis system, extracting invertible features from a given corpus of audio signals.

Analysis and inference *Analysis* methods, in the domain of audio DSP, are commonly based on the extraction of higher-order features from the signal, aiming to reflect structural, acoustical or musical properties of the signal. These high-order features, that can also be called *descriptors* in the audio DSP literature, can be based on the extraction of features from intermediate transforms, that are generally derived from *spectral* transformations (see 3.10), to reflect perceptual properties of the signal : statistical or perceptual descriptors such that fundamental frequency, spectral centroid, and many others. Contrary to the features extracted with standard machine-learning, that are not domain-dependant, audio descriptors are generally crafted by researchers to reflect these qualities, linked to perceptual attributes that can be confirmed with psycho acoustical experiments (see sec. 3.4.1). While these descriptors are generally short-term, some analysis methods rather target to model the *dynamics* of the signal, whether by tracking the evolution of these descriptors through time [220] or to model its temporal structure (for speech analysis [221], energy profiles

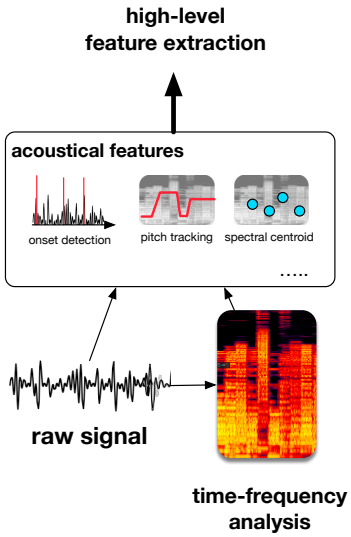


Figure 3.2: Most MIR approaches extract acoustical features from the signal, and/or from signal representations to extract higher-level features, as could be used here from chord detection.

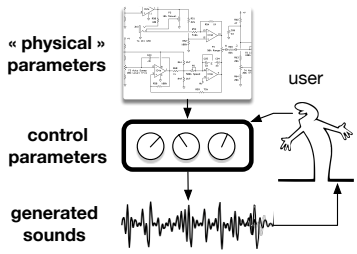


Figure 3.3: Classical audio synthesis methods are based on a physical (analog) or algorithmic (digital) structure, some parameters being offered to the user to shape the sound.

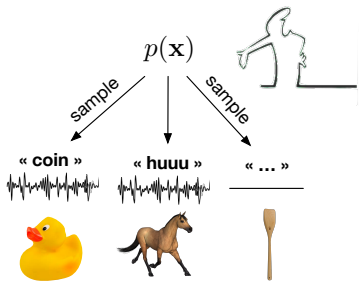


Figure 3.4: Even if a distribution $p(\mathbf{x})$ perfectly represents the modelled data, the lack of parameters makes it unusable.

[222] or even audio watermarking [223]). The domain of *music information retrieval* (MIR) [224–226] focuses on the extraction of *musical* features from an audio signal, that are typically high-order descriptors mixing temporal and spectral dimensions such as pitch and chord tracking [227, 228], beat extraction and tempo estimation [229–232], structural segmentation using similarity matrices [233] or spectral clustering [234–236] and more high-level tasks like emotion recognition [237, 238], genre recognition [239, 240] and music recommendation [241–243].

As MIR is focused on high-level tasks that often involve several descriptors on multiple temporal scopes, the resort to machine-learning techniques for feature extraction can be relevantly introduced to support audio analysis frameworks. The recent raise of machine learning techniques based on convolutional neural networks motivated their direct application on spectrograms, by example on classification/regression tasks [244–247]. However, the lack of interpretability and the data-centred nature of these algorithms aroused a mixed reception of the MIR community, despite a substantial increase in the results. Therefore, hybrid methods between MIR and machine-learning have since been investigated, combining best of both worlds to address complex tasks such as source separation [248–251], transcription [252] or voice recognition [253].

Synthesis and generation Audio synthesis is a one of the most seminal domain of audio DSP since its advent [254, 255], partly due to the early use of analogical and then numerical devices in music creation. Digital synthesis appeared almost simultaneously with computers, and is still a flourishing domain of research led by many different approaches : additive and substractive synthesis [256, 257], FM synthesis [3], *sampling*, *granular* synthesis [258], *physical modeling* [**Avanzini01controllingmaterial**, 259–261] and many other, that are now widely integrated in modern practices of music production and performance. Digital synthesis can be loosely defined as a generating system whose state is defined by a set of parameters, that conditions the structure of the generated signal (see fig 3.3). Parameters of digital synthesis techniques are often tied to structural properties of the generation model, such that the effect of each generative parameter is explicitly known. Some parameters are then offered to the performer or producer for direct interaction, providing *controls* for the generation model. Therefore, the identity of synthesis method is not only defined by the range of sounds it can produce, but also by the controls it gives to the human user.

Machine-learning based generative models are slightly more complicated, as their development come from different origins and initially bypassed this interaction aspect. As we summarized last chapters, generation in the machine-learning domain was rather formulated as the modeling an underlying stochastic process $p(\mathbf{x})$ under a finite set

of observations $\{\mathbf{x}_1\mathbf{x}_2...\mathbf{x}_n\}_{i=1}^{N}$. However, this distribution can be unsatisfactory for musical purposes as it only allows *sampling*, and do not allow explicit control (see fig. 3.4). Therefore, we saw in the last section a way to define an extended model $p(\mathbf{x}, \mathbf{z})$ with parameters or additional inputs $\mathbf{z}$, such that $\mathbf{z}$ can shape the density of $\mathbf{x}$ and then control the generation. However, the number of machine-learning generative method specifically developed for generation is not as flourishing than in digital audio synthesis, and the generative models can be gathered in four families : *auto-regressive* [262, 263], *auto-encoder* approaches (see sec. 2.2.3), *generative adversarial networks* [23, 264, 265]), and recently flow-based models [171, 266, 267]. However, only auto-encoding and flow-based approaches are invertible, and can be thus used for our purpose.

Therefore, a seminal difference between digital audio synthesis and machine-learning generative models is that, in the first case, *the model shapes the generated sound*, while in the second *data is shaping the sound*. We think that these solutions are equally good, and that this complementarity can significantly help human creativity. However, we think machine-learning based approaches have still to catch up digital audio synthesis on various aspects, especially in terms of perceptual qualities, usability and interaction. While we investigate the second chapter 5, the two following sections will propose diverse ameliorations of AEVB generative models for sound synthesis, tackling two different aspects : latent space regularizations, and temporal modeling.

Analysis-synthesis The complementarity between analysis and synthesis also exists in the signal processing domain, such that powerful analysis methods do not generally allow direct regeneration of the signal, and that conversely digital synthesis algorithms do not provide any way to infer corresponding parameters from a given sound. This duality then motivated the development of frameworks allowing both analysis and synthesis, providing bijective methods between audio and parameters domains. These models, called *analysis-synthesis* methods, often rely on *invertible* transformations that both extract features from the signal and allow the regeneration of original data. The obtained features can then be used whether for analysis purposes (such as spectral transforms, that ground most of perceptual descriptors) or sound transformations, modifying the incoming sound in the obtained representation. *Phase-Vocoder* algorithm is a seminal example of such method, using the invertibiliity of Short-Term Fourier Transform (STFT, see sec. 3.10) to transform existing sounds (such as transposition, time stretching, or partial tracking), or for direct generation by designing sounds directly in the spectral domain. Other analysis-synthesis methods, such as ESPRIT [268] or cepstrum-based representations [269], also provides bijective representations that are based on different modeling properties. However, a drawback of these methods is that their general invertibiliity enforces the bijectivity of the transform, such that the dimensionalities of data and representation domains are equivalent. Hence, these methods are often used as intermediary signal

representations for higher-level analysis / generation methods.

Hence, we propose to use the AEVB framework as an analysis-synthesis method for digital audio synthesis, combining modern probabilistic generative and latent models to extract high-level invertible audio features. This model gains from other generative models such as GANs by also learning an inference process on the latent representation, thus allowing the system to structure the space without supervision and to retrieve the parameters corresponding to an incoming data. We also think that this framework provides a complementary approach to classical DSP analysis-synthesis methods, targeting to extract meaningful features from a restricted set of data rather that requiring the global invertibility of the representation. The relaxation of this requirement thus allows us to reduce the input dimensionality, rather focusing to model a dataset with a finite amount of samples.
This section is divided in three parts. First, we will first provide some baselines of performances obtained with diverse AEVB systems, that we train with the magnitude of audio spectra. These baselines, developed during the early phase of the PhD, were at our knowledge the first application of AEVB to audio signals. Then, we will present two latent regularization strategies that we developed specifically for audio processing and generation : a first method based on external symbolic information ()such as pitch, octave and dynamics) using the latent space as a *translation space* between signal and symbol domains, and a second method based on perceptual constraints, obtained from psycho-acoustical measurements.

3.1 Spectral representations of audio signals

In this section, we evaluate the performances obtained with diverse AEVB systems when trained on audio signals. However, learning directly audio waveforms is not straightforward, containing short and long-term temporal dependencies that would involve higher-order time series models (see section 4). Hence, we rather perform training on *magnitude spectra*, that allows us to train on stationary and parsimonious data that more easily reflects structural perceptual attributes of audio signals. As such transforms are widely used in the DSP community and implemented in most audio software, it also allows to easily develop a real-time audio engine to generate from the obtained models (see section 5), hence offering a good compromise between usability and complexity.

3.1.1 Invertible spectral representations of audio signals

Raw representations. The *raw* representation of an audio signal is the direct description of the corresponding physical pressure wave, that can be represented using one or several signals $\mathbf{s}(t)$, or by an acoustical field $s(t, \mathbf{x})$, $\mathbf{x}$ being the position in space. This signal can be

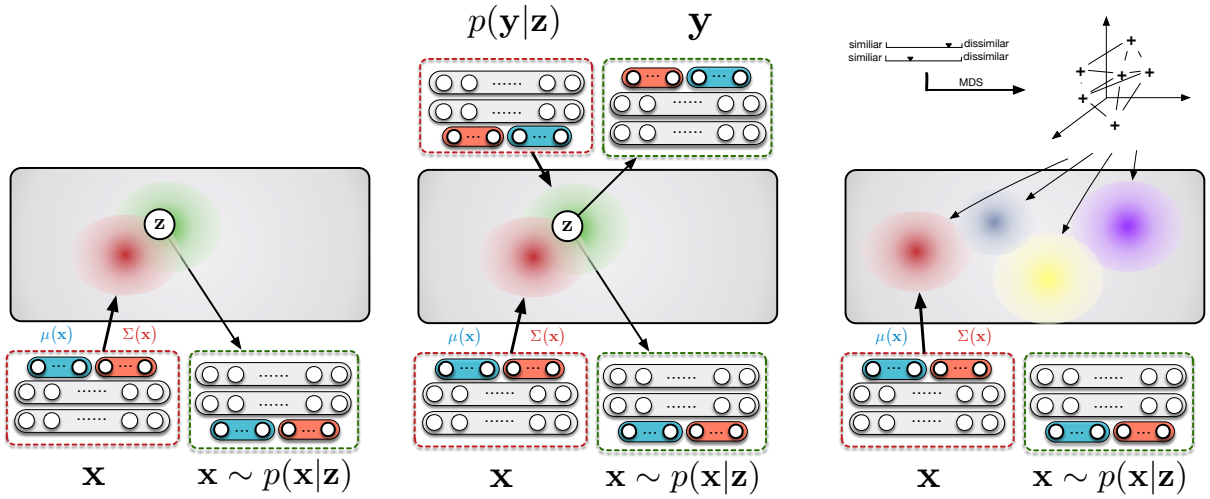


Figure 3.5: Diagrams showing the three models used in this section. (left) simple AEVB system, trained on audio reconstruction (middle) AEVB with translating latent space, using audio **x** and symbol data **y** 3.3 (right) perceptual regularization of the latent space, using dissimilarity experiments of instruments sounds from audio perception studies 3.4

recorded at one or several positions in space using a transducer, or directly generated with a synthesis engine. In the numerical domain, this wave is sampled both in time and amplitude in order to be encoded numerically. The sampling frequency, corresponding to the number of samples recorded by seconds, thresholds the maximum frequency that we are able to render by the Shannon-Nyquist equation $f_s = 2f_{max}$ (see fig 3.6). The standard value f_s for CD audio recordings, that is 44100 Hz, thus means that we can restore the original frequency content of the signal up to 22050 Hz, that is the standard upper perceptive threshold of human audition. The number of bits used to encode the numerical value of each sample, called the *bit rate*, defines the range of *dynamics* that the recording is able to render. This bit rate is usually expressed in decibels (dB), such that a 16 bits resolution allows a 90dB range.

Individual raw waveforms are then 1-dimensional discrete time series arbitrarily, that are defined onto a range $[-1; 1]$. These signals can be modelled through time with several analysis methods such as auto-regressive, integrated, or moving-average models (see sec 4.1). However, the high temporal density and non-stationary nature of these signals make most of these methods cumbersome on the long-term, such that their direct use for audio feature extraction is difficult. Moreover, the temporal scope needed to observe some high-level audio features (from beat detection to genre recognition) can be very large, and then make their extraction from time series dauntingly hard.
Modern machine learning techniques, based on the modeling of complex auto-regressive distributions $p(x_t|x_t, \mathbf{h})$ such as Hidden Markov Models, are also trained with difficulties, then requiring more subtle approaches (see chapter 4). As most time series analysis are defined

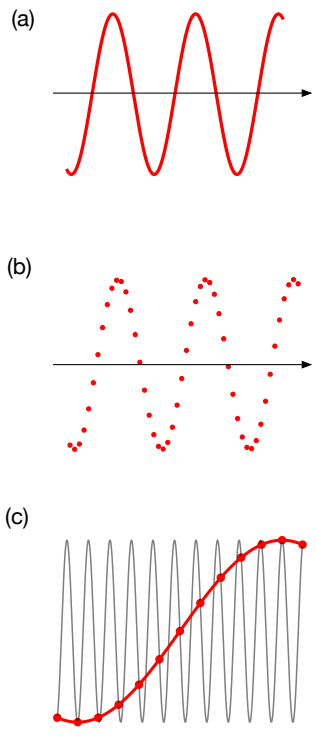


Figure 3.6: (a) analog signal x(t) (b) *sampled* series x[n] (c) *aliasing* : the discretization of the waveform introduces an infinite amount of possible frequencies $f_{alias} = f_s - nf - \pi$

in the real domain, some machine-learning algorithms rather use the underlying binarization of numerical signals to define a classification problem, sparsifying the data and speeding up the convergence [262]. μ-law is a common non-linear encoding scheme that encodes more precisely high amplitude values, thus reducing the signal/noise ratio

$$F(x) = \text{sgn}(\mathbf{x})\frac{\log 1 + \mu|\mathbf{x}|}{\log(1+\mu)}$$

However, this methods also significantly increase the dimensionality of the input space, transforming each sample into a $\mathbb{R}^d$ vector whose dimensionality scales exponentially with the number of bits (16 bits amounts to 65 536). Therefore, some downsampling is required, altering the quality of the audio signal. Raw waveform learning is then one of the most difficult machine-learning challenge, is cannot be tackled on sufficiently large time scales with standard AEVB frameworks. Hence, we rather resort to alternative representations with smoother properties, defined in the *spectral domain*, that significantly ease the training of the proposed models on audio information.

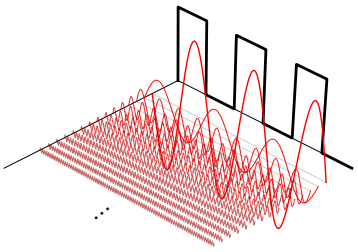


Figure 3.7: Decomposition of a square signal over the 15 first spectral components.

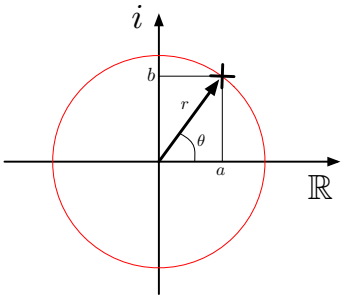


Figure 3.8: A complex number can be whether expressed in Cartesian coordinates $()a, b)$ or polar coordinates (r, θ).

Fourier transform and time-frequency trade-off While raw audio waveforms is the closest representation of the corresponding acoustical phenomenon, it is difficult to model with the proposed techniques. Furthermore, this representation does not provide explicit information about the *frequency* content of the signal, that provides a more direct interpretation of several properties of human perception such as pitch, harmonicity, or timbre-related attributes. Given a continuous signal $x(t),]$ its frequency content can be extracted using the Fourier transform, a representation defined in the complex domain $X(f) \in \mathbb{C}$

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-2\pi xf}\,dx \tag{3.1}$$

that can be thought as the projection of the signal onto an infinite function space, whose basis are the complex exponentials $\exp ix = \cos x + i \sin x$ (see fig. 3.7). The Fourier transform then describes the signal as a function of frequencies f instead of time t, and can be intuited as the convolution of the signal with a complex exponential of frequency f. As $X(f)$ is defined in the complex plane $X(f) \in \mathbb{C}$, it is composed by a real part a and an imaginary part b, such that $X(f) = a + bi$. The complex frequencies $X(f)$ can then also be expressed using polar coordinates $r = \sqrt{x^2 + y^2}$ and $\theta = \text{atan}\frac{y}{x}$ (see fig. 3.8)

$$X(f) = re^{i\theta}$$

where r is the radius and θ is the angle of $X(f)$. Expressing spectral components in polar coordinates provides a more explicit interpretation of $X(f)$: the radius is the *magnitude* of the spectral component $X(f)$, while the angle θ corresponds to its *phase*, the relative position of the component relatively to its period (see fig. 3.9). This decomposition is widely used in audio signal processing, as it allows us to detach the magnitude of each spectral component from its temporal information. Furthermore, this transformation is entirely invertible, such that we

can recover the original temporal $x(t)$ from $X(f)$ by performing inverse Fourier transform

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{2\pi x f} df \tag{3.2}$$

who is defined on the real domain $\mathbb{R}$, as the Fourier transform of a real signal is an even function.

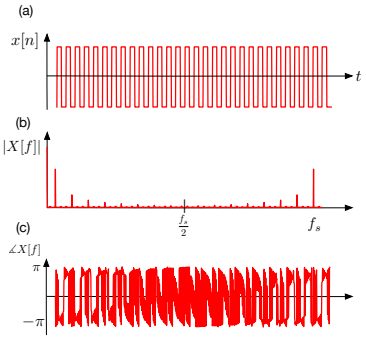


Figure 3.9: (top) signal x[n] (middle) magnitude spectrum $|x[f]|$, with the aliasing $k > f_s/2$ and (down) the phase $\angle X[f]$.

The expression of Fourier transform defined in 3.1 allows us to get an invertible and decomposable representation of a signal $x(t)$ in the frequency space $f \in \mathbb{C}$, and is then a seminal transformation in signal processing. However, in the numerical domain, the signal is a discrete time series $x[n]$ sampled at rate f_s, such that transforms 3.1 and 3.2 are not applicable. The *discrete Fourier transform* (DFT), that is an extension of continuous Fourier decomposition to discrete signals $x[n] = \{x_0, x_1 ... x_{N-1}\}$, can be expressed as

$$X[k] = \sum_{i=0}^{N-1} x[n] \dot{e}^{-\frac{i2\pi}{N} kn} x[n] \qquad = \sum_{k=0}^{N-1} X[k] \dot{e}^{-\frac{i2\pi}{N} kn} \tag{3.3}$$

that defines a finite complex vector $X \in \mathbf{C}^N$, that is periodic with period N because of the aliasing caused by sampling. The discrete frequencies k can be converted back in Hertz using the identity $f_k = f_s/k$, as the period N of the transform is the sampling frequency f_s of the signal.

An seminal property of spectral representations is their inherent duality between time and frequency. Indeed, the exact estimation of the frequency content requires the observation of the signal over an infinite time range, while conversely the exact invertibility of the spectral representation needs to integrate the spectrum over an infinite frequency domain. Therefore, the higher we want the estimated spectrum to be precise, the larger we will have to integrate the signal, hence discarding its dynamics. This time-frequency trade-off, also called *Gabor limit*, is seminal in spectral representation, implying that the spectral resolution of a signal is complementary to the time range of the observation. The hypothesis underlying this duality is the *stationarity* of the signal, assuming that its spectral content stays the same over an infinite time. In addition to being difficult in practical applications, the key point of audio signal analysis is also to retrieve the dynamics of these spectral features, such that this trade-off is problematic for signal analysis.

A trade-off between time and frequency resolution can be found by slicing the signal in short temporal windows, that are then used for spectral analysis. This can obtained by the *Short-Term Fourier Transform* (STFT), that is then dependent of both time and frequency

$$\text{STFT}[x(t)](\tau, f) = \int_{-\infty}^{+\infty} x(t) w(t-\tau) e^{-j\omega t} dt$$

where $w(t)$ is a window function with finite time support, such that the corresponding spectral content is extracted around $t = \tau$. If the sum of

windowing functions sums to one over time, such that $\forall t \sum_\tau w(t-\tau) = 1$, the original signal can be exactly recovered by inverting each frequency slice and performing the same windowing function (then called *overlap-add* reconstruction). STFT allows us to disentangle frequencies and time, and then to describe the evolution of spectral content over time (see fig 3.10). STFT then provides the baseline of most digital audio analysis techniques developed so far, and allowed the development of analysis-synthesis techniques such as the phase vocoder, providing an efficient method to both represent and generating signals [270].

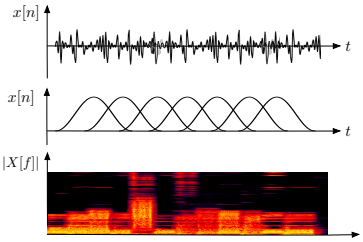


Figure 3.10: Windowing the signal and performing FFT on each segment allow to represent the frequency evolving through time.

Alternative time-frequency representations. Several others time-frequency representations based on the same principle have been developed. *Discrete Cosine Transform*, that is an alternative formulation of STFT defined on the real domain [271]. DCT allows to represent the phase by introducing the idea of *negative magnitude*, then entangling phase and magnitude. However, we found that machine-learning were struggling to learn this representation, as this disentanglement of phase and magnitude was preventing the algorithm to catch salient features.

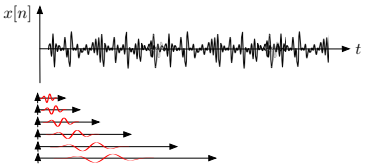


Figure 3.11: Examples of wavelets. The input wave is decomposed in *wavelets*, allowing to perform multi-resolution analysis.

A well-known alternative to Fourier transforms are *wavelet transforms*, that project the signal on arbitrary families of multi-resolution frequency bands. Wavelets allow to avoid the time-frequency trade-off of STFT, as the different frequency bands are defined on multiple time scales (see fig. 3.11). Contrary to DFT, that assumes a linear frequency scale, wavelet transforms allow to model an arbitrary range of frequencies, allowing to be arbitrarily precise over a specific spectral span. *Constant-Q transform* is an example of wavelet-based transform widely used in musical signal processing, defining a logarithmic frequency range centred on octave fractions, providing an precise representation for the analysis of pitched audio signals [272]. Unfortunately, this multi-resolution property makes wavelet transforms non-invertible, and then not suitable for generation. However, the recent *Non-Stationary Gabor Transform* (NSGT) alleviate this problem by proposing an invertible wavelet representation, using multi-resolution alignment of windows for accurate offline inversion [273]. This transform is used for the proposed perceputal regularization method of latent spaces (see sec. 3.28).

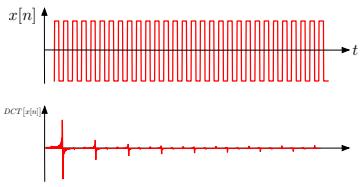


Figure 3.12: *Discrete Cosine Transform* (DCT) encodes the phase in the negative components of the spectrum. It can be understood as a DFT in a different base.

Learning complex representations. Time-frequency representations are then an efficient solution for training machine-learning methods on audio signals, providing a parsimonious representation that extract relevant features in terms of structure and human perception. However, most time-frequency are defined on a complex domain, such that the usual gradient-based methods based on derivability are not applicable. Neural networks developed specifically for complex-domain learning have been investigated [274–276], that could be combined with complex normal distributions over spectra [250, 277] ; however,

we leave this investigation for future works.

We take a more straightforward solution by rather resorting to the *polar decomposition* described above, decomposing a spectral component $X(f)$ into a positive real *magnitude* vector $|X(f)| \in \mathbb{R}^+$ and a *phase* vector, bounded between $\theta(f) \in [0; 2\pi($. In the audio domain, the information provided by the spectrum magnitude contains most of perceputal properties, corresponding to the amplitude of all the frequency component. Furthermore, magnitude spectrum of simple signals are generally sparse, then providing a suitable input space for machine learning algorithms.

Conversely, the phase information is difficult to learn because of its high variance and its invariance through time. Indeed, phase represents the temporal part of the spectral transform, and is seminal to preserve the temporal consistency of a given sequence of spectral frames. However, as in this chapter we will perform training on individual frequency frames, direct interaction from the model's latent space will not preserve the temporal coherency of initial samples, so it seems reasonable to drop the information contained in the phase.
However, as phase is important when inverting a sequence of several spectral frames, we have to leverage *phase reconstruction* algorithms to reconstruct the correct shape of the obtained generation. Phase reconstruction is a difficult task, as it implies recovering half of the information lost from the original spectrum. A standard method for phase reconstruction is the Griffin-Lim algorithm, based on iterative STFT and ISTFT passes to retrieve the phase distribution ensuring the consistency of the transformation [278, 279]. Several methods have then been proposed, such as phase gradient-heap integration proposed by Pruša and Søndergaard [280]. In this work, we use Griffin-Lim reconstruction with 30 iterations for offline examples.

3.2 Audio analysis-synthesis with variational auto-encoders

In this section, we will then provide a first baseline of AEVB performances when trained on spectrum magnitudes. As we propose to use AEVB as an analysis-synthesis method, our objective is twofold. First, we have to evaluate the generation capacities of the model, i.e. evaluating the reconstruction ability of the model over data samples. As usual in the machine-learning domain, this evaluation is performed on another set, called the *test set*, that is different from the train set, to measure the generalization ability of the model. We also have to study the emerging properties of the extracted representation $\mathbf{z}$, and if the obtained space is explaining the underlying structure of the data. As we summarized last section, the flexibility of AEVB techniques provide numerous possibilities in terms of architecture, regularizations, and training criteria ; we then propose to evaluate the influence of these choices on the emerging properties of the obtained latent spaces.

Furthermore, as machine-learning methods are highly dependent on the involved data, we also evaluate these methods on three datasets, with separated generative factors and different complexity.

3.2.1 Datasets.

The data-centred approach proposed by machine learning then requires a particular attention on the choice of evaluation datasets, as their design will condition the properties we will observe. Moreover, the comparison of different models involves to be applied on the same dataset, thus requiring *reference* datasets that are designed to reflect the main challenges of the targeted task. A typical example of such dataset is the MNIST corpus used in image processing, containing over 80,000 examples of hand-written digits [159]. This dataset still provides a very efficient baseline for evaluation purposes, as it encompasses most of the image processing challenges : classification of digits, style recognition, and limited variability (limited number of lines, 10 classes, and black & white). The comparison of different machine-learning models is then straight-forward, providing a direct feedback on how the concerned models are behaving on these different factors of variations.

While some reference datasets for machine learning are established in the image processing domain for decades, such datasets are unfortunately missing in the audio processing domain. Valuable reference datasets exist for main MIR domains such as pitch extraction, structural segmentation or style recognition, but datasets focused on evaluating the generative models performance are, to our knowledge, missing. Hence, we built different datasets for the purpose of this thesis, focusing on different factors of variation. The proposed datasets are split in two types : first, we designed *toy datasets* using sound synthesis, allowing us to focus on seminal generative factors such as pitch, harmonics, and spectral content. Then, we evaluate the proposed models on audio data coming from real audio recordings, then having more variance and allowing us to evaluate the credibility of generated samples.

3.2.1.1 Toy datasets.

As we aim to study the performances of AEVB on spectral magnitudes, we have to design our toy datasets on generative factors that are transparent under this representation. The proposed toy datasets, that can be easily generated using simple synthesizers, are very convenient to rapidly evaluate the VAE performances over a short range of audio factors with an arbitrary number of data samples. These toy datasets are available here here, and can be synthesized with the `vschaos` toolbox.

Additive dataset. The `toy_additive` dataset has been obtained by synthesizing harmonic sounds over a range of 100 different fundamental frequencies, then focusing on the *pitch* and *harmonic* content

of the sound. Additive synthesis is based on the following generative process

$$x[n] = \sum_{l=1}^{L} \exp\left(\frac{-ld}{L}\right) \cos\left(2\pi f_0 l \frac{n}{f_s} + \phi\right) s \tag{3.4}$$

where f_s is the sampling frequency of the target signal. This generator has 3 parameters : the fundamental frequency f_0, the number of partials L, and the harmonic decay d. The fundamental frequencies are chosen between 64 and 1024 Hz, the number of partials between 1 and 7, and the harmonic decay between 0.2 and 4. The phase is sampled individually from a normal distribution, and the sound is then enveloped with simple linear ramps of 100 milliseconds to avoid discontinuities. The total number of examples then amounts to 3500, and provides a simple dataset to grasp the VAE performances on fundamental properties of sounds.

FM dataset. We also provide a more complex toy dataset, `toy_fm`, based on *FM synthesis*, that is a widely used technique in signal processing and digital sound synthesis. This method allows us generate sounds covering a wider space of frequencies and inharmonic relations with a limited number of parameters, based on the generative process

$$x_{carrier}[n] = \cos\left(2\pi m * f_0 \frac{n}{f_s} + \phi_c\right) \tag{3.5}$$

$$x[n] = \cos\left(\alpha x_{carrier}[n] * f_0 \frac{n}{f_s} + \phi\right) \tag{3.6}$$

where $f0$ is the carrier frequency of the signal, m the ratio of the modulating frequency, and α the modulation amplitude. Indeed, if $m \in \mathbb{N}$, the obtained will be harmonic, producing an amount of frequency components that is controlled by the modulation amplitude. However, for non integer values of m, the generator will produce complex spectra involving non harmonic partials, whose strength can be controlled by α. We then generated a total amount of 21 250 examples, with 25 center frequencies from 60 to $600Hz$, 25 FM ratios from 0 to 3, and 34 modulation amplitudes from 0 to 30.

3.2.1.2 Ground-truth datasets.

To evaluate the capabilities of the models on real-life examples, in this PhD we selected two datasets: a first one, `acidsInstruments-ordinario` consisting of orchestral instruments samples, and a second one, `diva_dataset`, consisting of sounds coming from presets of a commercial synthesizer, *Diva*[*].

Orchestral ordinario sounds.

[*] https://u-he.com/products/diva/

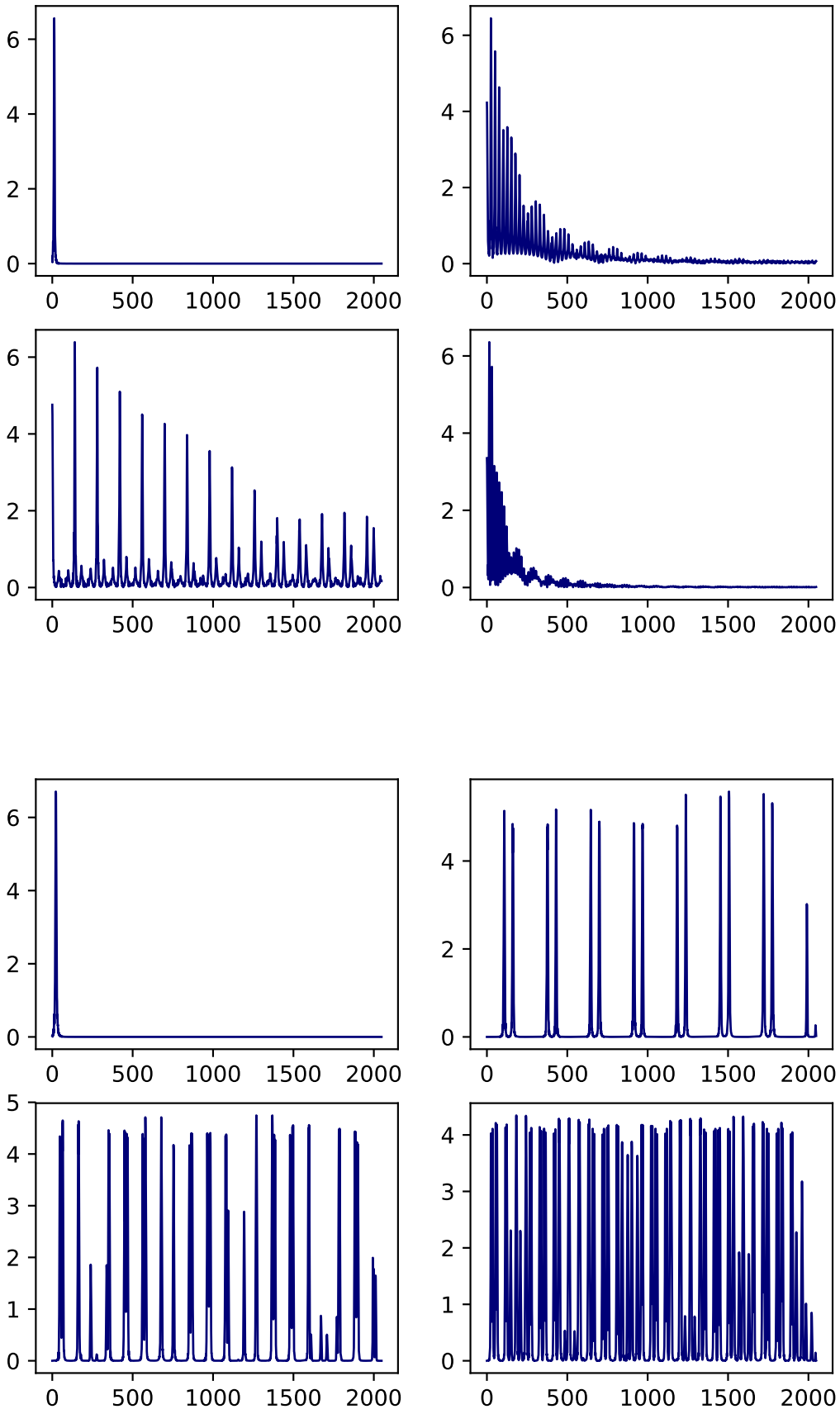


Figure 3.13: Examples of spectra generated by the toy dataset. (top) : examples from `toy_additive`, generated from a fundamental frequency and a given number of partials, whose decay are paramatrized with the exponential decay parameter. (down) : examples from `toy_fm`, that are typically much more complex than the ones from `toy_additive`, where the partials are not spread according to harmonic relationships.

The first dataset, `acidsInstruments-ordinario`, is built from orchestral instruments samples coming from the *Studio On Line* database [281]. This sample bank offers a wide variety of instruments playing in various modes, and is thus a precious source for building datasets. However, to restrict this bank to a tractable database for our VAE, we only retrieved the samples taken from *ordinario* playing modes of the following instruments : English-Horn, French-Horn, Tenor-Trombone, Trumpet-C, Piano, Violin, Violoncello, Alto-Sax, Bassoon, Clarinet-Bb, Flute, and Oboe. The dataset contains samples from the full span of the tessitura for each instrument, and is available in four different dynamics : *pp*, *p*, *mf*, *ff*, amounting to 1885 files. The size of this dataset is reasonable, quite small compared to most image processing datasets (80000 for MNIST), seems realistic for practical applications, reflecting how the system would be have if the user wanted to train it on a custom set of sounds (such as sample banks).

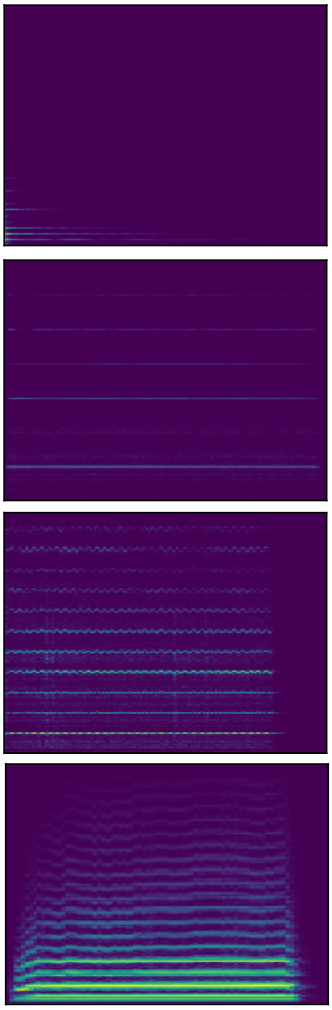


Figure 3.14: Examples of spectra from the `acidsInstruments-ordinario` dataset.

Diva dataset The second dataset, `diva_dataset`, is built from the audio signals generated with the synthesizer *Diva* from U-He, that is a cutting-edge polyphonic virtual synthesizer widely used in music production. *Diva* is based on the simulation of a analogue synthesizers, offering many features and a generous set of parameters allowing to generate a wide diversity of different sounds. This dataset was chosen because most of its parameters are MIDI CC controllable, allowing us to automatically generate samples according to a sampling scheme, and because of the amounts of presets available online.
We manually established the correspondence between the synth and MIDI parameters, as well as their values range and distributions. We only kept the continuous parameters, normalizing all their values to [0, 1], the other being set to their fixed default value. We used RenderMan* to batch-generate all the audio files by playing the note for 3 sec. and recording for 4 sec. to capture the note release. The files are saved in 22050Hz and 16bit floating point format. This procedure allowed us to gather a total amount of 11,000 presets, providing a precious ground-truth database for synthesized sounds. We use this dataset in the chapter 4, as we more interested into the temporal evolution of these sounds than into their spectral content.

3.2.2 Evaluation of generative and representational properties.

In this section, we provide preliminary results on the performances achieved by VAE when trained on magnitude spectra. The evaluation of generative models is a difficult task, especially in music where the creative and perceptive dimensions of the generated sounds are subject dependant. While we let the the *creative* evaluation to a further work (yet initiated, see chapter 5), we follow the *function-behavior* ontology of Gero & Kannegiesser [282, 283] as a first evaluation of our model, implying first to define an expected behaviour and then how to measure it.

* https://github.com/fedden/RenderMan

As our method targets both in generation and inference, we propose a twofold evaluation procedure. The first expected behaviour of the system is being able to reconstruct the samples from the original dataset. Hence, we evaluate the reconstruction performances performed by the models with different spectral losses, on a test set that has not been used for training.
The second aims to evaluate the emerging properties of the obtained latent representation. This is a rather hard task, this multi-dimensionality of this space preventing to obtain accurate visualizations. In addition to measure how much the obtained representation differs from the prior $p(\mathbf{z})$, that is a training criterion, we also perform independence measures, disentanglement measures, and descriptor-based topology maps, to get additional insights on the organization of the latent space.
Regarding baselines, the originality of inference and generation process of AEVB systems make them difficult to compare with other models. Therefore, we use invertible dimension reduction methods (PCA and ICA) to mirror the encoding / decoding processes, the latent space corresponding to the obtained projection.

Models. The flexibility of AEVB framework allows the full parametrization of both inference and generation processes, granting the right to choose almost every architecture parameters : encoding/decoding distributions, relative loss weights, and regularization loss. Moreover, we reviewed during last chapter the numerous evolutions proposed in the literature to tackle different aspects of the system, providing a plenteous amount of design choices.
As we can not obviously test all the models presented in the last chapter, we first chose to study the impact of latent space dimensionality (hence the compression performance of the model), the D_{KL} weighting β (hence, the impact of the representationr rate), and the regularization strategy. To isolate the respective effects of these design choices we start from a standard VAE, and vary these factors one by one. We obtain the following experiments

- **latent dimensions** : four different VAEs of respectively 2, 4, 8 or 16 latent dimensions
- **regularization weighting** : three different β-VAEs, comparing $\beta \in [1, 4, 10]$
- **regularization divergence** : three different explicit divergences (D_{KL}, Renyi with $\alpha = 2$, Jensen-Shannon divergence) an two implicit divergences (MMD, Adversarial).

all of these experiments are applied to the three datasets `toy_additive`, `toy_fm`, and `acidsInstruments-ordinario`. The encoders and decoders are symmetric 4-layers convolutional auto-encoders, with respectively 3, 7, 9 and 11 kernels, 64, 32, 16 and 8 channels, ReLU non-linearities [284] and batch-norm normalization [285]. The models were all trained using the ADAM optimizer [123], with an initial learning rate of $1e-4$ and a batch size of 64. For the adversarial loss, the classifier is defined as a fully-connected two-layer discriminator with 500 linear units,

trained separately with the same optimization parameters.

3.2.2.1 Reconstruction evaluation.

We evaluate the performances our model in reconstruction on both *train* and *test* partitions of our model. The table containing the full results can be consulted in appendix A.1. As evaluating the reconstruction loss on the same criterion than the one used for training, we provide several spectral losses :

- *log-density*, the likelihood of the data under the generative model $p_\theta(\mathbf{x}|\boldsymbol{\mu}_p(\mathbf{z}), \boldsymbol{\sigma}_p^2(\mathbf{z}))$ (training loss)
- ℓ^2 loss, the mean-squared error between $\|\cdot\|^2$ the generated spectrum and the target
- *spectral convergence* (SC) : $\|\mathbf{x} - \hat{\mathbf{x}}\|^F / \|\mathbf{x}\|^F$, where $\|\cdot\|^F$ is the Frobenius norm of the magnitude spectra $\mathbf{x}$. This loss is mainly focused on the large frequency components [286]
- *Itakura-Saito divergence* (ISD) : a divergence specifically used for audio signals, related to perceptual divergence [287]
- *log-*ℓ^1 , the absolute $\|\cdot\|^1$ loss between the generated spectrum and the target
- *log-*ℓ^2 , the Euclidean $\|\cdot\|^2$ loss between the generated spectrum and the target

As our systems model the generative distribution $p(\mathbf{x}|\mathbf{z})$ and that most of these criteria are taking definite values, we always choose the MAP estimation, that is the mean $\boldsymbol{\mu}_p$ of $\mathcal{N}(\boldsymbol{\mu}_p(\mathbf{z}); , \boldsymbol{\sigma}_p^2(\mathbf{z})$.

Influence of dimensionality. To study the influence of latent dimensionality on reconstruction results, we train models on each dataset with different latent shapes. A short results table is given 3.1 for the `acidsInstruments-ordinario` dataset, showing the log-density (training criterion), mean-squared error, and spectral convergence obtained with the test dataset.
The influence of latent dimensionality on reconstruction can be clearly observed, decreasing sharply from 2 to stagnate around 8 dimensions. This indicates that the model requires enough space to organize the whole dataset, emphasizing the role of VAE as a compression algorithm. However, even if the performances of two dimensions are clearly lower, reconstructions shown fig. 3.15 indicate that the VAE still honourably generates the given examples, showing that it is able to compress data with a very high rate (2 vs. 1024 dimensions). We also indicate the performances obtained with our baselines (PCA and ICA provide similar results in terms of reconstructions), that are clearly outperformed by the proposed model.

Impact of regularization. The divergence used to regularize the latent space has also an impact on the reconstruction, as shown by table 3.2. As suggested by the theoretical analysis of AEVB exposed in the

Table 3.1: Results obtained on test data with the `acidsInstruments-ordinario` dataset, for different latent space dimensionalities.

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^2	SC
2d VAE	334.34	68.12	21.5
Baseline	-	104.76	16.47
4d VAE	287.0	31.93	18.21
Baseline	-	92.7	17.02
8d VAE	269.9	20.11	12.28
Baseline	-	71.54	16.79
16d VAE	276.69	23.67	20.13
Baseline	-	48.65	15.16

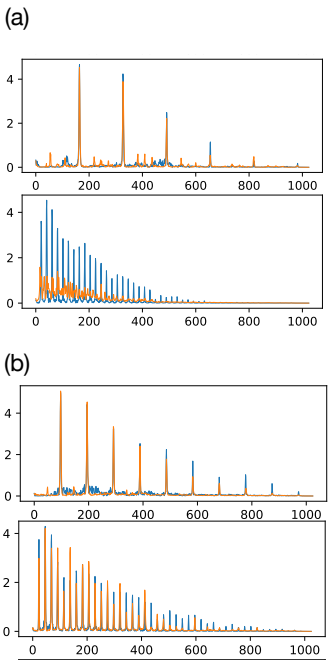


Figure 3.15: Examples or reconstructions obtained from a latent space with (a) 2 dimensions (b) 8 dimensions. While the system performs honourably with two dimensions, it cannot generate some examples.

last section, using implicit regularizations strategies (MMD, Adversarial) indeed achieve better reconstruction results, outperforming methods using explicit divergences D_{KL}, Renyi, JSD. However, this gap is lower on test examples, pointing out that implicit methods may be more subject to over-fitting. Indeed, adversarial regularization performs very well with datasets of low variability (`toy_additive_mini`, `acidsInstruments-ordinario`) but performs badly with `toy_fm`, indicating that this regularization method prevents to efficiently compress the dataset. This may be explained by the inherent discriminating of this regularization process, isolating each individual example. Constrastly, MMD regularization (similar to WAE [84], see sec. 2.3.2.3) seems to avoid this over-fitting tendency, achieving good performances in any case.
Regarding explicit regularization methods, Rényi divergence with $\alpha = 2$ seems to perform slightly better than D_{KL}, suggesting that enforcing a little the zero-avoiding behaviour of the regularization may help the system to reconstruct the examples. However, Jenson-Shannon divergence seems to performs very badly, having the worst reconstruction scores for alle the datasets. This may be due to the two-side construction of the JSD, that may be unsuitable as an optimization criterion.

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^2	SC
`additive`			
D_{KL}	488.37	13.21	11.07
MMD	474.75	6.06	6.67
$D_{\alpha=2}$	487.84	13.01	9.03
JSD	519.1379	24.36	7.60
`Adv`	485.88	10.18	10.0
Baseline	-	137.29	160.42
`fm`			
D_{KL}	910.10	340.40	15.81
MMD	766.8	221.78	62.19
$D_{\alpha=2}$	210.86	3.40e16	218.86
JSD	259.07	242.90	27.60
`Adv`	789.56	525.16	6.46
Baseline	-	859.12	527.28
`aIns-o`			
D_{KL}	276.69	23.67	20.13
MMD	248.19	8.43	5.31
$D_{\alpha=2}$	255.74	12.38	10.53
JSD	259.07	14.22	7.60
`Adv`	246.7	7.71	10.0
Baseline	-	48.65	15.16

Table 3.2: Reconstruction scores obtained for different regularization divergences, using test partition of the datasets.

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^2	SC
`toy_additive`			
$\beta = 1$	276.69	23.67	20.13
$\beta = 4$	518.12	31.43	19.46
$\beta = 10$	553.48	57.91	37.85
PCA	-	137.29	160.42
`toy_fm`			
$\beta = 1$	825.75	262.16	15.817
$\beta = 4$	897.32	316.24	15.61
$\beta = 10$	1099.0	453.92	15.97
PCA	-	859.12	527.28
`aIns-o`			
$\beta = 1$	261.37	15.34	13.74
$\beta = 4$	279.83688	27.77	16.56
$\beta = 10$	304.09	45.21	44.54
PCA	-	859.12	527.28

Table 3.3: Reconstruction scores obtained for different β, using test partition of the datasets.

Impact of D_{KL} weighting. We also evaluated the effect of the β term, that weighs the regularization term in the ELBO formulation (2.59), that can be used to control the rate of the representation (see sec. 2.16). In figure . , we can verify that increasing the rate of the representation, that is here the divergence $D_{KL}[q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))]$, reduces the quality of the reconstruction. A short table is displayed fig. 3.3, showing the results obtained with the test set. We can thus verify that rate and distortion are complementary, the quality of reconstructions decreasing as β is increased. However, as shown by the work of Higgins & al., increasing β should increase the significance of the representation, enforcing it to represent factors of variation underlying the data. We will investigate that in the following section, studying the emerging properties of the latent space.

3.2.2.2 Representation evaluation

While evaluating the quality of reconstructions is a rather easy task, evaluating the emerging properties of the obtained latent spaces is more difficult. Indeed, the relatively high amount of dimensions prevents us to get a direct visualization of the latent space, and the projections applied to PCA or ICA dimensionality reduction methods may flatten the intrinsic latent organization. Therefore, we rely on different evaluators to describe the organization of latent projections, aiming to provide additional descriptors of its topology.

Evaluation strategies. As we said, *visualizing* the space is an hard task, as the number of dimensions is relatively high and that common dimensionality reduction techniques flatten the inner organization of

the latent space (see fig. 3.16). A first possible evaluation is to measure how much the encoding distribution fits the prior using four different divergences : $D_{KL}[q\|p]$, $D_{KL}[p\|q]$, $D_{\alpha=2}[q\|p]$, $JSD[q\|p]$, and MMD. Remembering sec. 2.1.3.2, note that the four first divergences fit individual projections $q(\mathbf{z}|\mathbf{x})$, while the last one fit the aggregated posterior $q(\mathbf{z})$, and hence evaluates the global projection.

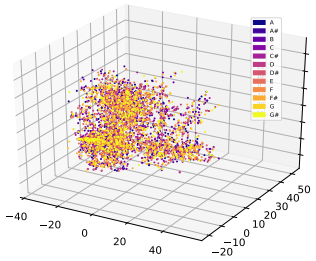


Figure 3.16: PCA representation of a VAE trained in `acidsInstruments-ordinario`, each point being a data point projection coloured by its pitch class. We can see that the extracted representation has been successfully shaped as an isotropic normal distribution, but that this visualization does not provide any information on the classwise repartition.

Another desired property of the representation is *disentanglement*, quantifying how much the axis of the obtained representation is reflecting the generative factors of variation of the dataset. However, we do not always have the required information to track the corresponding organization in the latent space. A way of measuring the disentanglement of a representation is to compute the *total covariance* (TC) of the representation, that is the D_{KL} between the full posterior distribution and the product of its components [150]

$$TC(z) = D_{KL}[q(\mathbf{z})\| \prod_i q_i(z_i))] \tag{3.7}$$

However, the computation of the aggregate posterior $q(\mathbf{z})$ is impossible, as its expression is intractable [69, 288]). A way to estimate it is to marginalize on $\mathbf{x}$, $q(\mathbf{x})$ being the empirical distribution of the data :

$$q(\mathbf{z}) = \int q(\mathbf{z}|\mathbf{x})q(\mathbf{x})d\mathbf{x} \quad = \int q(\mathbf{z}|\mathbf{x})\frac{1}{N}\sum_{n=1}^{N}\delta[\mathbf{x} = x_n]d\mathbf{x} = \frac{1}{N}\sum_{n=1}^{N} q(\mathbf{z}|\mathbf{x}_n) \tag{3.8}$$

However, the D_{KL} of total correlation is still intractable, as $q(\mathbf{z})$ is a mixture of normal distributions. Another way to estimate $TC[q(\mathbf{z})]$ is to use DRE (see 2.13), that is the density ratio between the estimated aggregated posterior and the product of its components $\hat{q}(\mathbf{z}) = \prod_{n=1}^{N} q_i(z_i)$ [289]. In that case, the contrasting distribution $\hat{q}(\mathbf{z})$ is obtained by randomly shuffling batches across the latent dimensions. Total correlation can thus be estimated with

$$\hat{TC}[q(\mathbf{z})] = \mathbb{E}_{q(\mathbf{z})}\frac{D(\mathbf{z}^{true})}{1 - D(\mathbf{z}^{false})} \tag{3.9}$$

where the discriminator D is also learned during learning. An additional whitening procedure is also proposed in [290] using PCA decomposition, such that the total covariance is rather processed in the PCA space.

While Total Covariance estimate the independence between representation axis, they do not explicitly link them to existing factors of variation of the data. In their β-VAE paper, Higgins et al. propose a validation procedure using external label information [194]. Based on the intuition that variance of latent projections should be lower when a factor is fixed, this procedure trains a linear classifier to predict the targeted labels when given the pairwise distances of their corresponding projections. While the original paper uses a data generator to perform this validation procedure, this evaluation can still be performed with the corresponding metadata. This procedure is based on the following steps :

- create K classifiers for each of the K independent factors of variations
- randomly choose a factor of variation $k \in [0, K]$
- obtain two sets $\{\mathbf{x}_1\}$ and $\{\mathbf{x}_2\}$, with a factor K fixed to given value k
- obtain corresponding latent vectors $\mathbf{z}^1$ and $\mathbf{z}^2$
- compute the average L1 norms $\mathbf{z}_{diff} = |\mathbf{z}^1 - \mathbf{z}^2|$
- train classifier K on $\mathbf{z}_{diff}$ to predict the corresponding factor k

Concomitantly, a related procedure was proposed by Eastwood & Williams, where all the input $\mathbf{z}$ is directly used to feed each factor-wise classifier. This way, we can directly obtain the responsibilities of each axis, and identify the corresponding permutation of the axis to get the desired representation [291]. A disentanglement score can then be computed using the entropy of each axis responsibility $D_i = (1 + \sum_{k=0}^{K} P_{ik} \log_K P_{ik}))$, where $P_{ij} = R_{ij} / \sum_{k=0}^{K} R_{ik}$ is the responsibility of z_i for the corresponding factor. This approach allows to evaluate the *completeness* of the representation, evaluating simultaneously the different tasks.

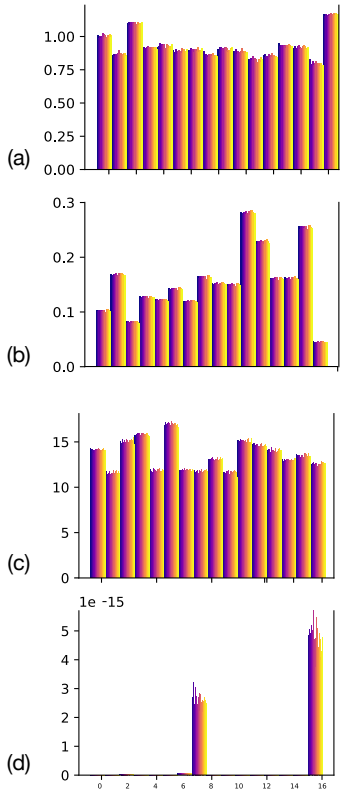


Figure 3.17: Statistics of latent projections obtained with two VAEs, one trained with D_{KL} and one trained with adversarial criterion. (a) variances of latent means, obtained with D_{KL} regularization (b) means of variances obtained, with D_{KL} regularization (c) variance of means, obtained with adversarial regularization (d) means of variances, obtained with adversarial regularization

Table 3.4: Latent organization descriptors obtained from the three datasets.

	$JSD_{0.5}[p\|q]$	$MMD[p,q]$	$TC[p,q]$
additive			
D_{KL}	104.05	159.52	22.89
MMD	521.5e9	588.83	22.67
Adv	8.81e+31	117231	211.75
Baseline	-	759.39	400.13
fm			
D_{KL}	2176	2406	65.32
MMD	9.67e27	405.44	35.98
Adv	1.13e+29	143165	409.4
Baseline	-	207162	406.33
aIns-o			
D_{KL}	59.63	35.72	23.36
MMD	65.98e12	588.83	19.35
Adv	4.37e30	471229	535
Baseline	-	39409	204.82

Impact of divergence strategies. The full results table is displayed in appendix A.2. A summarized table is provided 3.4, where we show three different latent descriptors

- the *Jensen-Shannon divergence*, that is a symmetrized version of D_{KL} that measures the average distance between $q_\phi(\mathbf{z}|\mathbf{x})$ and $(\mathbf{z})$,
- the MMD, that computes the divergence between the aggregated posterior $q(\mathbf{z})$ and $p(\mathbf{z})$
- the total covariance $TC[q(\mathbf{z})]$, obtained with the method presented in (3.9)

We can see that the divergence obtained with JSD on models trained with implicit distributions is very high. This can be observed by comparing the statistics of the obtained latent projections with two different regularizations, one with D_{KL} and the other with adversarial (see fig 3.17). In the case of D_{KL}, the variances of the variational distributions' means are very close to 1, while their variance varies between 0 and 1, showing that the space is well regularized. Conversely, the latent space obtained with an adversarial criterion is *atomic*, such that no information is shared between examples. However, the total covariance shows that the MMD regularization efficiently provides orthogonal axis, showing that it successfully matched the aggregated posterior with the prior.

Impact of β strategies. Table 3.18 show the divergence obtained when varying the β factor with three different estimators : the training criterion $D_{KL}[q\|p]$, plus two divergences $MMD[p,q]$ and $TC[p,q]$, that describe how the aggregate posterior fits the prior $\mathcal{N}(0, \mathbb{I})$. We can see that, as expected, increasing the β factor will reduce the D_{KL}; however, more surprisingly, we can see that increasing β does not decrease the TC, showing that it does not necessarily enforce the independence of

latent axis (the TC is even increasing with `toy_fm`). A further analysis of the latent statistics indeed points out that increasing β does not seem to *orthogonalize* the latent representation but rather to *polarize* it, at least for magnitude spectra. Indeed, as can be seen fig. 3.18, all the dimensions of models trained with $\beta = 1$ seem to bring an equal contribution of the latent code, while with $\beta = 10$ seems to clearly deactivate some dimensions, where the mean variances of the corresponding latent dimension z_j equals 1 (see section 2.3.2.3). Therefore, increasing the regularization weight does not spread the latent representation, but rather concentrates the information on a fewer number of axis.

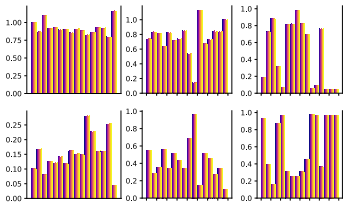


Figure 3.18: Statistics obtained from a VAE trained with $\beta = 10$. (a) variances of means obtained with D_{KL} (b) means of variances obtained with D_{KL} (c) variances of means obtained with adversarial (d) means of variances with adversarial

Disentangling. While the estimators used so far allow to measure the performance of the regularization and the latent axis independence, it does not quantify if the obtained dimensions reflect the generative factors of each dataset. We show table 3.6 the disentanglement results obtained with the dataset `acidsInstruments-ordinario`, using the Higgins' disentangling measure. To ensure the consistency of these measurements, we also perform prediction from the trained classifier on random labels, and compare the scores with the ones obtained with true labels.
While the best results are obtained, for octave and pitch, with the model with 8 latent dimensions, we can see that the difference with the results taken with random labels is little, showing that the classifier could have learn the distribution of the point-wise distances. We also see that increasing the β do not especially lead to better disentanglement measures, even worsening it in the case of the *instrument* class, that is surprisingly best disentangled by ICA. We obtain similar behaviours with other dataset, showing that the latent spaces do not automatically disentangle the generative factors of audio datasets, at least with the Higgins evaluation method.

	$D_{KL}[q\|p]$	$MMD[p,q]$	$TC[p,q]$
`additive`			
$\beta = 1$	35.20	159.52	22.89
$\beta = 4$	18.28	143.10	19.83
$\beta = 10$	12.47	140.6	20.10
PCA	-	759.39	202.13
ICA	-	8875.41	6.77e-4
`toy_fm`			
$\beta = 1$	49.20	2406	65.33
$\beta = 4$	34.77	2534	89.76
$\beta = 10$	24.87	2419.8	113.10
PCA	-	205598	405.70
ICA	-	7594.93	6.10e-4
`aIns-o`			
D_{KL}	19.28	2840.25	50.27
$\beta = 4$	9.36	2527.10	50.90
$\beta = 10$	5.423	1807.7	45.52
PCA	-	39409	204.82
ICA	-	7881.27	3.92e-4

Table 3.5: Table of results for latent descriptors obtained with models with various β. As we can see, increasing β does not lead to reduce the total covariance TC

Perceptive maps. Finally, another method to analyse the properties of the latent space is rather to explore its topology in the data domain, performing grid sampling of the latent space and computing acoustical descriptors of the corresponding generation. However, grid-sampling scales exponentially with the number of dimensions, and can then be expensive to compute. Hence, we resort to invertible dimensionality reduction algorithms (PCA and ICA) to project the latent space in three dimensions, and then obtain the *perceptual maps* for the corresponding model. These projections are computed on the latent variables

	octave	random	**pitch**	random	**instrument**	random
2d	2.52	2.46	2.72	2.74	2.74	2.72
4d	2.55	2.42	**2.41**	2.67	2.79	2.76
8d	2.51	2.47	2.60	2.65	2.75	2.65
D_{KL}	2.77	2.47	2.76	2.66	2.46	2.69
MMD	2.4	2.35	2.50	2.64	1.68	1.75
`Adv`	6.94	7.52	10.112	10.03	8.74	8.25
$\beta = 4$	2.64	2.45	2.75	2.62	2.38	2.61
$\beta = 10$	**2.33**	2.33	2.65	2.58	2.78	2.57
PCA	3.00	3.04	3.35	3.27	3.19	
ICA	2.41	2.265	2.53	2.53	**1.61**	2.53

Table 3.6: Disentanlgment scores for tasks *octave*, *pitch*, *instrument* obtained with the dataset `acidsInstruments-ordinario`.

obtained from examples of the dataset, and provide an interesting analysis of the *local* behaviour of the latent space. These perceptive maps provide intuitive feedback of the latent organization, and could be advantageously used for interactive exploration.

First, we use the *spectral centroid*, that is the weighted-mean of frequencies according to their corresponding amplitude [292]

$$\text{Centroid}[X(f)] = \frac{\sum_{i=0}^{N-1} f(i)X_i(f)}{\sum_{i=0}^{N-1} f(i)} \tag{3.10}$$

that has a connection with the perception of *brightness* of the corresponding sound. The second descriptor is *bandwidth*, that is defined [293]

$$\text{Bandwidth}[X(f)] = \left[\sum_{i=0}^{N-1} X(f) \Big(f(i) - \text{Centroid}\big[X(f)\big] \Big)^p \right]^{1/p} \tag{3.11}$$

that estimates the width of a the spectra over its frequency range, and is related to the perception of *narrowness* of a sound. Finally, we also compute the *flatness* descriptor [294]

$$\text{Flatness}[X(f)] = \frac{\exp \frac{1}{2\pi} \int_{-\pi}^{\pi} \log X(f) df}{\frac{1}{2\pi} \int_{-\pi}^{\pi} X(f) df} \tag{3.12}$$

that can be roughly described as the difference between the marginal entropy $\mathbb{E}[X(f)]$ of the spectrum distribution $X(f)$ and the Kolmogorov-Sinai entropy, that estimates the entropy of the conditional distribution $\mathbb{E}[X(f)_k | X(f)_{k-1}]$. Flatness is then related to the *noisiness* of the spectrum, where signals close to noise will verify $\text{Flatness}[X(f)] \to 0$, while signal with a more *deterministic* structure such that pure sounds will rather have high values.

An example of such maps are given fig. 3.19 for `toy_additive` and `acidsInstruments-ordinario` using PCA and ICA. These maps allow us to draw very interesting conclusions on the *perceptual* organization of this space. First, we can see that perceptual maps obtained with PCA, that lies close to the projected points, provides perceptually smooth maps, showing the regularization effect of the VAE. However, this contrasts with the maps obtained with ICA, showing rather complex latent structures. While the maps obtained with ICA are sufficiently smooth when trained with `toy_additive` with D_{KL} regularization, the ones obtained on `acidsInstruments-ordinario`, with adversarial divergence are more irregular. We can explain this by the fact that ICA, contrary to PCA, enforces the orthogonality of its axis and thus can represent untrained zone of the latent space.

Overall conclusions. These experiments then allow to draw important conclusions on how vanilla auto-encoders behave with magnitude spectrum. First, we can see that VAEs can compress the dataset very efficiently, even providing convincing reconstruction results with only

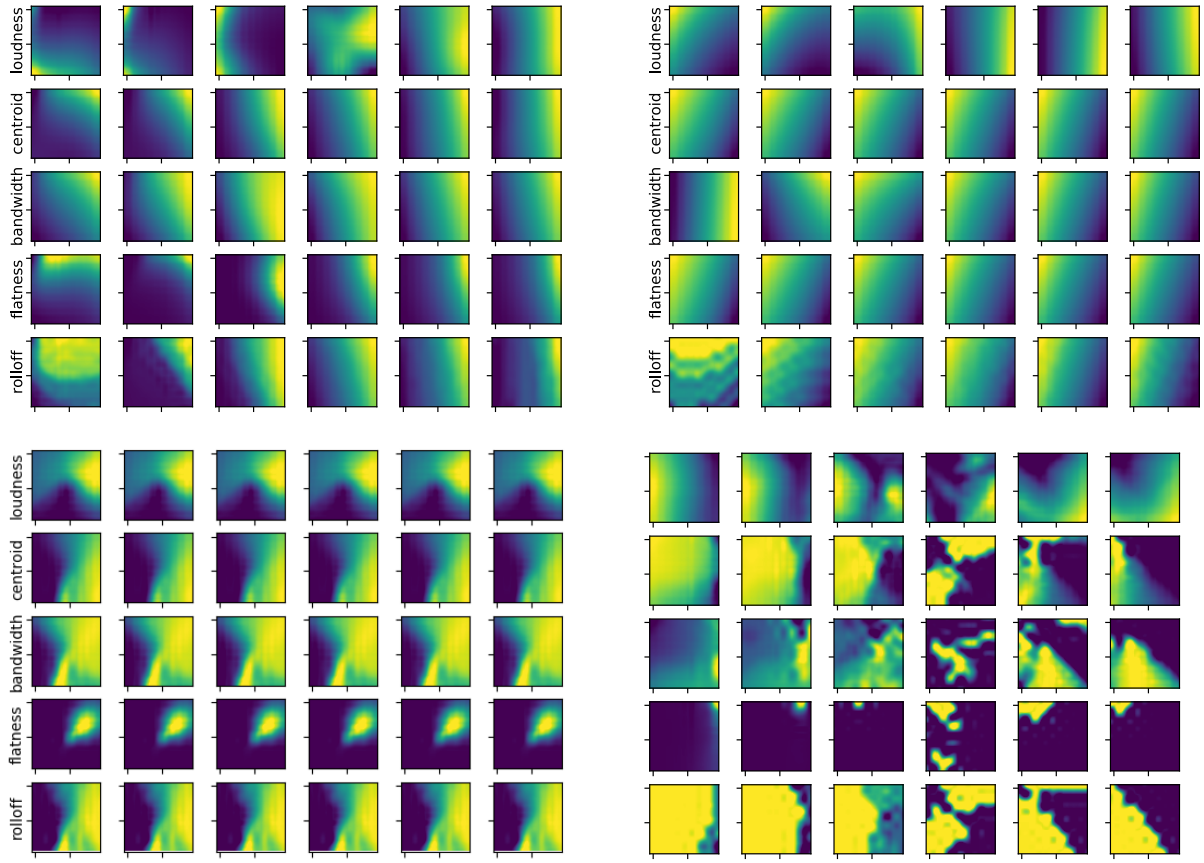


Figure 3.19: Descriptor plots obtained from (left) `toy_additive` with D_{KL}, with PCA (above) and ICA (below), and from (right) `acidsInstruments-ordinario` with adversarial, with PCA.

two latent dimensions. Regarding the impace of regularization methods, MMD and adversarial divergences allow to reduce even more the reconstruction error, at the cost of developing atomic representations that discourage information sharing between examples. We also showed that the β parameter, at least for magnitude spectra, does not encourage the orthogonality of latent axis but rather the *sparsity* of the latent dimensions, encouraging fewer active latent units at the cost of lower reconstruction abilities. However, regardless of the chosen divergence, the system manages to regularize the representation on the chosen prior, showing the efficiency of the regularization process. However, we saw that VAE *did not* disentangle inner factors of variation for audio data, at least for this standard configuration*. Further investigations of disentanglement abilities for audio signals could be pursued by taking even simpler datasets and black-box analysis approaches, such as the Riemannian approaches such as the ones proposed in [197] ; however, we leave that to future work. In this PhD, this disentanglement issues rather motivated us to develop regularization strategies specific to audio signals, relying on the flexibility of variational process. Such strategies would allow us to specify externally latent organization policies for the system, that would be beneficial from a design point of view. We thus propose two different regulariza-

* we did similar experiments with fully-connected instead of convolutional networks, and obtained similar conclusions

tion strategies: one based on *symbol-signal translation spaces*, allowing the user to define custom vocabularies that would be reflected in the latent space, and another one based on metrics matching, based on the perceptual regularization of the latent space.

3.3 Bijective signal-symbol regularization

In the previous section, we studied the emerging properties of the latent representations extracted with AEVB trained on magnitude spectra. We concluded that, while the obtained representations were able to drastically reduce the dimensionality of data, the interpretability of latent features was still difficult and not representing inner factors of variation. However, in some applications, we can access complete or partial metadata on the dataset, such that we may want to enforce the representation to reflect these properties. Furthermore, some methods take inspiration of *domain transfer* to propose multi-modal latent spaces, such that we can infer the symbolic information from the signal and reversely, using the latent space as a *translation space*. Such models can be very beneficial for the proposed approach, as this could allow us to explicitly model generation parameters without interfering too much with the unsupervised learning process of AEVB. In this section, we will summarize the most common methods to influence the latent representation with external information, from full supervised approaches such as *conditioning* to loose semi-supervised methods such as *domain translation* (see fig. 3.20). We will then propose an application of such methods to musical signals, using the latent space as signal-symbol translation spaces between some orchestral instruments and symbolic information (pitch, octave, and dynamics).

3.3.1 (Semi-)supervised methods for latent regularization

Several methods can be used to introduce external symbolic information in the learning process of AEVB models. Some approaches are based on *conditioning*, where the symbols are directly provided to the encoding/decoding modules ; however, these methods are entirely supervised, such that the model is untrainable if we miss some label information. Other methods propose to alleviate this constraint with *semi-supervised* learning, where an symbolic inference module is added to AEVB such that the system is able to recover the missing information. Finally, some methods take inspiration from *domain transfer* to provide shared representation between the data and corresponding symbols.

Conditioning. We can influence the latent representation by conditioning whether the variational or generative distributions on the label information. Given a dataset $\{\mathbf{x}, \mathbf{y}\}_{i=0...N}$ composed by data $\mathbf{x}$ and corresponding symbols $\mathbf{y}$, provided as class labels, we can *condition* the encoder by providing the label information as a additional input,

modeling a variational distribution $q(\mathbf{z}|\mathbf{x}, \mathbf{y})$. Hence, the symbol information can directly influence the latent space through the encoder. Similarly, we can also condition the decoder to model the generative distribution $p(\mathbf{x}|\mathbf{z}, \mathbf{y})$, where we also add the label information to the module's input. As the variables $\mathbf{z}$ and $\mathbf{y}$ are assumed independent, conditioning the generation model on symbols enforces the *invariance* of the latent representation to the corresponding label, then disentangling the space from the class information. The obtained latent space is thus shared between classes, and we can then *conditionally generate* from the decoder to obtain samples belonging to the target label. These models can then be used for between-class translation, encoding a sample and decoding the corresponding latent vector with another class label. This translation have been showed to also transfer the stylistic properties of the encoded data to the decoded sample, thus proving the consistency of the obtained representation [295]. Furthermore, Yan & al. showed that model was able to perform *class interpolation*, interpolating between the two one-hot vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ [296].
This method, that we call *concatenative conditioning*, allows us to efficiently obtain class-invariant latent representations. However, concatenating the symbolic information to the data may be not sufficient to accurately condition the decoder to the class information. Conditional Normalization methods rather propose to condition the normalization layers of neural networks, then to directly condition some of the network parameters to the class information. The *Conditional Instance Normalization* proposed by Dumoulin, Shlens & al., have been successfully applied to style transfer in the image processing domain, providing interesting translation results [297]. This method have then been extended by Perez, Strub & al. with Feature-wise Linear Modulation (FiLM),that uses embeddings extracted from class information to directly modulate the features provided by each layer of the network [298]. While these conditioning methods are widely used in image processing, their use in the audio domain is still marginal. However, FiLM modulators have been successfully applied to many-to-many instrument translation, showing the efficiency of these methods for audio [299].

Semi-supervised learning. Despite the efficiency of advanced conditioning methods described above, the availability of symbolic information for the whole dataset can sometimes be burdensome. We can circumvent this requirement by modeling the label information $\mathbf{y}$ as a random variable, and adding an inference model $q(\mathbf{y}|\mathbf{x}, \mathbf{z})$ to the system that is able to predict the class information in the case of incomplete data. The symbolic variables $\mathbf{y}$ can also be modelled as additional latent variables, that are extracted with a second recognition model $q(\mathbf{y}|\mathbf{x})$. The latent representation is then disentangled between class-related and class-insensitive features [300].
Alternatively, we can infer the symbol information using the latent space, running a discrimination task on a high-level representation of the data. This framework, called *semi-supervised learning*, combines the benefits of both supervised and unsupervised approaches, and allow us to train the model on data even if the corresponding label

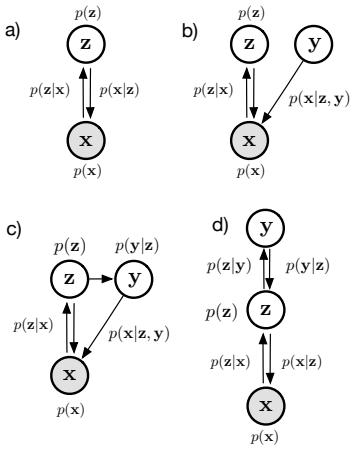


Figure 3.20: Four different approaches for conditioned data generation. (a) unconditioned system (b) *direct conditioning* : the label information is directly given to the decoder, making the latent representation independent to the corresponding task. (c) *semi-supervised learning* : an additional classification system is added on the top of latent space, predicting the label it is missing (d) *translation* : the latent space is used as a **translation space**, that is shared between labels and data

information is missing. This framework was first proposed by Kingma & al. [295], proposing a 2-layer variational model where label information was extracted from the first latent layer, and the given to the decoder for concatenative conditioning. Provided the low capacity of the discriminator (such as a simple linear classifier), the variational distribution will then be enforced to linearly separate the latent projections of the different classes. Semi-supervised learning is then also an efficient way to regularize the latent space of the variational model on external symbolic information, influencing the representation to discriminate examples belonging to different classes. However, the original model from Kingma & al. does not explicitly model the distribution $p(\mathbf{z}|\mathbf{y})$, limiting the interpretability of the process. We then take inspiration from *domain-translation*, a more general framework that can be applied for multi-modal inference.

Multi-modal domain-translation. The different approaches described in the last paragraphs can be considered as a particular case of *multi-modal learning*. Multi-modal learning addresses the problem of modeling a dataset composed by K different views of a same example $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2 ... \mathbf{x}^K\}$, that can possibly be incomplete when some views are missing. A way to extract useful features from multi-modal datasets is to infer a representation $q(\mathbf{z}|\mathbf{X}) = q(\mathbf{z}|\mathbf{x}_1) = ... = q(\mathbf{z}|\mathbf{x}^K)$ shared by the different views, such that all views $\mathbf{x}^1, \mathbf{x}^2 ... \mathbf{x}^K$ are represented by the same latent vector $\mathbf{z}$. Provided the corresponding generation models $p(\mathbf{x}_k|\mathbf{z})$, this shared representation can be then used as a *translation space* by modeling the distribution $q(\mathbf{x}_k|\mathbf{x}_m) = p(\mathbf{x}_k|\mathbf{z})q(\mathbf{z}|\mathbf{x}_m)$, using the same vector $\mathbf{z}$. Sharing latent representations then allow to perform both domain translation and direct generation, by directly sampling all the views from a latent vector $\mathbf{z}$.
Hence, a variational formulation of this setting would require a set of encoders $q(\mathbf{z}|\mathbf{x}_1), q(\mathbf{z}|\mathbf{x}_2), ..., q(\mathbf{z}|\mathbf{x}_K)$ and a set of decoders $p(\mathbf{x}_K|\mathbf{z}_1), p(\mathbf{x}_K|\mathbf{z}_2), ..., p(\mathbf{x}_K|\mathbf{z}_K)$, and formulating the overall loss by accumulating the individual ELBOs of each domain

$$\mathscr{L}_{\text{total}} = \mathscr{L}_1(q^1, p^1) + \mathscr{L}_2(q^2, p^2) + ... + \mathscr{L}_K(q^K, p^K)$$

however, this criterion does not provide any explicit transfer objective between different views. Taking two domains $\mathbf{x}$ and $\mathbf{y}$, Vedentam proposes to model both single-domain encoders $q(\mathbf{z}|\mathbf{x})$ and $q(\mathbf{z}|\mathbf{y})$ and joint encoder $q(\mathbf{z}|\mathbf{x}, \mathbf{y})$, and learn on the triple ELBO objective $\mathscr{L}[q(\mathbf{z}|\mathbf{x}, \mathbf{y})] + \mathscr{L}[q(\mathbf{z}|\mathbf{x})] + \mathscr{L}[q(\mathbf{z}|\mathbf{x}, \mathbf{y})]$ with shared decoders $p(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{z}|\mathbf{x})$ [301]. This objective ensures the symmetry of the inference process between $\mathbf{x}$ and $\mathbf{z}$, and enforces the single-domain encoders to match the joint encoder. Alternatively, the SCAN method proposed by Higgins & al. bypass the addition of a supplementary by adding a explicit transfer penalty to the ELBO, giving [302]

$$\mathscr{L}_{\text{SCAN}} = \mathscr{L}_1(q^1, p^1) + \mathscr{L}_2(q^2, p^2) + D_{KL}[q^1 \| q^2] \tag{3.13}$$

where the sense of the additional D_{KL} between the two posterior

approximations has been chosen to encourage the mass-covering behaviour of $q(\mathbf{z}|\mathbf{y})$ with respect to $q(\mathbf{z}|\mathbf{x})$, such that the symbolic posterior $q(\mathbf{z}|\mathbf{y})$ does not under-match any example $\mathbf{x}$ of the class. Adding this transfer term then allows us to directly match the latent representations of both models without any additional module, thus speeding up the process. Suzuki & al. propose to match both approaches, modeling the joint distribution and adding an explicit transfer term [303]. Liu & al. also propose to enforce even more this transfer by sharing the weights of upper layers of each encoder-decoder pairs, and use an additional adversarial criterion to evaluate the transferred data [304].

3.3.2 Signal-symbol translation with latent space matching

Motivations. Our motivation for investigating latent regularization strategies based on external symbolic information is twofold. From an analysis perspective, recovering symbols from latent spaces obtain from audio signals encompass most of MIR techniques such as instrument recognition, pitch extraction, and many others. However, most machine-learning based MIR methods are entirely *supervised*, such that the system is untrainable if the data-symbol pairs are incomplete. Furthermore, models trained from an supervised manner often provide representations that are too task-specific, such that extracted features can represent poorly the underlying structure of data and cannot be used for other tasks. Semi-supervised methods then alleviate this drawbacks, being able to both extract robust representations from data and efficiently extract the corresponding symbol information. These methods can then allow us to perform *multi-task learning*, where the latent space provide efficient representations that is enforced to represent diverse modalities of the data.
Moreover, using latent representations as a bijective *translation* space between symbols audio signals is interesting from a creative point of view. Indeed, considering symbolic information as alternative modalities of the data then allows to learn on arbitrary data/symbol pairs, and can be customized by the user during the training process. The method can thus be used for *constrained generation*, where the signal generator is conditioned on complete or partial symbol information, but also reversely for inferring the corresponding symbols from a given audio signal. Furthermore, this framework still allows free navigation in the latent space, such that both data and symbols are generated from $\mathbf{z}$, then providing useful symbolic feedback for generation.

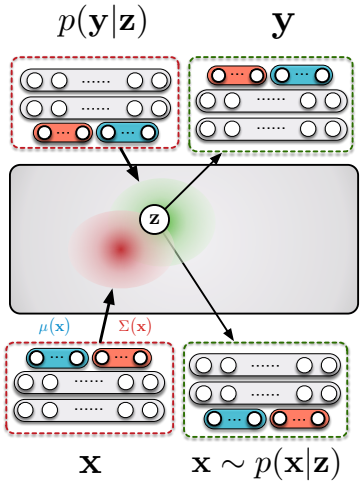


Figure 3.21: Two VAEs, one in the signal domain $\mathbf{x}$, one in the signal domain $\mathbf{y}$, that share the same representation.

Formulation. We then extend the audio data $\mathbf{x}$ with symbol labels $\mathbf{y}_1, ...\mathbf{y}_K$ to obtain our multi-modal dataset $\{\mathbf{x}, \mathbf{y}_1, ...\mathbf{y}_K\}_{i=1}^N$.

$$q(\mathbf{z}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_q(\mathbf{y}), \boldsymbol{\sigma}_q^2(\mathbf{y}))$$
$$p(\mathbf{y}|\mathbf{z}) = \prod_{i=1}^{L} \mathcal{B}(\mu_{p,i}(\mathbf{z})) \text{ or } \prod_{i=1}^{L} \text{Cat}(\boldsymbol{\mu}_{p,i}(\mathbf{z}))$$

where we model the label density with a Bernoulli distribution $\mathcal{B}(\mu_{p,i}(\mathbf{z}))$ if the label is a binary vector, or a Categorical distribution $\text{Cat}(\boldsymbol{\mu}_{p,i}(\mathbf{z}))$ if the label correspond to a given class. Note that, in the Bernoulli case, each dimension of a label information $[\mu_{k,1}, \mu_{k,2}...\mu_{k,i}]$ is independent while in the Categorical case class probabilities $[\mu_{k,1}, \mu_{k,2}...\mu_{k,i}]$ sum to one. Taking inspiration from the SCAN model proposed by Higgins & al. [302], we train our auto-encoders on the multi-modal data dataset $\{\mathbf{x}, \mathbf{y}_1, ...\mathbf{y}_K\}_{i=1}^N$ using a VAE for the signal domain $\mathbf{x}$, and another VAE on the joint label $\{\mathbf{y}_1, ...\mathbf{y}_K\}$ space, on that we train on individual ELBO plus an additional transfer term $D_{KL}[q(\mathbf{z}|\mathbf{x})||q(\mathbf{z}|\mathbf{y})]$ (see 3.13).

We propose to evaluate the performance of this framework on bijective audio-signal translation by addressing an *audio transcription* challenge, where label information is a triplet [`pitch`, `octave`, `dynamics`]. The corresponding generative distribution is then

$$p(\mathbf{y}|\mathbf{z}) = \prod_{i=1}^{K} p(\mathbf{y}_i^p|\mathbf{z})p(\mathbf{y}_i^o|\mathbf{z})p(\mathbf{y}_i^d|\mathbf{z}) \tag{3.14}$$

where the index i denotes the number of target instruments. In this case, we use a single symbolic VAE to output the concatenated symbols $[\mathbf{y}_1^p, \mathbf{y}_1^o, \mathbf{y}_1^d ... \mathbf{y}_K^p, \mathbf{y}_K^o, \mathbf{y}_K^d]$ to lighten the process, and such that sharing the encoding/decoding functions allows the VAE to internally model correlations between the different symbols. The total loss of the joint model is then, reformulating 3.13

$$\begin{aligned}
\mathcal{L}_{\text{SCAN}} =& \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}\big[\log p(\mathbf{x}|\mathbf{z}) - \beta D_{KL}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]\big] && \text{data-domain ELBO (3.15)} \\
&+ \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{y})}\big[\log p(\mathbf{y}|\mathbf{z}) - \beta D_{KL}[q(\mathbf{z}|\mathbf{y})||p(\mathbf{z})]\big] && \text{symbol-domain ELBO (3.16)} \\
&- \alpha D_{KL} q(\mathbf{z}|\mathbf{x})||q(\mathbf{z}|\mathbf{y})] && \text{weighted transfer penalty (3.17)}
\end{aligned}$$

Experiment details. We evaluate the proposed approach on two datasets. We first perform training on the proposed *acidsInstruments-ordinario* dataset, composed by individual samples from classical instruments, where the corresponding symbolic information is entirely available. In the multi-instrument detection case, we obtain the mixtures of audio signals by randomly sampling data samples over different instruments and sum the obtained spectra, provided the linearity of the spectral transform. In this study, we selected five different instruments (violin, alto-sax, flute, piano, and trumpet-C) and evaluate the model on each individual domain, and on instruments mixtures up to three different sources (saxophone + violin, and saxophone + violin + trumpet-C). We obtained the spectral representation from audio signals using an NSGT transform, based on a Constant-Q scale with 48 bins per octave. We only take the stationary part of each sound for training, that we obtain by cropping each spectra from the 20th to the 50th frame. We also evaluate our models on a validation dataset,

ensuring the robustness of obtained representation. However, datasets of single instruments recordings with corresponding annotations are uncommon, such that we could not provide validation measures for every instruments. Fortunately, Cantos & al. proposed a dataset of annotated flute recordings * gathering single notes, arpeggios, and improvisation fragments, that we can use to validate our models trained on flute sounds [305].

Regarding the models, the models used for both data and symbolic domains are simple VAEs with 32 dimensions, with MLPs modules as encoding/decoding functions. For the data domain, the MLPs are given 2 layers of 2000 units when learning single instruments, and 5000 units when learning mixtures of two instruments or above, using in both cases ReLU non-linearity and batch-normalization. For the symbol domain, networks' architectures was set to 2 layers of 1000 units, regardless of the instruments number. Both systems were jointly optimized using ADAM gradient descent with an initial learning rate of 1e-3, progressively shrunk during training depending on the loss derivative. The β weighting ot the regularization term is set to 1, with a warm-up procedure during the first 100 epochs. The α weight of the transfer loss is set to 10, as recommanded in the original SCAN article. We also compared the obtained results in symbolical inference with a baseline model that mimicking the VAE procedure from audio to data, first applying a PCA with 32 dimensions and then using a two-layer MLP for classification.

3.3.3 Evaluation of domain transfer and further applications.

Our evaluation strategy is split in three parts : we first obtain the performances achieved in data domain, both on reconstruction and transfer, and comparing the two to catch the consistency of translation space. We then do the same analysis in the symbolic domain, focusing more on transfer as the reconstruction quality of the symbolic part is quite secondary in this application. Then, we evaluate the performances of the proposed model on a validation dataset made of flute samples, and provide examples of the diverse creative applications using the proposed system. This article has been submitted and accepted at DaFX 2019 conference, and will be released soon through the arXiv plateform. The corresponding code and support page can found on github.

Audio reconstruction results. Reconstruction results are shown table 3.7, where we compare the log-likelihood and Itakura-Saito Divergence (ISD) obtained of reconstructions $-\log p(\mathbf{x}|\mathbf{z})$ and transferred data from symbols $-\log p(\mathbf{x}|\mathbf{y})$ (see 3.2). Both scores are presented for signal-to-signal reconstruction (left) and symbol-to-signal inference

* The dataset is available at : https://zenodo.org/record/1408985

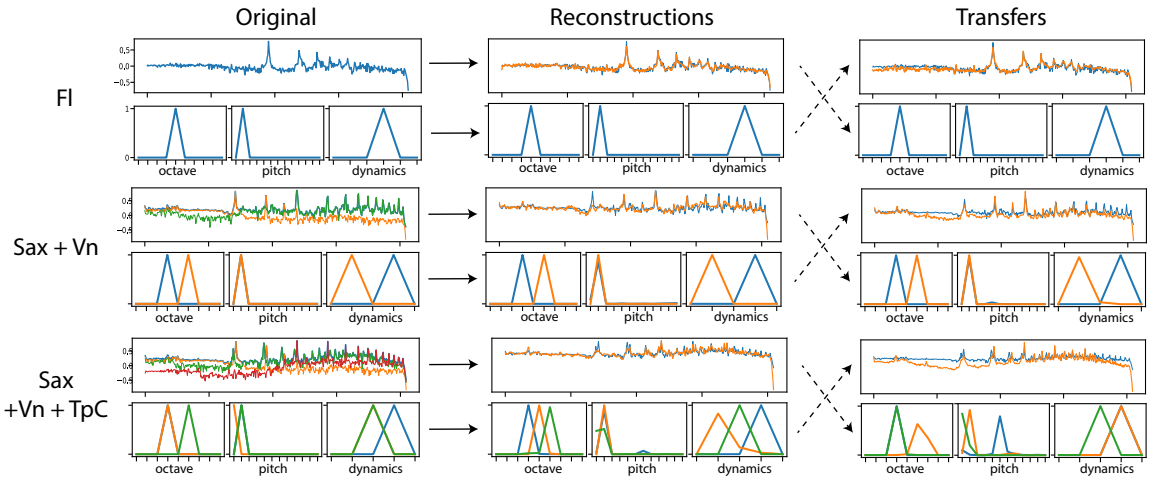


Figure 3.22: Reconstruction and transfer performed by the system. We can see that both transfer and reconstructions are perfect with one instrument. While the performances are still satisfactory with two instruments, they decrease with three instruments, showing that the system struggles to disentangle correctly the sources. This can be observed by observing the reconstructions and transfers in the symbolic domain : while the most symbols are accurately found, they are affected to the wrong instrument, showing the limitations of the followed approach.

Table 3.7: Signal reconstruction and transfer performances

	$-\log p(\mathbf{x}\|\mathbf{z})$	ISD	$-\log p(\mathbf{x}\|\mathbf{y})$	*ISD*
Alto-Sax (Sax)	-694.1	0.093	-416.6	0.177
Violin (Vn)	-671.4	0.104	-551.1	0.151
Trumpet-C (TpC)	-706.9	0.073	276.71	0.35
Flute (Fl)	-706.2	0.076	-379.2	0.147
Piano (Pn)	-813.5	0.044	-361.13	0.112
Sax + Vn	-358.71	0.364	-27.37	0.852
Sax + Vn + Fl	-268.7	0.624	692.4	3.813

(right). We can see that performances in both signal reconstruction and transfer decrease with the number of instruments, as the complexity of the incoming signal increases. Both reconstruction and signal-to-transfer scores are almost perfect in the case of solo instruments, providing convincing and high-quality sound samples generation. In the case of mixtures of two or more instruments, reconstruction scores maintain an acceptable performance, but symbol-to-signal transfer scores clearly decrease. This observation correlates with the decrease of performance observed in the symbolic domain, as discussed in the following sub-section.

Symbolical inference. Results obtained in the symbolic domain are displayed table 3.8. We provide three different evaluations : the log-density of the generated symbol, the classification ratio performed by the symbolical generator, and then the corresponding classification scores obtained from our baseline. The results within parenthesis are the scores achieved by transfer, when available. In the multi-instrument case we also provide a *loose classification ratio*, that consider a given label correct regardless of the corresponding instrument.

In the single-instrument case, both generation and transfer scores are near to perfection, proving the efficiency of the proposed method. Our model also clearly outperforms the baseline model, especially with dynamics. However, the performances degrade with the number of

		$-\log p(\mathbf{y}\|\mathbf{z})$	Success Ratio (%)	*loose* (%)	Baseline (*loose*)
Sax	p	-1.0 *(-1.0)*	100% *(100%)*	-	*94%*
	o	-1.0 *(-1.0)*	100% *(100%)*	-	*97%*
	d	-1.0 *(-1.0)*	100% *(100%)*	-	*46%*
Vn	p	-1.0 *(-1.0)*	100% *(100%)*	-	*89.8%*
	o	-1.0 *(-1.0)*	100% *(100%)*	-	*99.0%*
	d	-1.0 *(-1.0)*	100% *(100%)*	-	*35.3%*
TpC	p	-1.0 *(-1.0)*	99.9% *(100%)*	-	*76.1%*
	o	-1.0 *(-1.0)*	100% *(100%)*	-	*99.8%*
	d	-0.998 *(-1.0)*	99.7% *(100%)*	-	47.8%
Fl	p	-1.0 *(-1.0)*	100% *(100%)*	-	*52.4%*
	o	-1.0 *(-1.0)*	100% *(100%)*	-	*81.8%*
	d	-1.0 *(-1.0)*	100% *(100%)*	-	*41.4%*
Pn	p	-1.0 *(-1.0)*	100% *(100%)*	-	51.6%
	o	-1.0 *(-1.0)*	100% *(100%)*	-	63.9%
	d	-1.0 *(-0.999)*	99.9% *(100.0%)*	-	40.0%
Sax + Vn	p	-0.534 *(-0.871)*	54.0% *(87.9%)*	62.6% *(81.6%)*	*65.3%*
	o	-0.782 *(-0.980)*	84.6% *(99.2%)*	94.9% *(88.7%)*	*79.1%*
	d	-0.712 *(-0.939)*	74.4% *(95.9%)*	82.4% *(66.3%)*	*52.0%*
Sax + Vn + TpC	p	-0.381 *(-0.725)*	38.6% *(75.0%)*	62.6% *(84.5%)*	*56.6%*
	o	-0.377 *(-0.641)*	42.4% *(67.8%)*	79.3% *(88.7%)*	*62.3%*
	d	-0.347 *(-0.616)*	34.6% *(62.4%)*	66.9% *(69.5%)*	*41.2%*

Table 3.8: Symbolic inference reconstruction and classification results (successively pitch, octave and dynamics). Scores without parenthesis are reconstruction scores obtained within the symbolic domain, while scores in parenthesis are obtained when performing transfer from the signal domain

involved instruments, being still acceptable in the two-instruments mixtures case but clearly insufficient with more. This can be explained by observing the loose ratio classification scores : indeed, the results are significantly increased if we compare the obtained symbols regardless of the instrument, even outperforming the baseline. This shows that the VAE is facing a *permutation* problem, where the accurate symbols are retrieved from the mixture but given to the wrong instrument. This issue then prevent the actual formulation of the problem to be applied on instrument mixture ; we leave that to a future work, restraining ourselves for the moment to monophonic signals.

Likelihood	ISD	Class. Ratio	Baseline
2648 (*1057*)	1.065 (*0.632*)	65.4% 81.9%	63.8% 76.8%

Table 3.9: Results obtained with the validation flute dataset.

Validation results. We then evaluate the performance of our approach on a validation dataset, to challenge the robustness of the translation processed by our method when facing unknown data. This validation step is important, as checking if the system still performs accurately on different instruments and/or recordings ensures that the extracted features are meaningful. To this end, we take the annotated flute recordings dataset given by Cantos & al. [305], and evaluate the performance of our model trained on monophonic flute signals from the *acidsInstruments-ordinario*. As the annotations provided by the dataset are given in a MIDI format, we then extracted a piano-roll representation that we converted into a symbolic activation matrix, where we sliced the time scale accordingly to the time positions of the corresponding audio spectrum. Unfortunately, dynamics annotations were not provided, so we could not evaluate the robustness of the corresponding inference. Results are shown table 3.9, among with those obtained with the corresponding baseline.

We can observe the achieved performances are still decent, proving that our model is able to generalize on new datasets. More interestingly, the performances obtained in the signal domain are better in transfer

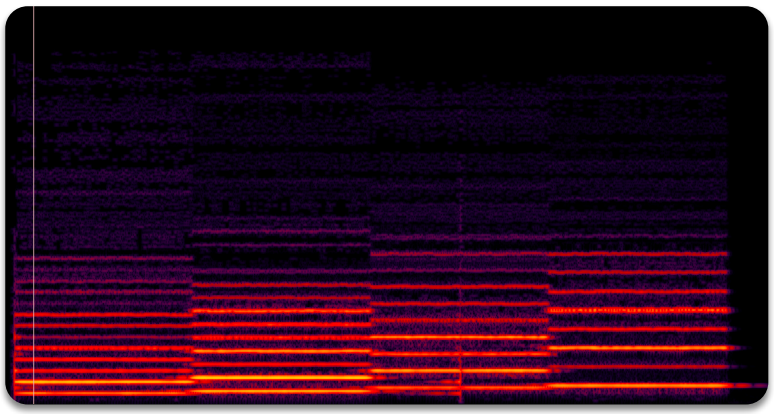


Figure 3.23: Reconstruction obtained from a MIDI file converted in audio with our model, trained on flute sounds.

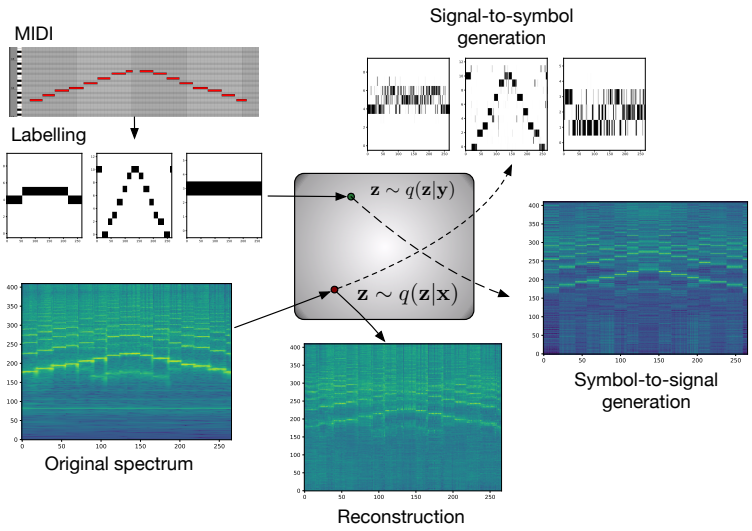


Figure 3.24: Functional graph of the symbol-signal translation system.

than reconstruction, showing that our system successfully managed to "abstract" the learned symbolic information. Also, the obtained results have to be contextualized on two points. First, the reconstruction results in table 3.9 have been obtained by forwarding full sounds into the model, while the system was only trained on the stationary parts of the recording. Then, an analysis of spectrograms showed that the flute used in the validation dataset presented a stronger octave-lower harmonic that is not present in the *acidsInstruments-ordinario* dataset, and may explain the drop in octave classification. However, these problems have to be alleviated in order to develop robust signal-symbol translation with this method, with more complete datasets involving several instruments, and training on the full range of the signal. Furthermore, a more complete approach would be to model full sounds rather than single spectral bins, and then model more accurately temporal features such as dynamics ; this precise point motivates the models developed in the chapter 4 of this PhD.

Creative applications. In addition to allow efficient multi-modal modeling, this system allows additional creative application than with uni-modal variational auto-encoders. Indeed, this system then allows

the user to use custom symbolic information to regularize the latent space, and can be then provided explicit control parameters for the generation of feedback during free exploration. The smooth regularization method performed by domain translation then opens up the system to novel possible interactions that direct supervision methods would not allow. We summarize below the new features allowed by using such framework :

- **free trajectory**: we can directly sample the latent spaces to generate the corresponding spectrum. The symbolic decoder can thus been used the provide interesting feedback to the user, relaxing the lack of interpretability of the latent representation. Examples are provided
- **sequence generation**: the navigation in the latent space can be constrained by an external sequence of symbols, such that the user can navigate the distribution $q(\mathbf{z}|\mathbf{y}$ while specifying some of the attributes of the target sounds (see fig. 3.23)
- **spectral morphing**: we can retrieve smooth interpolations between two sounds or two symbols in the latent space, then achieving a *spectral morphing* between these two sources.

Fig. 3.24 provides a graphical summary of the interactions allowed by the system, with example taken from the validation flute datasets. Audio examples of these different generation methods are available on the support page.

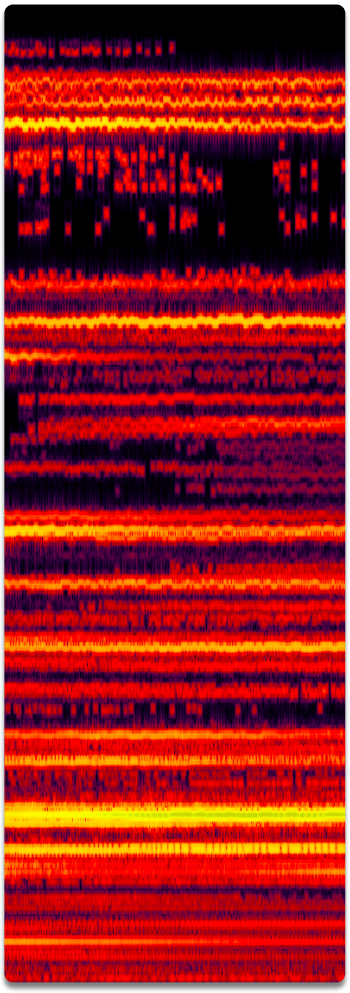


Figure 3.25: Example of random trajectory navigating the latent space obtained from a model trained on violin.

3.4 Perceptual regularization of orchestral instruments

As we saw in section 3.2, the latent representation provided by the AEVB when trained on *ordinario* orchestral instruments is not immediately interpretable, as pointed out by the obtained perceptual maps 3.19. This difficult interpretability can not only make the direct exploration of the latent space unsatisfactory, but also limit the use of the obtained representation as an high-order feature space for analysis. Hence, we may need an additional regularization constraint to enforce the space to match properties of the human perception, making this exploration more intuitive and the representation more likely to encode relevant features about the audio content of incoming signals. To this end, we propose in this section a regularization based on *timbre*, a seminal notion in audio perception.

3.4.1 Audio perception and timbre spaces

The notion of timbre is central in numerous music-related domains such as composition, orchestration, musical analysis, instrument making, or digital synthesis. However, giving a formal definition of timbre is a hard task, as this notion is intrinsically linked tp human perception and then vary significantly among individuals and musical practices.

However, the discovery of digital audio synthesis in the early 60s enabled an experimental approach to timbre understanding, underlining the connection of between the perceived timbre properties of a signal and its spectral envelopes, as investigated by Risset & al. [2] and Grey & al. [306]. The link between spectral content and timbre thus motivated the development of quantitative descriptors based on spectral representations, hand-crafted to correspond given timbre properties such as *brillance*, that can be described using *spectral centroid*, or the *attack quality*, that can be evaluated using slope time estimation [307]. Alternative methods rather derive descriptions for the *cepstrum* of a signal, that takes the inverse Fourier transform of the spectrum logarithm of the signal, estimating the variation rate [308]. Cepstral analysis thus provide an interesting representation of the spectrum envelope and led to very compressed timbre representation, such as Mel-Frequency Cepstrum Coefficients (MFCC) that are still widely used in speech processing and timbre analysis [309, 310]. Timbre analysis being an interesting bridge between digital signal processing and audio perception, this field has then been extensively investigated and provided an impressive number of different features, up to 300 as MIR toolbox [311].

While the development of these audio descriptors is helpful to evaluate timbre properties of a given signal, we also have to ensure that the proposed descriptors correspond to actual perceptual properties of the human hearing [312, 313]. Such psychological approaches then study the consistency of these descriptors on perception by organizing perceptive tests, aiming to find a correlation between the proposed descriptor and the target timbre property [292, 314]. However, an issue raised by such approaches is the problem of *vocabulary*, that can vary a lot depending on the social and psychological context of the human subject. A way to remove this bias is to rather obtain measures of global dissimilarity of sounds, and then analyse the respective pairwise similarity distances between the different types of sounds [315]. This approach is very interesting has it subsumes the idea of an underlying *perceptive measure* behind similarity judgement, opening the idea of a *continuous space* behind human perception.

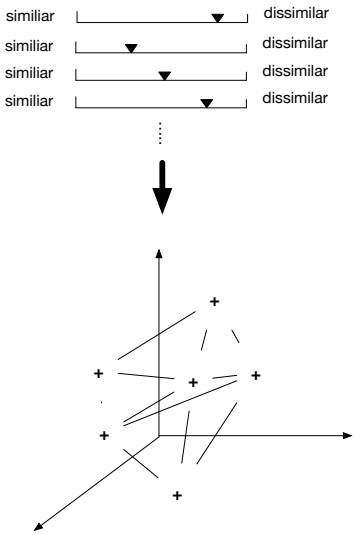


Figure 3.26: MDS can be leveraged to obtain an Euclidean space, whose distances are enforced to matching a given set of dissimilarity measures.

3.4.2 Defining perceptual regularization and similarity space matching

Timbre spaces from similarity matrices. Dissimilarity measurements then led to the idea of *timbre space* equipped from a similarity measure, such that two near points correspond to perceptually similar sounds while two distant ones would be very contrasting. Such spaces can be obtained using *Multi-Dimensional Scaling* (MDS), a method that project a set of points in an Euclidean space provided the pairwise distances between the points [110]. Remembering section 2.2.1, given N points and the corresponding pairwise dissimilarity measures $d_{n,m}$ between the points x_n and x_m, performing MDS comes back to the optimization

problem

$$\min_{x_1 \dots x_N} \sum_{i \leq j} (\|x_i - x_j\| - d_{i,j})^2 \tag{3.18}$$

to obtain a space $x_1 \dots x_N$ that minimizes the mean-squared error between the dissimilarity measures and the Euclidean measures of each pair of points (see fig.3.26). MDS then allows to extract continuous similarity spaces from human dissimilarity measurements, then providing valuable analysis spaces where the relative properties of acoustical instruments could be easily visualized [316, 317]. Timbre spaces obtained from dissimilarity measurements have been first propose by Grey, using 16 samples from classical instruments rated by 22 subjects over a scale from 0 (similar) to 1 (dissimilar). Consequently, similar experiments were conducted by Krumhasnl [318] using 21 instruments sounds rated by 9 subjects using a discrete scale, Iverson & al. [319] with 16 samples and 10 subjects that was specifically targetting the role of *dynamics* in timbre perception, McAdams & al. with 18 instruments and 24 instruments [320], and Lakatos & al. with 17 subjects and several sets of instruments [321].
In these studies, the dimensions obtained from MDS were generally semantically annotated after visualization and qualitative analysis of the space, as the *brightness-roughness* two-dimensional timbre space obtained by Grey or the spectral centroid / spectral flux / rise-time space obtained by McAdams & al [320]. A strength of this approach is then to extract continuous timbre spaces directly from similarity measurements, without further structural assumptions on the signals nor using specific vocabularies that could bias the experiments. However, while these spaces are very interesting for analytic purposes, they are also limited because of their static construction, implying the re-computation of the MDS for each additional input data, and does not allow *perceptual inference* of external sounds nor direct generation from the inferred timbre space.

Latent perceptual regularization. As we saw in the previous sections, perception of timbre is strongly correlated to the spectral content of the perceived signal. We could then beneficially use the latent spaces extracted with AEVB from classical instrument sounds to infer their perceptual properties. AEVB framework would then allow not only to infer the perceptual properties of instruments unknown from the dissimilarity measurements, but also to generate new spectra directly from the timbre space, that would be regularized with perceptive criteria. To this end, we thus have to find a regularization strategy to enforce our latent space to represent whether the pairwise perceptual dissimilarity between two encoded latent vectors, whether more global properties of the timbre space obtained with MDS. To this end, we propose different methods to enforce the geometry of the latent space $\mathbf{z}$ to match some properties of the similarity space $\mathbf{T}$ obtained with MDS. To our knowledge, we proposed the first regularization strategies based on external metric constraints.

Prior regularization. The idea underlying prior regularization is to encode the perceptual information given by the MDS directly in the prior

$p(\mathbf{z}$ of the latent space, then replacing the uninformative isotropic normal $\mathcal{N}(\mathbf{0}, \mathbb{1})$ by an informative prior $\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_i, \boldsymbol{\sigma^2}_i)$ where the mean $\boldsymbol{\mu}_i$ and variances $\boldsymbol{\sigma^2}$ are conditioned by some information relating the target instrument. Therefore, we model the distribution of each instrument in the dissimilarity MDS with a normal distribution $\mathcal{N}(\mu(\mathscr{C}_i), \sigma^2(\mathscr{C}_i))$, where the mean $\mu(\mathscr{C}_i)$ is the average and $\sigma(\mathscr{C}_i)$ is the estimated variance of the same corresponding projections. We then condition the prior $p(\mathbf{z})$ on the instrument class during training, then regularizing the inference distribution $q(\mathbf{z}_i|\mathbf{x}_i)$ on the corresponding distribution $p(\mathbf{z}_i)$. This regularization allows to directly embed the information given by the perceptive space $\mathbf{T}$ in the ELBO, and then naturally integrates into the Bayesian formulation of the AEVB.

ℓ^2 **regularization**. In a similar way to the MDS criterion, the ℓ^2 strategy regularizes the distance of two inferred latent positions $\mathscr{D}(\mathbf{z}_i, \mathbf{z}_j)$ on their distance in the perceptive space $\mathscr{D}(\mathscr{T}_i, \mathscr{T}_j)$. We thus add another criterion $\mathscr{C}(\mathbf{z})$ to the ELBO $\mathscr{L}(q)$ (2.18)

$$\mathscr{L}_{\ell^2}(q) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\, p(\mathbf{x}|\mathbf{z})] + \beta D_{KL}[q(\mathbf{z}|\mathbf{x}), p(\mathbf{z})] + \alpha \mathscr{C}_{\ell^2}(\mathbf{z}) \tag{3.19}$$

where $\mathscr{C}_{\ell^2}(\mathbf{z}) = \|\mathscr{D}(\mathscr{T}_i, \mathscr{T}_j) - \mathscr{D}(\mathbf{z}_i, \mathbf{z}_j)\|^2$, such that this regularization enforces distances in the latent space to match those of the perceptual space using mean-squared error. The α hyper-parameter is the weight controlling the perceptive regularization weight. However, this criterion does not model the uncertainty over the perceptual distances $\mathscr{D}(\mathscr{T}_i, \mathscr{T}_j)$, and thus may be subject to over-fitting.

Gaussian regularization. Extending the idea of ℓ^2 regularization, we rather assume the distances $\mathscr{D}(\mathscr{T}_i, \mathscr{T}_j)$ as drawn from a normal distribution $\{\mathscr{D}^{\mathscr{T}}_{i,j}\}_{inst} \sim \mathcal{N}(\mu_{i,j}, \sigma_{i,j})$, where the parameters $\mu_{i,j}$ and $\sigma_{i,j}$ are estimated on the mean and variances of inter-class perceptual distances. This regularization method thus model the uncertainty of the distances $\mathscr{D}(\mathscr{T}_i, \mathscr{T}_j)$. We similarly add this criterion to the ELBO 2.18

$$\mathscr{L}_{\mathcal{N}}(q) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\, p(\mathbf{x}|\mathbf{z})] + \beta D_{KL}[q(\mathbf{z}|\mathbf{x}), p(\mathbf{z})] + \alpha \sum_{i,j} p(\mathscr{D}(\mathbf{z}_i, \mathbf{z}_j)|\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2) \tag{3.20}$$

Student-t regularization. Pushing further the idea of Gaussian regularization, with the Student-t regularization we also target to model the uncertainty of both mean and variances of the normal distribution $\mathcal{N}(\mu_{i,j}, \sigma_{i,j})$. To this end, we then replace this distribution by a Student-t distribution and, taking inspiration from t-SNE, (see sec. 2.2.1) we rewrite the density probability of the perceptual distance as

$$\mathscr{D}(\mathscr{T}_i, \mathscr{T}_j) = \frac{(1 + \|\mathbf{z}_i - \mathbf{z}_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|\mathbf{z}_i - \mathbf{z}_k\|^2)^{-1}} \tag{3.21}$$

and the density probability in the latent space as

$$\mathscr{D}(\mathbf{z}_i, \mathbf{z}_j) = \frac{\exp\left(-\|\mathbf{z}_i - \mathbf{z}_j\|^2 / 2\sigma^2\right)}{\sum_{k \neq l} \exp\left(-\|\mathbf{z}_i - \mathbf{z}_k\|^2 / 2\sigma_i^2\right)} \tag{3.22}$$

we then formulate the additional objective $\mathscr{C}_{tsne}$ as the D_{KL} between these two distributions, then providing the ELBO

$$\mathscr{L}_{\mathcal{N}}(q) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\ p(\mathbf{x}|\mathbf{z})] + \beta D_{KL}[q(\mathbf{z}|\mathbf{x}), p(\mathbf{z})] + \alpha \sum_{i,j} \mathscr{D}(\mathbf{z}_i, \mathbf{z}_j) \frac{\mathscr{D}(\mathbf{z}_i, \mathbf{z}_j)}{\mathscr{D}(\mathcal{T}_i, \mathcal{T}_j)} \tag{3.23}$$

The three first regularization strategies were accepted to ISMIR2018 conference, where it gained the best presentation award. The further results regarding the student-t regularization, that obtained the better results, have been also accepted in DaFX2018. The corresponding paper is availbale on arXiv* [322].The code and corresponding page can be found here : https://github.com/acids-ircam/variational-timbre/.

3.4.3 Evaluation, perceptual inference and descriptor-based generation

3.4.3.1 Experiment details.

For this experiment, we then had not only to get a suitable set of audio data, but also the corresponding similarity ratings. In this section we describe precisely the chosen experimental protocol, and evaluate the performance of both the AEVB and the effect of the perceptual regularization, comparing the different methods proposed in the last section.

Dissimilarity ratings. For this study, we gathered a collection of perceptual dissimilarity measurements from the different sources cited in the previous sections. We ensured the statistical consistency of the overall ratings by only selecting the instruments that were fairly represented across the different studies, and normalized their respective similarity evaluation scales to get uniform ratings between 0 and 1. Despite these constraints, we managed to obtain a consistent set of 12 instruments (Piano, Cello, Violin, Flute, Clarinet, Trombone, French Horn, English Horn, Oboe, Saxophone and Trumpet), for a total amount of 11845 ratings coming from an overall number of 1217 subjects. We then perform a MSD over d dimensions on these normalized perceptual ratings, whose PCA can be visualized fig. 3.27. We can see that the obtained representation is perceptually consistent, gathering family of instruments together (brass, woodbrass, strings, and piano as an outlier), and resemblant to the timbre spaces obtained from previous studies.)

* 322.

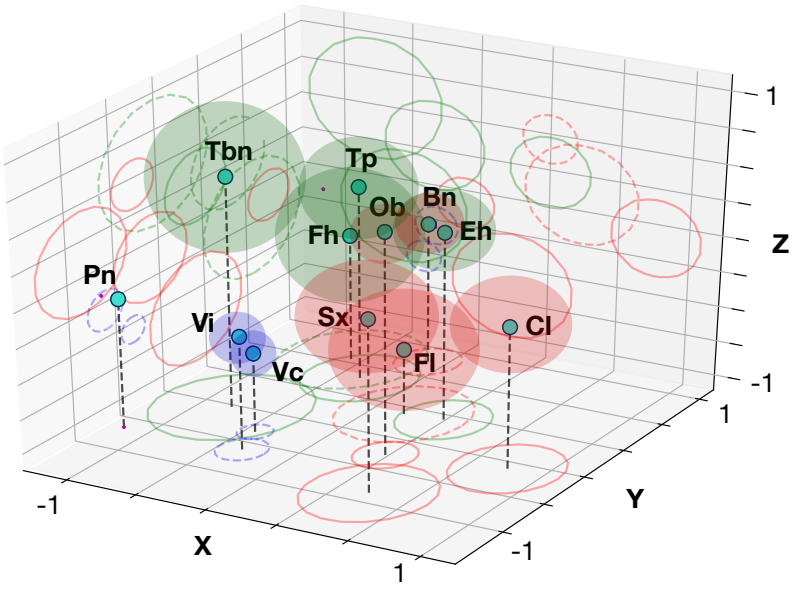


Figure 3.27: Obtained perceptual spaces from 11845 dissimilarity measures, representing averages human perceptive distances between orchestral instruments.

Audio datasets. As the dissimilarity measurements have been obtained on classical instruments, we then used the *acidsInstruments-ordinario* dataset collected from the Studio On-Line database (see sec.) as audio dataset. Indeed, this database contains samples of all the 12 required instruments over the full range of their respective tessiture, and additionaly provides for each three dynamical ranges (*pp*, *mf*, and *ff*). Furthermore, as within the scope of this study we only train on individual spectral bins, the amount of data available for each pitch and instruments is comfortably sufficeint, ensuring enough variability to overcome over-fitting. The total number of sample then amounts to 2,200 samples, that we randomly split across notes to obtain two different datasets for training and testing, with a respective balance of 90%-10%. Moreover, the additional instruments contained in this database can be projected in the extracted timre space, allowing us to evaluate the *perceptual inference* performance of our models.. Regarding the transformation, we trained our models on four different transforms, computed on signal resampled at 22050Hz. First, we used STFT and DCT time-frequency representations, using a Hamming window of 40ms and an hop size of 10ms. We then used NSGT on three different scales: a Constant-Q scale, using 48 bins per octave, a Mel scale, and an ERB scale of 400 bins, defined over a range form 30 to 11000 Hz. As training is performed on spectral magnitude, we ued Griffin-Lim procedures to reconstruct the corresponding phase.

Models. In this experiment we used a simple VAE with 64 latent dimensions, using 3-layer MLP with 2000 hidden units. The β weight of the D_{KL} regularization was set to 2 to encourage the rate of the representation, scheduled with a *warm-up* procedure on the first 100 epochs. The system is jointly trained with an ADAM gradient-descent algorithm, setting the learning rate to 1e-5. We adopted a two-step training procedure, such that the system is trained without perceptive regularization during the first 5000 epochs. This schedule enforces

the perceptual regularization to adapt to the unsupervised features extracted from the data, ensuring that the additional criterion does not interfere too much with the learning process. We further observed that this two-step procedure was mandatory to the success of the procedure.

3.4.3.2 Evaluations.

Generative evaluations. We evaluate the generation abilities of the trained systems with two different criteria : we first evaluate the likelihood of the encoded example over the generative model $p(\mathbf{x}|\mathbf{z})$, and then the mean-squared error of the encoded signal with the mean of the generated distributions. While the first allows us to validate the probability of the generated signal given a certain uncertainty, the second allows us to evaluate the precision of the maximum a-posteriori estimation of the targeted signal. We also compared the obtained results with two baseline PCA and unregularized auto-encoder, whose architecture can be compared with the proposed model. For clarity purposed, we divided the results table in two : in table 3.10 are written the results obtained across the different regularization strategies, while the results obtained across different transforms are shown fig. .

The results table 3.10 show that methods based on Bayesian formulations, that are the *prior* and *student-t* regularizations, achieve the better likelihood results. Conversely, the ℓ^2 regularization method is providing the worst likelihood performance, but provide the best reconstruction over the mean-squared error criterion. As will be confirmed in the following sections, these results are coherent with the uncertainty trade-off of Bayesian methods : as ℓ^2 does not model the uncertainty of the perceptual distances it performs better on individual examples, but has the worse performances when the variance of the generated distribution $p(\mathbf{x}|\mathbf{z})$ is taken into account. However, we can see that the reconstructions obtained across the different regularization strategies are all satisfactory.

Method	$\log p(\mathbf{x})$	$\lvert\mathbf{x} - \tilde{\mathbf{x}}\rvert^2$
PCA	-	2.2570
AE	-1.2008	1.6223
VAE	-2.3443	0.1593
Prior	-2.7143	0.1883
ℓ^2	17.8960	**0.1223**
Gaussian	0.2894	0.1749
Student-T	**-2.8723**	0.1610

Table 3.10: Generative capabilities evaluated on the log likelihood and reconstruction error over the test set.

Latent space evaluation The observation of the respective latent for each representation confirm the assumptions developed in the prece-

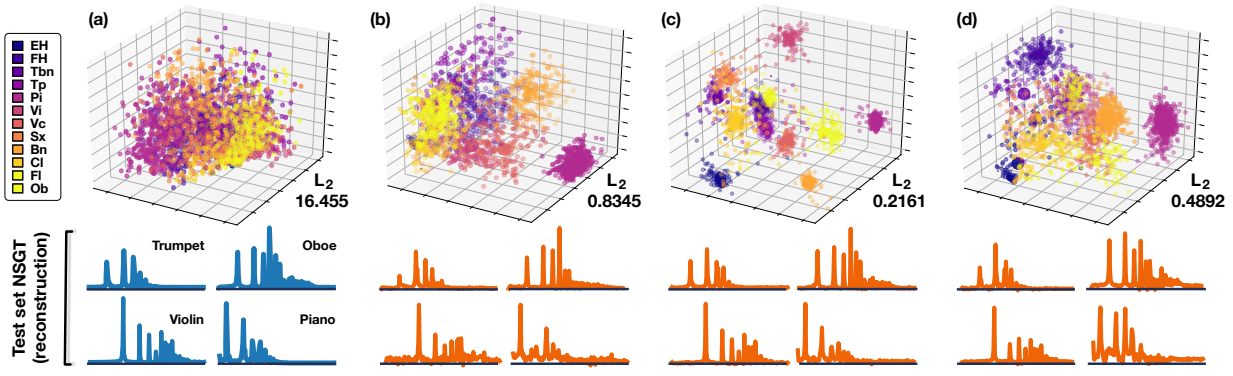


Figure 3.28: PCAs of perceptually regularized latent spaces for the different regularization techniques (a) un-regularized (b) *prior* (b) ℓ^2 (d) Student-*t*

Table 3.11: Discriminative capabilities in classifying *family*, *instrument*, *pitch* and *dynamics* of the test set.

Method	Family	Instrument	Pitch	Dynam.
PCA	0.790	0.697	0.167	0.527
AE	0.973	0.957	0.936	0.597
VAE	0.978	**0.993**	0.963	0.941
Prior	0.975	0.991	**0.993**	0.936
Euclidean	0.972	0.990	0.990	0.943
Gaussian	**0.982**	0.991	0.989	**0.948**

dent paragraph. Indeed, the over-fitting behaviour of the ℓ^2 is clearly visible, the latent projections being shrunk towards the instruments centroid. Conversely, regularization methods modeling the uncertainty of the perceptual distances and are then providing smoother distributions, a the cost of slightly reducing the MSE performance. Moreover, the prior regularization indicates an *over-regularization* effect, that can be taken similar to the results obtained with vanilla VAEs taking non-informative prior. The results obtained with Gaussian regularization naturally lie between both methods, as it assumes a distribution over the distance spaces but without modeling the uncertainty of its parameters. However, regardless of the chosen regularization, we can see that the structure of the perceptual MDS is successfully enforced onto the latent spaces, proving the validity of our approach.

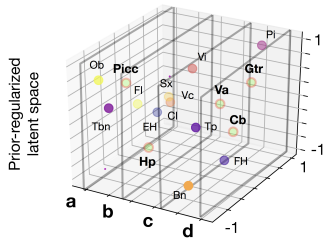


Figure 3.29: Perceptive features inferred by the model from unknown instruments.

We further evaluate the representational properties of the extracted latent space by training low-capacity discriminators over a bunch of tasks corresponding to inner criteria of variation of the data : *pitch*, *dynamics*, *instrument*, and *family*. The discriminators all have the same architecture, composed from a single-layer network of 512 ReLU units with batch normalization and softmax non-linearity. Corresponding results are provided in table 3.11. We can see that the performances of every proposed methods are highly satisfactory, and significantly gain from the proposed baselines, especially on dynamics. This indicates that the VAE successfully use its latent representation to express the variety of data, while providing consistency across these different tasks to make the problem linearly separable.

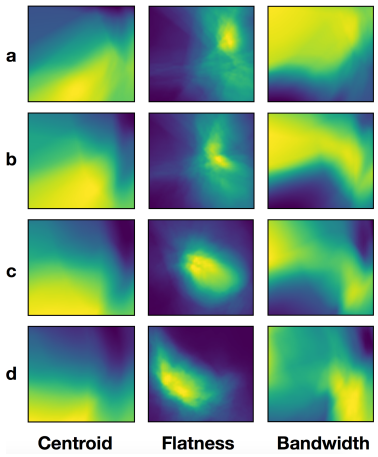


Figure 3.30: Perceptual maps obtained from PCA planes of the model.

Perceptual inference. Unlike pure MDS approaches to timbre space extraction, the proposed models allow to project signals unknown to the model onto the extracted perceptual space, allowing us to achieve *perceptual inference*. We then selected samples from the *acidsInstruments* database not present neither in the training nor test sets (Contrabass, Guitar, Harp, Piccolo, Viola), and projected them onto the perceptual space. We can see that the inferred centroids of the obtained projections are consistent, each instrument being close to ones with similar timbre properties : the piccolo lie in the wind part of the space, close to flute and oboe, viola and contra-bass are projected close, and the guitar is projected close to piano. The location of harp however seem a little hazardous, lying between trombone and english horn. Further investigations are then required to describe furthers the capacities of the model.

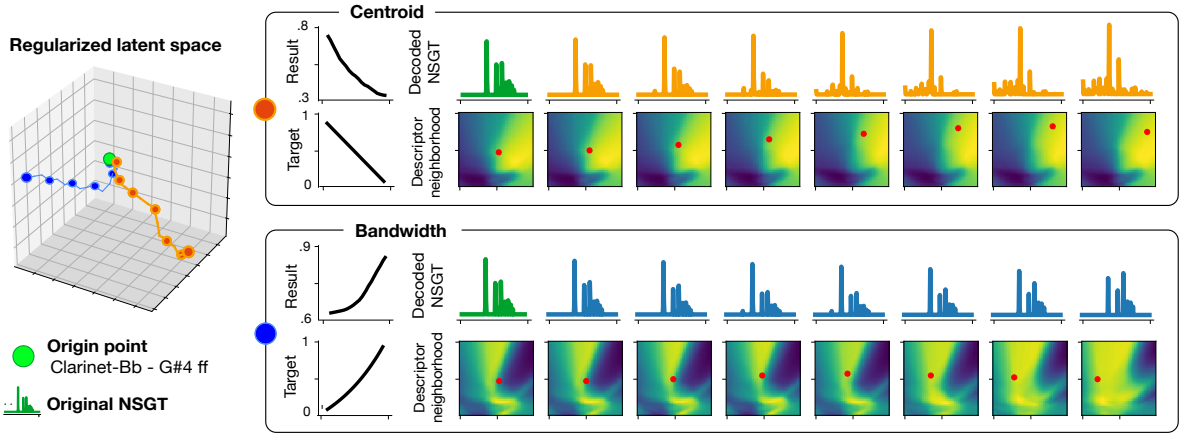


Figure 3.31: Example of descriptor synthesis

Topology of audio descriptors. As summarized section 3.4, some perceptual characteristics of sound have been successfully correlated with *audio descriptors*, that are DSP measures applied on a given slice of spectrum. We can then evaluate the global perceptual properties of the extracted representation by studying the repartition of these descriptors across the latent space. Similarly to section 3.2.2.2 , we generate the spectral distributions from the space following a grid sampling procedure of the PCA showed in 3.29, and then compute the corresponding descriptors to obtain a *topology map* of the provided representation. Here we provide topology maps of three different descriptors, whose correlation with definite perceptual attributes ws confirmed by human tests : *spectral centroid*, *spectral bandwidth*, *spectral flatness*, and *roll-off*.

The obtained maps are shown fig 3.30, where we split the space in four planes among the x axis on positions $-.75, -.25, .25, .75$, that we sample on a 50×50 grid between $[-1; 1]$ over the y and z dimensions. We can see that the repartition of audio descriptors in the latent space is quite non-linear but smoothly evolving, such as the spectrum are coherently spread across the latent space. Interestingly, sound with high flatness seems to be concentrated around the origin of the representation, that correspond to the latent space with higher probability $p(\mathbf{z})$. Furthermore, this non-linear repartition of the audio descriptors confirms the idea of a non-linear relationship between timbre spaces and the corresponding descriptors. We can see that, contrary to the maps fig. 3.19, the descriptors are more parsimoniously spread, and seem to correspond to the features of the target perceptive space.

Descriptor-based synthesis. The smoothness of topology maps previously observed points out that the descriptors are behaving locally in a linear way. Writing a descriptor function as a function of $\mathbf{z}$ such that $\text{Desc}[\mathbf{z}] = \text{Desc}[\boldsymbol{\mu}_p(\mathbf{z})]$, where $\boldsymbol{\mu}_p$ is the mean function of the decoder, this observation comes back to assuming the local derivability of $\text{Desc}[\mathbf{z}]$. Given a starting point $\mathbf{z}_0$, we can obtain a rough estimation of this derivative by taking a neighbourhood $\mathbf{N}$ around $\mathbf{z}_0$, evaluating the descriptor $\text{Desc}[\mathbf{z}]$, and computing the local evolu-

tions Desc[$\mathbf{z}_0$] – Desc[$\mathbf{N}$]. We can thus derivate a path search algorithm to retrieve the optimal path that matches a target descriptor profile $t[n]$, from a point $\mathbf{z}_0$. While quite rudimentary, this method allows us to perform *descriptor-based synthesis*, and provides a very intuitive way of sampling new sounds from the latent space. This algorithm is presented in Alg. 1, and an example is shown fig. 3.31.

Algorithm 1: Descriptor-based path synthesis

Data: space $\mathbf{z}$, encoder $q_\phi(\mathbf{z}|\mathbf{x})$, decoder $p_\theta(\mathbf{x}|\mathbf{z})$
Data: origin spectrum $\mathbf{x}_0$, target series $\mathbf{t}_{1..N}$, descriptor d
Result: spectral distrib. $S \in \mathbb{R}^{N \times F}$

// Find origin position in latent space
$\mathbf{z}_0 = q_\phi(\mathbf{x}_0)$
// Evaluate origin descriptor
$\mathbf{d}_0 = evaluate(\mathbf{x}_0, d)$
for $i \in [1, N]$ **do**
    // Latent 3-d neighbourhood of current point
    $\mathbf{N}_i = neighborhood(\mathbf{z}_{i-1})$
    // Sample and evaluate descriptors
    $\mathbf{X}_i = q_\phi(\mathbf{N}_i)$
    $\mathbf{D}_i = evaluate(\mathbf{X}_i, d)$
    // Compute difference to target
    $\Delta_i = |(\mathbf{D}_i - \mathbf{d}_{i-1}) - (t[i] - t[i-1])|^2$
    // Find next latent point
    $\mathbf{z}_i = argmin(\Delta_i)$
    // Decode distribution
    $S[i] = p_\theta(\mathbf{z}_i)$
end

3.4.4 Conclusion

In this section, we then investigated the use of variational auto-encoders and variants (Wasserstein Auto-Encoders, Adversarial Auto-Encoders, Renyi Auto-Encoders) for magnitude spectrum learning. Despite efficient reconstruction abiilities, we saw that the unsupervised latent space extracted from spectral features were not organized to correspond main audio factors of data, while seeming to get a smooth (but not sufficient) perceptual organization. Unfortunately, main approaches proposed in the literature to increase the disentangling abilities of the latent representation, such as increasing the β parameter, did not seem to be efficient with audio factors of variation.
Therefore, we proposed two different regularization strategies to make variational spaces more compliant to audio features : a method based on external symbol information, that used the latent representation as a translation space between audio data and musical labels, and a method based on perceptual regularization using human similarity experiments. Both approaches significantly increased the representational properties of the latent space for audio analysis, and can also provide smoother and controllable latent topologies for creative uses. Furthermore, we saw that the proposed processes allowed numerous applications, as perceptual inference, constrained generation, descriptor-based syntheses, direct interaction, instrument morphing,

providing a complete analysis-synthesis framework.
However, there is still an important aspect missing in this system : *time*. Indeed, time is fundamental in sound and music, and so far we were not able to model the temporal shape of the addressed signals. However, this spectral approach provided interesting results, such that we would like to keep this process while modeling time at a higher level. This is also coherent with the temporal *multi-scale* nature of musical signals, that can be modelled as stacked levels of temporal hierarchies, from the spectral bin to the full signal. This approach would also allow us to directly perform *raw audio signals*, that is still a challenge for most machine-learning methods. We address time and sequentiality in the next chapter.

Time and prediction in generative models

4

"I am sure that i lost my keys, but when...?"
(30 september 2012)

In the previous section, we investigated the capacity of Auto-Encoding Variational Bayes to learn single spectral frames obtained from time-frequency representations, that can be seen as a local snapshot of the signal frequency content. This allowed us to model incoming signals on a short time range (about 100ms, taking 2048 samples with a sample rate of 22050Hz). However, these frames are learned individually, such that the dynamical structure of each individual is not modeled by the system. This limitation prevents not only to extract *dynamical* features from the representation, but also to represent the true underlying manifold involving the time dimension, that is yet mandatory if we target to model audio information.
Hence, we have to develop AEVB-inspired models that are able to model the dynamical aspect of the incoming audio files, such that the obtained representation also reflect the *temporal* evolution of a given signal. However, modeling time in AEVB models requires an additional step, as these systems are developed on data sets composed from singletons $\mathbf{x}$ rather than sequences $\{\mathbf{x}_1 ... \mathbf{x}_T\}$. In this work, we model this input sequence in the latent space by a trajectory $\{\mathbf{z}_1 ... \mathbf{z}_T\}$, hence using the latent representation as a *state space*. This state space, that is independent to time, is then used to model the sequence dynamics, corresponding to a higher-level features of much lower dimensionality. Recalling the manifold hypothesis underlying variational auto-encoders (see sec. 2.2.2), this means that a full sound $\mathbf{x}_{1:T}$ corresponds to a trajectory curve in the manifold underlying all the possible realizations of the modeled dataset.

However, we still need a way to encode the dynamics of the original sequence in the system. We address this shortage with two original contributions, targeting to model the sequence $\{x_t\}_{t \in 0...T}$ by only modeling the corresponding $\{z_t\}_{t \in 0...T}$. The first contribution is to model the distribution $p(\mathbf{z}_{t<\tau} | \mathbf{z}_{t \geq \tau})$, that can be seen as an extended version of the transition model $p(\mathbf{z}_t | \mathbf{z}_{t-1})$ used in state-space models. The modeling of this distribution, that we call the *prediction* task, allows to learn the latent trajectories of the corresponding time series, and then to perform the dynamical modeling of the data. As this prediction task is performed jointly with inference and generation, the chosen prediction method will shape the topology of the latent space, performing implicitly a *temporal regularization* of the representation. We then proposed three different prediction methods, based on separate mathematical backgrounds : a first method based on *contrastive predictive coding*, a representation framework used for slow-feature analysis, a second based on *normalizing flows*, modeling deformations of the latent space across time index t, and a third based on *Gaussian processes*, a powerful

continuous regression method.

However, while modeling the predictive distribution $p(\mathbf{z}_{t<\tau}|\mathbf{z}_{t\geq\tau})$ allows to model the dynamics of the input sequence directly in the latent space, the extraction of *dynamical* features from this trajectory is still impossible. Hence, we propose a new model, called *ShrubVAE*, based on the progressive encoding/decoding of increasing time scales. The interest of the proposed model is twofold : first, it provides a new method to use AEVB for time series modeling, enabling to generate entire sequences using a single latent vector, while preventing the undesired degeneracies that occur with auto-regressive decoders (see ref. 2.3.2.3). Secondly, ShrubVAE leverages multi-layered latent architectures to model latent dynamics of increasing temporal scopes, enabling analysis and generation of multiple time granularities from the lower (single spectral frames) to the highest (in this work, up to 150 frames) level of temporal modeling. Moreover, this consecutive downsampling tasks during inference allow to greatly extend the scope of time steps model by a single latent state, hence also significantly extending the scope of the prediction methods proposed previously.

4.1 Time series analysis and prediction

Time series analysis is a seminal field of statistics, that have been extensively studied for decades and gave rise to numerous diverse approaches. As we will not be able to give an exhaustive state of the art of such a large research field, we will focus on models that are related to our purpose. First, we will shortly summarize stochastic processes, that are widely used in the audio signal processing (especially auto-regressive processes, that are related to linear systems modeling). Then, we will review dynamical Bayesian systems, leveraging statistical Bayesian learning (see sec. 2.1.1)for sequence analysis, that provided popular models such as Hidden Markov Models and State-Space Models. Finally, we will review Gaussian Processes, a powerful Bayesian modeling method that can be used for regression or classification, that we use in one of our prediction method.

4.1.1 Stochastic processes and model identification

Formally, a time series is a sequence of dimensional vectors $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2 ... \mathbf{x}_N]$, here defined on $\mathbb{R}^D$, generally assumed to be taken over regular time steps. Time series analysis consists in extracting statistics or modeling the underlying process of one or multiple time series, generally assuming an *underlying structure* between time steps, that can be used for feature extraction or prediction. The complexity of this task gave rise to many different methods, targeting distinct applications and series types. By example, some approaches propose to model time series with a parametric function of t such that $\mathbf{x}(t) = f_\theta(t)$, coming back to *regression* techniques (see 4.1.3). Alternatively, a flourishing class

of models rather model the transition function $\mathbf{x}_t = f_\theta(\mathbf{x}_{t-1})$, obtained from initial conditions. Such approaches are much more powerful, as they can be understood as the discrete identification of a differential equation of arbitrary order (more details are given next paragraph).

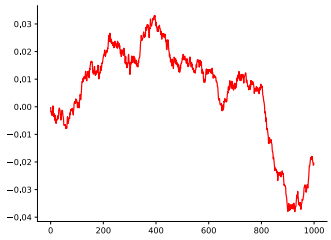


Figure 4.1: Example of a continuous Weiner process, that can be obtained from a cumulative sum of centered normal distributions.

Stochastic processes. Time series can also be defined using probabilities with the notion of *stochastic process*, formally defined as a collection of random variables $\{\mathbf{x}(t)\}_{t\in\mathbb{T}}$ defined on an index set $\mathbf{T}$, that can also be taken stochastic. In that case, a given signal is considered as a *realization* or *sample path* of the corresponding process. The notion of stochastic process is central in time series analysis, grounding the probabilistic approach to time series analysis. Particularly, the Wiener process has a special role in stochastic analysis, that is *normally distributed* noise continuous everywhere but differentiable nowhere (see fig. 4.1). A Wiener process, sampled with discrete index set $\mathbf{T} = \mathbb{N}$, is then called *gaussian noise*, and is often added to deterministic system to model them using probabilities. (see fig. 4.1)

An interesting property of Weiner processes is that their mean $\mu = \mathbb{E}[\mathbf{x}(t)] = 0$ (or first-order moment) is constant, and that their correlation function $\mathbb{E}[(\mathbf{x}_t - \mu_t)(\mathbf{x}_u - \mu_t)]$ (or second-order moment) only dependant from the distance $|u - t|$. Furthermore, a Weiner process verifies $p(\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \mathbf{x}_{t_n}) = p(\mathbf{x}_{t_1+\tau}, \mathbf{x}_{t_2+\tau}, \mathbf{x}_{t_n+\tau})$ for any $\tau \in$, verifying the *stationarity* of the process. *Stationary* is a seminal property of stochastic processes, meaning that its properties do not change over time, and therefore can be learned by a given model. As stationarity is a strong assertion, weaker definitions can be given, as the wide-sense stationarity that only requires the first- and second-order moments of the process to be constant. The concept of stationary is of central importance in time series analysis, as is it conditions the learnability of the process. However, it may be too restrictive to model complex signals, such that some stochastic models propose to bypass this assumption by proposing alternative formulations.

4.1.1.1 Moving-Average and Auto-Regressive processes.

The two notions of moving-average auto-regressive process are seminal in time series analysis, and furthermore find a natural application in digital audio processing for system identification. A *moving-average process* can be described

$$\mathbf{x}_t = \mu + c + \epsilon_t + \sum_{i=0}^{p} b_i \epsilon_{t-i} \tag{4.1}$$

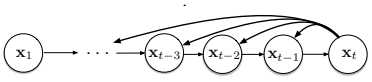


Figure 4.2: Graph of a probabilistic auto-regressive process

The model parameters are thus the p coefficients b_i, then defining a MA-process $MA(p)$ of order p. Analysing a time series with MA then implies to identify the parameters $\mathbf{b}$, that can understood as a local linear regression of the new step under a fixed temporal scope p. A realization of such process can be then generated by iteratively applying

the equation 4.1 with a given set of initial conditions $[\mathbf{x}_0 ... \mathbf{x}_p]$. However, MA-processes have a limited expressiveness, and the identifying of the coefficients is non-trivial, as we cannot access the stochastic components ϵ_i. Another family of processes, called *auto-regressive processes*, are defined

$$\mathbf{x}_t = c + \epsilon_t + \sum_{i=0}^{p} a_i \mathbf{x}_{t-i} \tag{4.2}$$

where the parameters are the coefficients a_i, defining a AR-process $AR(p)$ of order p. AR-processes are very expressive processes that are widely used in practical applications for time series modeling, the mean-squared error between the modelled and the target signals decreasing with order p. The coefficients $\mathbf{a}$ can be easily retrieved using Yule-Walker equations, or using the estimated auto-correlation function of the signal

$$R_{ff}(\tau) = \sum_{n \in \mathbb{Z}} x[n] x[n-l]$$

the describes the signal auto-correlation depending on lag τ. Furthermore, the auto-correlation function is an efficient method to detect periodicities in the signal, and then to estimate the optimatal order p necessary to accurately model the signal $x[n]$ without over-fitting (see fig. 4.3). Alternative schemes, such as *Akaike-Information Criterion* or *Minimum Description Length*, provide information-theory criteria that can be used to optimally retrieve the optimal order p of the model.

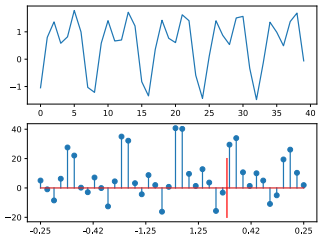


Figure 4.3: Auto-correlation allow to give estimates of the intern regularities of the signal (here, original frequencies in red)

The two processes 4.1 and 4.2 can be combined, giving the following *Auto-Regressive Moving-Average* (ARMA) process

$$\mathbf{x}_t = c + \epsilon_t + \sum_{i=0}^{p} a_i \mathbf{x}_{t-i} + \sum_{i=0}^{m} b_i \boldsymbol{\epsilon}_{t-i} \tag{4.3}$$

whose parameters $\mathbf{a}$ and $\mathbf{b}$ can be fully extracted with the Box-Jenkins method. ARMA processes still provide a strong baseline for time series analysis, and are still widely used in numerous domains. Further evolutions of this models can be found in literature, such that *Autoregressive Conditional Heteroskedasticity* models that updates the variance of stochastic components based on previous errors, or *auto-regressive modules with exogeneous inputs* that allows to model ARMA processes depending on external inputs.

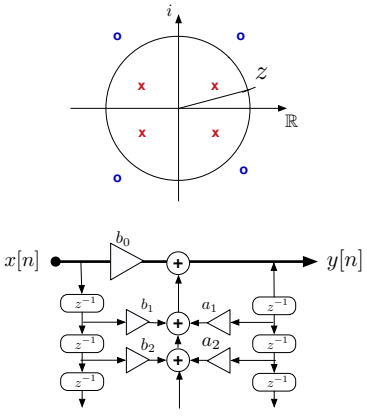


Figure 4.4: Digital filter equivalent to ARMA model 4.3, where z^{-1} is a unit delay. Poles and zeros can be extracted from coefficients $a_0 ... a_N$ and $b_1 ... b_M$, representing the frequency response on the unit circle.

An interesting property of ARMA models is their natural incidence in digital processing. Indeed, the equation 4.3 can be formulated as the *difference equation* of a infinite impulse response (IIR) digital filter, plus an additional source of noise. Hence, identifying the coefficients $\{\mathbf{a}, \mathbf{b}\}$ of an ARMA process can thought as finding the coefficients of

the digital filter (see fig. 4.4)

$$H(z) = \frac{b_0 + \sum_{i=1}^{p} b_i z^{-i}}{a_0 + \sum_{i=1}^{q} a_i z^{-i}} \tag{4.4}$$

$$= \frac{\prod_{i=0}^{q}(1 - q_i z^{-1})}{\prod_{i=0}^{p}(1 - p_i z^{-1})} \tag{4.5}$$

$$\tag{4.6}$$

where q_i and p_i are respectively the *zeros* and *poles* of the filter, the zeros being given by the MA coefficients and the poles by the AR coefficients. ARMA thus provides a stochastic framework for linear model identification, such as the signal was obtained by forwarding a white noise in the linear system identified by coefficients $\mathbf{a}, \mathbf{b}$. Model identification can be whether performed by averaging these coefficients on windows of data, assuming the *ergodicity* of the signal, or successively updated to provide a representation of the signal across time. The second method, called *Linear Predictive Coding* (LPC) [323, 324], is a well-known representation for audio source-filter separation, and can be used to speech processing [325] and harmonic-noise separation [326]. Furthermore, LPC can be understood as an estimation of the Discrete Fourier Transform of size T whose approximation is optimal when $p \to T$, under some assumed noise and up to an amplitude coefficient proportional to the variance of the noise. Auto-regressive approaches are thus fundamental in digital audio processing, and still provide the basis of more recent algorithms as WaveNet (see sec. 4.2.1).

4.1.2 State-space models.

Auto-regressive approaches are powerful methods for time series analysis, performing model identification at the data level. However, the structure of the underlying process strongly conditions the model identification, and does not allow to model the uncertainty of the extracted parameters. Alternatively, some methods rather model the observed data $\mathbf{x}_t$ as being generated from an unknown system with an hidden *state* $\mathbf{z}_t$, taking inspiration from continuous state-spaces representations in automation theory that model

$$\begin{cases} \mathrm{d}\mathbf{z}/\mathrm{d}t = \mathbf{F}(t)\mathbf{z}(t) + \mathbf{B}(t)\mathbf{u}(t) \\ \mathbf{x}(t) = \mathbf{H}(t)\mathbf{z}(t) \end{cases}$$

where $\mathbf{u}(t)$) is an external action modifying the system state at time t. In discrete time, the continuous derivative can be replaced with a first-order difference, such we aim to model the transition from $\mathbf{z}_t$ to $\mathbf{z}_{t-1}$. If some normal noise ϵ_t is added to both transition and observation models, and all functions as linear transforms, we obtain the *Kalman filter*

$$\begin{cases} \mathbf{z}_t = \mathbf{F}_k \mathbf{z}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \boldsymbol{\Delta}_t^z \boldsymbol{\sigma}_z^2 \boldsymbol{\epsilon}_t \text{transition model} \\ \mathbf{x}_t = \mathbf{H}_t \mathbf{z}_t + \boldsymbol{\Delta}_t^z \sigma_z^2 \boldsymbol{\epsilon}_t \text{observation model} \end{cases} \tag{4.7}$$

that is a classical probabilistic formulation of state-space models (SSM, see fig. 4.5). Provided the linearity of transition and observation functions, the system is fully tractable with Bayesian inference, assuming the following generative model

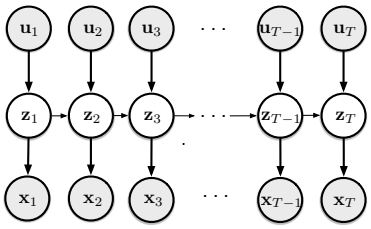


Figure 4.5: General graph of a state-space model (SSM), with observed variables $\mathbf{x}$, hidden variables $\mathbf{z}$ and actions $\mathbf{u}$

$$\begin{aligned} p(\mathbf{z}_0) &= \mathcal{N}(\mu_0, \sigma_0^2) \\ p(\mathbf{z}_t|\mathbf{z}_{t-1}) &= \mathcal{N}(\mu_{z,t}(\mathbf{z}_{t-1}, \mathbf{u}_t, \boldsymbol{\Delta}_t), \sigma_{z,t}(\mathbf{z}_{t-1}, \mathbf{u}_t, \boldsymbol{\Delta}_t^z)) \\ p(\mathbf{x}_t|\mathbf{z}_t) &= \mathcal{N}(\mu_{x,t}(\mathbf{z}_{t-1}), \boldsymbol{\Delta}_t^x) \end{aligned}$$

hence inferring the posterior $p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{x}_t)$. Kalman filters are then an example of *dynamic Bayesian learning*, that aims to perform Bayesian inference to model the transition function. The availability of such models to accurately describe the target time series is based on Markov hypothesis, that assumes

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \ldots, \mathbf{x}_0) = p(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{4.8}$$

such that the corresponding process is memoryless. Such processes, then called *Markov chains*, ensure the Bayesian tractability of dynamical models. Despite this strong assumption, Bayesian dynamical models can be expressive enough to model complex relationships on probability graphs of arbitrary shape. *Hidden Markov Models* (HMM) [327] is an example of dynamical Bayesian learning, modeling the dependencies between the series time steps with higher-level latent variables (see fig. 4.6. Unlike state-space models, the underlying Bayesian graph of HMMs is not necessarily causal, allowing to infer more expressive models. However, the tractability of HMMs requires discrete latent variables, then estimating the probability $p(\mathbf{x}_t|\mathbf{z}_1 = z_1, ...\mathbf{z}_N = z_N) = p(\mathbf{x}_t|\mathbf{z}_N = z_N)$ and a set of transition probabilities . HMMs are then usually trained using Expectation-Maximization or MCMC techniques on the log-likelihood of the data. Similarly to non-dynamical Bayesian systems, variational approaches to HMMs have been proposed [328, 329], allowing to infer continuous latent spaces. However, inference in Hidden Markov Models is quite computationally heavy, and scale to long time series with difficulty.

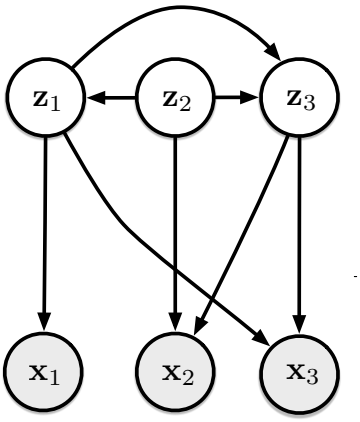


Figure 4.6: Example of a *Hidden Markov Model*, a generalized acyclic graph where temporality is driven in hidden discrete latent variables.

4.1.3 Gaussian processes.

Formulation. Finally, *Gaussian Processes* is a commonly used method for continuous time regression over finite observations. While classical regression models use a finite number of K basis functions $\{\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), ..., \phi_K(\mathbf{x})\}$, Gaussian processes can be proven to be equivalent to a regression with an infinite basis of functions, and is thus appealing because of its expressive power. Given a collection of random variables $\mathbf{X} = \{(t_i, \mathbf{x}_i)\}_{i=1}^N$,

where t is the continuous time index and $\mathbf{x}$ the corresponding output, Gaussian processes define a probability density over the function space $\mathbf{f}$ defined by $\mathbf{x} = f(t) + \sigma_n^2 + \epsilon_t$ [330]

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{m}(X), k(\mathbf{X}, \mathbf{X})) \tag{4.9}$$

where the mean vector can be taken as the empirical mean $\mathbf{m}(X) = \frac{1}{N}\sum_i x_i$ and $k(\cdot,\cdot)$ is the *kernel function* defining the process. Hence, provided new observations $\{(t_i^\star, \mathbf{x}_i^\star)\}_{i=1}^N$, we can tractably obtain the Bayesian *posterior distribution* over $\mathbf{f}$, that is

$$\begin{cases} \mu^\star = k(t^\star, t)(K + \sigma_n^2 \mathbb{I})^{-1}\mathbf{x} \\ \sigma^\star = k(t^\star, t^\star) - k(t^\star, t)^\top (K + \sigma_n^2 \mathbb{I})^{-1} k(t^\star, t) \end{cases} \tag{4.10}$$

such that we can update the prior $\mathcal{GP}(\mathbf{f})$ after each new arriving values (see fig. 4.7). The ability of Gaussian processes to perform continuous-time regression on N basis on function is based on the *kernel trick*, that models the $N \times N$ correlation in input space using a kernel function, rather than in the $K \times K$ feature space. Several kernels $k(\cdot,\cdot)$ are provided in the literature, that can be also parametric and then learned during training on some criteria. In this work we use a simple *radial basis kernel* $k(\mathbf{x}, \mathbf{x}') = \frac{|\mathbf{x}-\mathbf{x}'|^2}{\ell^2}$ where the parameter ℓ is the scale of the kernel.

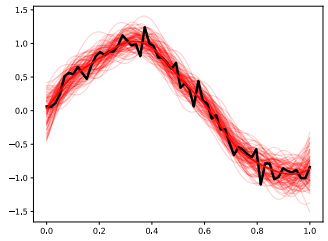


Figure 4.7: Example of Gaussian Process regression performed on a noisy curve. The input signal is the noisy black line, the bold red line is the MAP posterior of the GP, and all the other little little lines are functions drawn from the GP.

Sparse methods. While Gaussian processes is an efficient framework for continuous-time regression, it unfortunately suffers from scalability problems, as the complexity of the kernel matrix grows with the number N of examples. As the posterior computation requires the inversion of a $N \times N$ matrix, the complexity of this regression is $\mathcal{O}(n^3)$ and can thus become prohibitive for big datasets. A method to alleviate this heavy computation is then to perform the regression on a reduced number of points, allowing to enlighten the heavy inversion of $\mathbf{K_{xx}}$. Several methods have then been proposed to reduce this computational cost, such as *Projected Process Approximation* that approximates the covariance matrix with a matrix $\mathbf{Q_{xx}} = \text{diag}[\mathbf{K}_{xx} - \mathbf{K}_{xu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{ux}]$, where $\mathbf{u}$ are M inducing points [331]. Alternatively, *Sparse Posterior Gaussian Process* use a matrix $\mathbf{Q_{xx}} = \text{diag}[\mathbf{K}_{xx} - \mathbf{K}_{xu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{ux}] + \mathbf{K}_{xu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{ux}$, enforcing the Nystörm approximation to be exact on the diagonal. These methods allow to reduce the complexity up to $\mathcal{O}[nm^2]$, but do not ensure finding the optimal inducing points $\mathbf{u}$ that minimizes the approximation error.
Alternatively, Titsias propose to use variational methods to approximate a light Gaussian process $\mathbf{f}_q$ depending on independent auxiliary inputs $\mathbf{u}$, whose variational distribution is defined $q(\mathbf{f}_q, \mathbf{f}) = p(\mathbf{f}|\mathbf{f}_q)q(\mathbf{f}_m)$ [332]. The full model is then trained to reduce the distance between approximated and true posteriors, that is $D_{KL}[q(\mathbf{f}, \mathbf{f}_q)\|q(\mathbf{f}, \mathbf{f}_q, \mathbf{x})]$. This variational approximation performs significantly better with a lower number of points than previous sparse methods, and is then a common scheme for scaling GP to heavy datasets. Sparse GP methods can also advantageously be used in Bayesian Neural Networks [333],

allowing to model a GP prior over the network parameters, or used to perform probabilistic non-linear PCA [334].

GP for SSM modeling. As the linearity of the functions involved in usual state-space formulations prevents them to model expressive relations between the observations $\mathbf{x}$ and the hidden states $\mathbf{z}$, Gaussian processes can provide a solution to model richer relationships without dropping the tractability of the posterior. Wang & al. then propose a GP formulation for both observation and dynamical models, using a radial basis kernel for modeling $p(\mathbf{y}|\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, \mathbf{K_{zz}})$ and a transition model [335]

$$p(\mathbf{x}|\mathbf{f}) = p(\mathbf{x}_1) \int \prod_{t=2}^{N} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{f}) p(\mathbf{f}) d\mathbf{f}$$

using a *linear + RBF* kernel $k_z = k_{RBF}(\mathbf{z}, \mathbf{z}') + \alpha \mathbf{x}^T \mathbf{x}$ to add an auto-regressive component to the prediction. However, the method proposed by Wang & al. have the same drawbacks than standard GP when applied to big datasets, thus limiting their applications. A more flexible method is provided by Frigola & al., leveraging variational learning to infer the optimal GP inputs $\mathbf{u}$ [56]

$$p(\mathbf{x}, \mathbf{z}, \mathbf{f}, \mathbf{u}) = p(\mathbf{x}_0) \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{f}_t) p(\mathbf{f}_t|\mathbf{f}_{1:t-1}, \mathbf{x}_{0:t-1}, \mathbf{u})$$

$$q(\mathbf{z}, \mathbf{f}, \mathbf{u}) = q(\mathbf{u}) q(\mathbf{z}) \prod_{t=1}^{T} p(\mathbf{f}_t|\mathbf{f}_{1:t-1}, \mathbf{x}_{0:t-1}, \mathbf{u})$$

that is fully tractable and trainable with stochastic variational inference, and can be sampled using MCMC particles. However, in this model the approximated transition function $q(\mathbf{z}|\mathbf{z}_{-1})$ is not explicitly modelled, and the MCMC-based particle sampling procedure can be heavy to run. Subsequent similar models alleviate this constraint : Eleftheriadis & al. explicitly model $q(\mathbf{z}|\mathbf{z}_{-1})$ using LSTM networks (see sec. 4.2.1) allowing to model expressive transition functions and to train the model using black-box VI methods [336]. Doerr & al., rather suggest to use the real transition model $p(\mathbf{z}|\mathbf{z}_{-1})$, and also use stochastic gradient descent to train the overall model [337]. Mattos & al. also propose a hierarchical state-space framework using Gaussian processes, also resorting to variational GP sparse approximations [338].

All of these recent advances in GP processing provided powerful models for Bayesian time series analysis, and have been proven to model complex behaviours. However, their scalability to large and multi-dimensional datasets is still very demanding, and GPs are still designed to model only one sequence, being deprived of a conditioning latent model. However, its appealing properties are still encouraging researchers to work develop similar approaches, and GPs will ground one of the proposed prediction methods.

4.2 Neural methods for time series modeling

The approaches developed in the previous sections are based on the modeling of time series according to explicit structures, whether based on model identification or likelihood methods using tractable Bayesian posteriors. However, the development of connectionist approaches also allowed to address time series analysis with specific neural networks using gradient descent methods, then providing an alternative approach to this domain. These methods have been now extensively investigated in the literature, and motivated the development of numerous frameworks that are whether entirely based on neural processing, or based on hybrid models that use both the flexibility of these methods and the constrained architectures of previous modeling approaches.

4.2.1 Recurrent Neural Networks

Neural networks, that have been briefly introduced section 2.2.2, are powerful function approximators that can be trained with given loss criterion, using a stochastic gradient-descent optimization algorithm. However, standard neural networks are not able to naturally handle time series, and have to be given a *memory* to keep information about previous states. Such networks, called *Recurrent Neural Networks*, are embedded with an hidden memory $mathbf h_{t-1}$ updated at each time-step [339]. The classical formulation for RNNs is the one formulated by Elman [340]

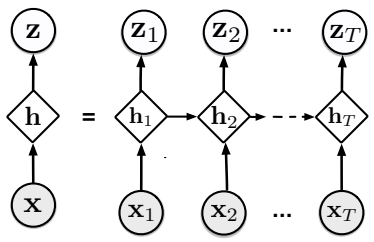


Figure 4.8: Example of an Recurrent Neural Network, that uses an hidden representation updated at each step with a specific operation.

$$h_t = \tanh(\mathbf{W}_{ih}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{z}_t) \tag{4.11}$$

where the hidden representation given at time t also depends on the last output by adding a transition matrix $\mathbf{W}_{hh}$ to the standard neuron (see fig. 4.8). RNNs can thus be thought as modeling a non-linear Markov Chain that learning its transition matrix. Despite the simplicity of this method, training RNNs over long time scales is very difficult, as the iterative update for each sequence step can whether make the gradient vanish, preventing the system to learn, whether to explode, providing degenerated functions. Indeed, performing an iterative map on high-capacity parameteric function such as RNNs can show a *chaotic* behaviour, and can then present hysteresis cycles or instability [341]. Several methods have then been developed to regularize RNNs during training such as ℓ_2 weight regularization, teacher forcing [342], or leaky integration [343]. The professor forcing learning schedule also help to increase the consistency of RNN units, using an adversarial method to match the generations obtained by feedback with those directly generated from the sequence [344]. However, while these methods can help to alleviate their instability, they do not address the problem of learning long-term dependencies. Subsequent models, such as hierarchical RNNs, can be used to model longer timescales, limiting the scope of individual units to prevent the degeneracy of gradients [345]. Alternatively, a decisive milestone was

the *Long Short-term Memory* networks by Hochreiter & Schmidhuber [346], allowing the system to *forget* information by gating before and after each subsequent step (see fig. 4.9)

$$\mathbf{i}_t = \sigma_g(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \quad \text{input gate} \quad (4.12)$$

$$\mathbf{o}_t = \sigma_g(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_i) \quad \text{output gate} \quad (4.13)$$

$$\mathbf{c}_t = \mathbf{i}_t \odot \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \quad \text{gated recurrent input} \quad (4.14)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \mathbf{c}_t \quad \text{gated recurrent output} \quad (4.15)$$

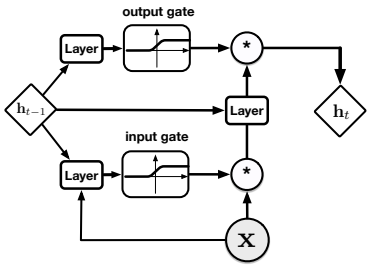


Figure 4.9: Graph of LSTM unit. Layer here is a sum of each input multiplied by a specific parameter matrix, and the gate is just a sigmoid unit allowing to mask given dimensions with Hadamard product $\odot$.

An additional *forget gate* can be added [347], such that $\mathbf{c}_t^f = \mathbf{f}_t \odot \mathbf{c}_{t-1}^f + \mathbf{c}t$, where $\mathbf{f}_t$ is an additional gating mechanism. These recurrent units are then able to forget data, allowing to free some memory in the retained representation to learn new features during iteration. The LSTM can also be added a reverse process, learning features $\mathbf{h_t}$ from $\mathbf{h}_{t+1}$, then also learning information from the backward sequence [348].

Gating mechanisms allow to accurately train recurrent neural units over large time spans, and are then commonly used for neural time series modeling. Their efficiency have then been extended to non-temporal purposes, providing efficient gradient selection methods as *attention mechanisms* [349, 350] and *transformers* [351, 352]. Alternative gating mechanisms have been proposed since such as *Gated Recurrent Units* (GRU) [353], that are used in this thesis. Some approaches also perform recurrent modeling in the spectral domain, that have been shown to accurately describe chaotic processes such as Lorentz attractors [354]. Finally, the recently proposed Neural Processes propose to combine neural networks and stochastic processes to infer probabilities directly in the function space, providing a very efficient framework for Bayesian learning of stochastic processes [355–357]. Time series modeling using neural networks is still an active field of research, that is likely to be further improved in the near future.

4.2.2 Dynamical Bayes and AR-models

Recurrent neural networks are then powerful units to model functions $\mathbf{y}_t = f_t(\mathbf{x}_t)$ with an independent recurrent mechanism. Hence, neural networks can be used in Dynamical Bayes to model observation or transition functions, empowering the expressiveness of these systems. In that case, neural networks can be used whether to model the transition functions of the latent variables, or directly generating new data samples, conditioned on a definite range of observed steps. Provided the *causality* of the system, meaning that the probability density of a step $\mathbf{x}_t$ is conditioned over the *previous* ones $\mathbf{x}_t$, these systems can be used to model the underlying auto-regressive process $\mathbf{x}_t = f(\mathbf{x}_{t-1}...\mathbf{x}_{t-T})$ of the series.

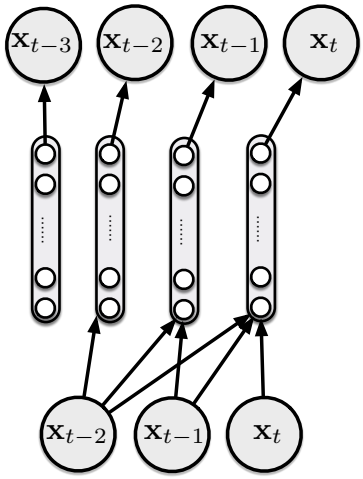


Figure 4.10: The Neural Autoregressive Distribution Estimator [358]

The *Neural Auto-regressive Distribution Estimator* [358, 359] network, proposed by Larochelle & al., aims to model the auto-regressive distribution of a binary data vector $\mathbf{x} \in [0;1]^N$ (see fig. 4.10)

$$p(x_i = 1|\mathbf{x}_{<i}) = \sigma(b_i + \mathbf{W}_i^\top \mathbf{h}_i)$$
$$\mathbf{h}_i = \sigma(\mathbf{c} + \mathbf{W}_{<i}\mathbf{x}_{<i})$$

that can be understood as an hybrid between recurrent neural networks, also using a memory $\mathbf{h}$, and an auto-encoder with hidden representation $\mathbf{h}_i$ (see fig. 4.10). As we summarized sec. 4.1.1.1, auto-regressive distributions are very expressive models, and are widely used outside the scope of temporal models. In that case, auto-regressive distributions are used to successively model the dimensions of a multi-dimensional vector $\mathbf{x_n}$, such that $\mathbf{x}_n = \prod_i^D p(x_d|\mathbf{x}_{<d})$. In that case, the ordering of dimensions $d_1, d_2 ... d_N$ does not have to be causal, and an arbitrary ordering can be chosen. Based on the NADE model, the *Masked Auto-Regressive Distribution Estimator* (MADE) rather uses the full matrix $\mathbf{W}$ rather than $\mathbf{W}_{<t}$ during generation, but uses mask to perform automatic dimension selection for the auto-regressive models.

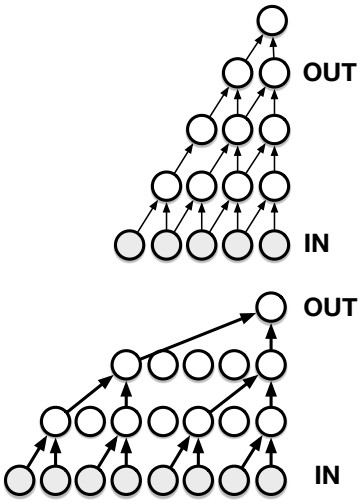


Figure 4.11: The WaveNet auto-regression module, that transform auto-regressive neural modules (top) using dilated convolutions (down) to extend the temporal scope using less connections.

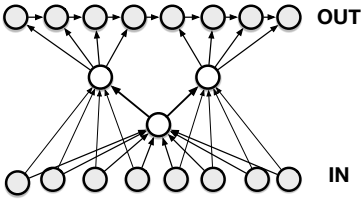


Figure 4.12: SampleRNN hierarchical modeling used to learn the underlying auto-regressive distribution $p(\mathbf{x})$

However, auto-regressive models such as NADE and MADE can be slow to converge, and assume weight matrices whose size matches the sequence length, and can then become prohibitive with long-term sequences. Another method for auto-regressive modeling is the WaveNet model, proposed by van den Oord & al., that provides so far the baseline for raw waveform synthesis. WaveNet is based on stacked 1d *dilated convolution networks* with exponentially growing dilation factors, that significantly increase the receptive field of the context $\mathbf{x}_{t_i} ... \mathbf{x}_{-1} t$ used to generate the next step $\mathbf{x}_t$ (see fig. 4.11). The operation processed by a dilated convolution unit is

$$\mathbf{h}_c = \mathbf{b}_c + \sum_{k=0}^{c_{in}-1} \mathbf{w}_{c_{out},k} \star_d \mathbf{x}_k \tag{4.16}$$

$$\mathbf{w}_n \star_d \mathbf{x}_n = \sum_n \mathbf{w}_n \mathbf{x}_{d \cdot n+m} \tag{4.17}$$

that can be described as a *spaced* convolution operation, such that a kernel of m samples will be applied over a time range of $m \times d$ samples of the sequence. The use of dilated convolutions allowed WaveNet to extend its receptive field up to 300 milliseconds, that is sufficient to ensure the perceptual consistency of the generated signal. A similar method is the SampleRNN model proposed by Mehri & al., that uses structured hierarchical RNNs to predict windows of samples, thus fastening the generation process [360] (see fig. 4.12). Both methods are compatible with *teacher forcing* and *professor forcing*, that can help the robustness of prediction.

While WaveNet and SampleRNN are the first machine-learning models able to generate convincing audio waveforms, an impressive success regarding the complexity of the challenge, these models rely on heavy conditional architectures that makes their learning tedious. Indeed, these models requires weeks of training on single GPUs, and the generation is still far from being real-time. Haque & al. recently proposed reduce WaveNet architectures using attention LSTMs, fastening a little computation during generation time [361].
In addition to this performance issues, another weakness of such auto-regressive networks is the lack of latent representation inferred from the data, then deprived of higher-level space that could use the extracted features for audio analysis. While systems such as WaveNet can be conditioned during training to allowing class-dependent generation, the number of possible classes is limited, and the scale of the datasets needed to train such algorithms can make their annotation discouraging. A step towards controlling WaveNet decoders have been proposed by Engel & al., based on the extraction of latent features extracted from variational auto-encoders to condition a WaveNet decoder [362]. However, these features are obtained from spectral information, and the do not extract a representation based on the raw information itself. Hence, real-time raw signal synthesis is still a challenging task, and developing models that are able long-term dependencies with an acceptable computational cost is an on-going active research problem.

4.2.3 Variational auto-encoding for time-series

AEVB methods for inference/generation of time series were investigated shortly after the publication of the original article. However, addressing time series analysis with AEVB-inspired methods can be divided in two different approach (see fig. 4.13). The first consists in modeling the full sequence $\{\mathbf{x}_1 ... \mathbf{x}_T\}$ with a single latent vector $\mathbf{z}$, that we call *dynamical compression*. A first approach using VAEs with auto-regressive decoders was investigated by Bowman for language processing, targeting to generate full phrases from a single latent vector[183]. However, this approach unveiled some undesirable properties of AEVB, such as *latent over-pruning*, that showed the inherent trade-off between the capacity of the decoding module and the information represented in the latent space (see sec. 2.3.2.3). Hence, Chen & al. then proposed to rather encode *local* features of the input, that were decoded and then assembled by an additional auto-regressive module. The second approach is rather based on representing the input $\{\mathbf{x}_1 ... \mathbf{x}_T\}$ with a latent sequence $\mathbf{z} = \mathbf{z}_1, ..., \mathbf{z}_T$, modeling one different latent position for each step. Hence, this approach rather considers the variational auto-encoder as a state-space model, using the generative distribution $p(\mathbf{x}|\mathbf{z})$ as the observation model (see sec. 4.5). Hence, the different solutions proposed in the literature for sequential AEVB will design different structures for the transition model $p(\mathbf{z}_t|\mathbf{z}_{t-1})$, whether by explicitly modeling this distribution in the variational process, or by modeling recurrence directly in the encoding and decoding modules

using RNNs.

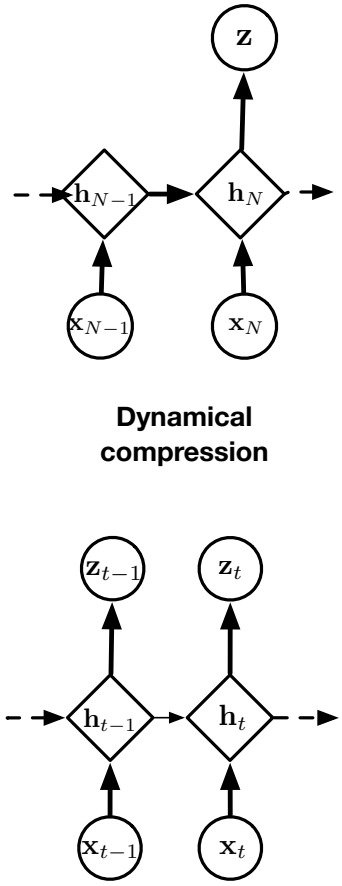


Figure 4.13: The process that we call *dynamical compression* rather extracts a single latent vector for a given sequence, using a RNN-like feature extractor, while the state-space approach used previously models one latent vector by incoming observation.

Latent auto-regressive modeling. Considering AEVB as an observation model $p(\mathbf{x}_t|\mathbf{z}_t)$ with a corresponding approximated posterior $q(\mathbf{z}_t|\mathbf{x}_t)$, we thus only miss to model the corresponding transition operator $p(\mathbf{z}|\mathbf{z}_{t-1})$ to turn the system into a full state-space system. In a recent paper Razavi & al. proposed to model a prior $p(\mathbf{z}_t)|p(\mathbf{z}_{t-1}) = \mathcal{N}(\alpha\mathbf{z}_{t-1}, \sigma_\epsilon)$, such that the correlation between successive latent steps can be controlled with the parameter α. Similarly, Rochemore & al. propose to model the latent dynamics using Dynamical Movement Primitives, that is a point attractor system described by second-order dynamics $\tau\ddot{\mathbf{z}} = \alpha(\beta(\mathbf{g} - \mathbf{y}) - \dot{z}) + \mathbf{f}$, $\mathbf{g}$ being the position goal of the movement and $\mathbf{f}$ enforcing the trajectory dynamics. The work proposed by Archer & al. is the first one to explicitly bridge AEVB and state-space models, modeling the posterior distribution $q(\mathbf{z}|\mathbf{x}) = q(\mathbf{z}_1, ..., \mathbf{z}_T|\mathbf{x}_1)$ as a product of normal distributions, such that $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{C})\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\Sigma}(\mathbf{x}))$, hence separating the covariance matrix obtained from amortization and the desired temporal deviation of the sequence [363]. A variational derivation of Kalman processes 4.7 was also proposed by Krishnan & al, relaxing the linear assumptions of transition and observation matrices using variational inference. This model enabled to use variational learning for *counter-factual* inference, allowing to predict the reaction of the system's behavior over a different set of external actions [364].

RNN-based approaches. Another way to introduce recurrence in AEVB models is to define encoding and decoding functions as RNNs modules, generating a vector of hidden variables $[\mathbf{h}_1...\mathbf{h}_T]$ for each step $\mathbf{x}_1...\mathbf{x}_T$. If we aim to model the full sequence using a single latent vector $\mathbf{z}$, we can then use the last hidden state $\mathbf{h}_T$ as a *context vector*, that is fed to an additional layer to obtain the parameters of the distribution $p(\mathbf{z}|\mathbf{x}_1...\mathbf{x}_T)$. In that case, we can generate a full sequence $\mathbf{x}_1...\mathbf{x}_t$ from a single vector $\mathbf{z}$ by giving a replicated sequence $[\mathbf{z}...\mathbf{z}]$ as input for the decoding RNN. This method differs from auto-regressive decoders by using $\mathbf{z}$ as an input for each step of decoding, preventing over-pruning of the latent space. This model, called *recurrent variational auto-encoder*, does not change the expression of ELBO 2.18 and then can be trained as a regular auto-encoder [365].

Alternatively, we can also choose a state-space approach to variational auto-encoding, using the vector $[\mathbf{h}_1...\mathbf{h}_T]$ given by the encoding RNN to compute a sequence $\mathbf{z} = [\mathbf{z}_1, ..., \mathbf{z}_T]$. Hence, the difference with standard AEVB is that the encoding module is using both the past state $\mathbf{h}_{t-1}$ in addition to $\mathbf{x}_t$, then inferring the latent state $\mathbf{z}_t$ using the module memory. Assuming an auto-regressive distribution for the model evidence $p(\mathbf{x}) = \prod_{i=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{<t})$, we then derive a *sequential lower-bound* such that [366, 367]

$$\log p(\mathbf{x}_{0...T}) = \mathbb{E}_{\mathbf{z}_1...\mathbf{z}_T \sim q(\mathbf{z}|\mathbf{x})}\Big[\sum_{i=0}^{T} \log p(\mathbf{x}_t|\mathbf{z}_t)\Big] - D_{KL}[q(\mathbf{z})\|p(\mathbf{z})] \quad (4.18)$$

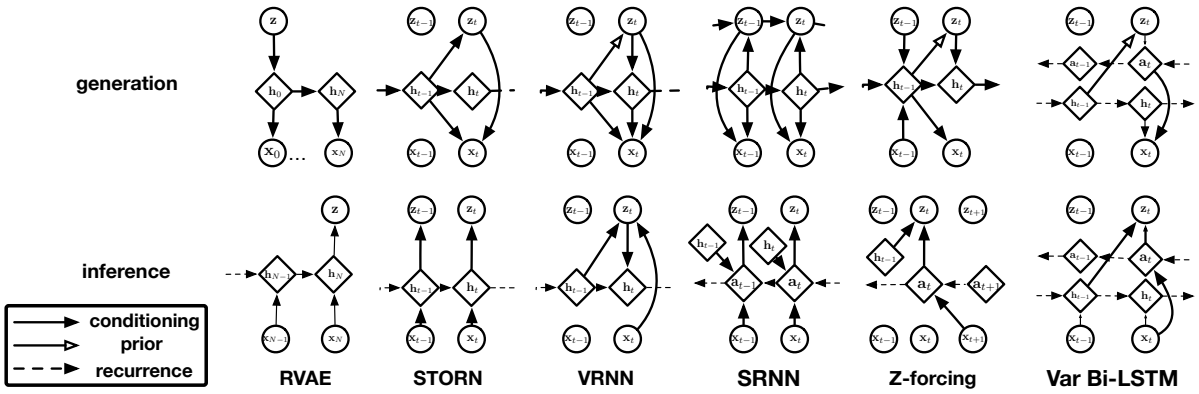


Figure 4.14: Different propositions of variational recucrrent networks : RVAE [365], STORN [372], VRNN [373], SRNN [369], Z-forcing [370]

using the Markov hypothesis of the decoding function $\log \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{<t}) = \log \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{h}_t)$. Bayer & al. propose to define the prior $p(\mathbf{z})$ as Gaussian process $\mathcal{GP}(0, k(\min r, t))$, regularizing the latent sequence $[\mathbf{z}_0, \mathbf{z}_1 - \mathbf{z}_0, ..., \mathbf{z}_T - \mathbf{z}_{T-1}]$ on $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ [366]. However, in this case the transition model $p(\mathbf{z}_t|\mathbf{z}_{t-1})$ does not explicitly appear in the variational process. Chung & al. then proposed *Variational Recurrent Neural Network* (VRNN), defining inference and generation processes as

$$
\begin{aligned}
p(\mathbf{z}_t) &= \mathcal{N}\big(\boldsymbol{\mu}_p(\mathbf{h}_{t-1}), \boldsymbol{\sigma}_p^2(\mathbf{h}_{t-1})\big) \\
p(\mathbf{x}_t|\mathbf{z}_t) &= \mathcal{N}\big(\boldsymbol{\mu}_p(\mathbf{h}_{t-1}, \mathbf{z}_t), \boldsymbol{\sigma}_p^2(\mathbf{h}_{t-1}, \mathbf{z}_t)\big) \\
q(\mathbf{z}_t|\mathbf{x}_t) &= \mathcal{N}\big(\boldsymbol{\mu}_q(\mathbf{h}_{t-1}, \mathbf{x}_t), \boldsymbol{\sigma}_p^2(\mathbf{h}_{t-1}, \mathbf{x}_t)\big)
\end{aligned}
$$

where the transition model is modeled in the prior, using an additional MLP module. A similar model was recently proposed by Gregor & al., performing random time jumps between states to reinforce the temporal robustness of the learned transition function [368].
Another way to reinforce the temporal features learned by sequential AEVB is to train the encoder using the future steps of the sequence, splitting the information used by the encoding and decoding modules to prevent latent over-pruning. Fraccaro & al. propose to directly mode the transition $q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{h}_{t>T}, \mathbf{x})_t$ performing variational approximation of the transition model using an anti-causal RNN for inference [369]. This system can then be understood as a stochastic bi-directional RNN, with an additional latent space that joins the information between backward and inference processes. This proximity to LSTMs brought some authors to directly use these models as LSTM units in more complex architectures, where additional variational models can infer anti-causal features from causal features [370] and vice-versa [371]. A summarizing table showing probabilistic graphs for inference and generation models for all these systems can be seen fig. 4.14.

Disentangling temporal and static features. The methods introduced in previous paragraphs are then split in two different approaches, modeling a full sequence $\mathbf{x}_{0:T}$ whether by a single latent state $\mathbf{z}$, or by a sequence of states $\mathbf{z}_{0:T}$. However, performing both could provide a way to disentangle dynamical from static features, splitting the latent variables in two kinds : *dynamical* latent variables $\mathbf{z}^t = [\mathbf{z}_1^t..\mathbf{z}_N^t]$, that encode dynamics of the input series, and *static* features $\mathbf{z}^d$ that are invariant across the whole sequence (see fig. 4.15). Hsu & al. recently proposed to disentangle these latent variables using a structured prior $p(\mathbf{z}_t^t, \mathbf{z}_t^d, \boldsymbol{\mu}) = p(\boldsymbol{\mu})p(\mathbf{z}_t^d|\boldsymbol{\mu})p(\mathbf{z}_t^d)$ where the variables $\mathbf{z}_t^d$ is conditioned on a feature vector $\boldsymbol{\mu}$ that is extracted from the whole sequence. A version of this model was consequently proposed by Li & al., where an explicit transition model between $\mathbf{z}_t$ and $\mathbf{z}_{t-1}$ in both variational and generation models [374]. In a similar manner, Grathwohl & al. disentangled the two latent variables $\mathbf{z}^d$ and $\mathbf{z}^t$ between space and time by using different generation priors, the first modeling independent variables across time and the second with an explicit transition model [375]. These experiments show that AEVB is able to shape its variational distribution on sequential data depending on its architecture, motivating our temporal hierarchical approach developed in section 4.4.

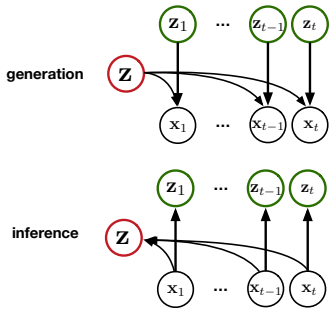


Figure 4.15: Separating the latent space between static (red) and dynamical (green) dimensions can enforce the disentangling between global and local features of the sequence.

4.3 Variational methods for latent space prediction

In the last section we described two different approaches for time series analysis, that are somehow reflecting to the two first sections of chapter 2. The first rather model a given time series $\{\mathbf{x}_t\}_{t\in[0...T]}$ as a sample of a stochastic process, using model identification or dynamical Bayesian inference to infer higher-order parameters of the underlying model, while the second rather rely on high-capacity series models such as RNNs to whether extract high-level features using a suitable loss function, or to model an auto-regressive process that learns by predicting the next inputs over a short time range. While the first are still commonly used in the domain of DSP, they do not allow to model long-term temporal scales, and the tractability assumption of Bayesian models often prevent to model complex time series, and enforce the definition of discrete latent spaces. Reversely, the second approach provided powerful modeling tools for time series, the recent models being able to generate convincing raw audio waveform with a extended temporal field compared to former auto-regressive techniques. However, the prohibitive computational cost of these models, as well as their lack of interpretability and control make these models very difficult to involve in creative processes.

We saw that AEVB could be beneficially used to mix both approaches, whether by performing dynamical compression (identifying a full sequence with a single vector $\mathbf{z}$) or by defining a state-space model (identifying the series with a latent sequence $\mathbf{z}_1, \mathbf{z}_T$). Sequential approaches

with AEVB, contrary to most neural methods to time series modeling, allow to perform both analysis and generation sequences, thus gaining in terms of control and interpretability. Furthermore, as they allow expressive relations between the data and latent spaces, they also provide an alternative to most Bayesian dynamical approaches, whose tractability requirement often reduced the scope of possible models. We summarized sec. 4.2.3 the models proposed so far to perform time series modeling with the AEVB framework, that whether rely on the explicit definition of a transition model, or resorting to RNNs as encoding and decoding functions. A full definition of recurrent AEVB objectives can be then described

- *analysis*, extracting useful local features from a time series for a supervised objective
- *generation*, using the extracted features to generate the corresponding data
- *temporal modeling*, retrieving the latent dynamics

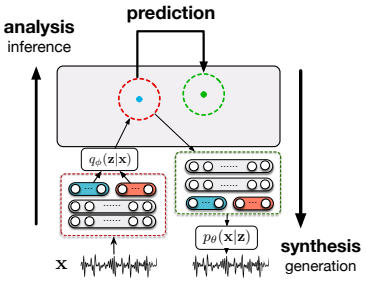


Figure 4.16: We add a *prediction* task in addition to analysis and synthesis process, modeling the dynamics of the data.

then adding an additional task to standard AEVB methods. In this section, we propose to add a fourth objective to recurrent AEVB, that is the *prediction* of the future latent states of the series. Indeed, most of the described models are not able to *predict* future values of the time series, except the ones from Chung & al. and Bayer & al. [372, 373]. However, even these two systems are modeling recurrence directly in the encoding and decoding modules, preventing the dynamical information to be entirely represented in the latent space. In this work we rather propose to model the series dynamics entirely in the latent space, relying on the idea that performing prediction only using then latent space will enforce it to get a temporal consistency. Moreover, performing dynamics modeling in the latent space is much cheaper computationally than in the data or hidden layers because of its much lower dimensionality, allowing to predict much longer temporal scopes than previous models, only predicting one step at a time (except for SampleRNN). Hence, in this section we do not model one-step transition models $p(\mathbf{z}_t|\mathbf{z}_{t-1})$ but extended predictive distributions $p(\mathbf{z}_{>t}|\mathbf{z}_{\leq t})$, that are jointly trained with the VAE. As the joint training of the model and of the predictor will apply an implicit regularization of the latent space, its topology will be strongly influenced by the chosen prediction method.

We then propose three different prediction methods, each of them relying on diverse theoretical backgrounds. The first proposed method is based on *Contrastive Predicting Coding*, a prediction technique that is based on mutual information maximization between extracted *contexts* and the predicted latent features. The second method uses *Normalizing Flows* (see sec. 2.3.2.1), that uses a sequence of invertible local transformations to model the temporal evolution of the predicted latent sequences. Finally, we propose a *Gaussian Process* prediction method, that performs a continuous regression on the encoded sequences to predict the next latent states.

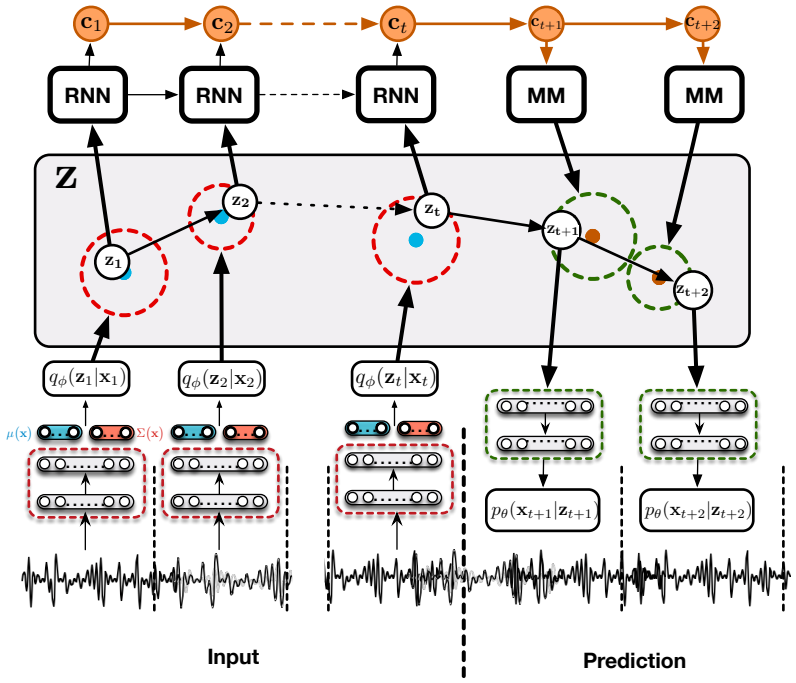


Figure 4.17: In the CPC prediction method, we extract higher-level variables, named *contexts*, that are enforced to model *slow features* that are used to predict the next step using linear predictors.

4.3.1 Contrastive Predictive Coding

Contrastive Predicting Coding (CPC) is a framework proposed by van den Oord to extract *context variables* from a given set of latent features, based on maximizing the mutual information between the extracted contexts and the feature vector [376]. CPC is based on Noise-Contrastive Information (NCI), a framework proposed by Gutmann & Hyvärinen used for density estimation, allowing to train a parametric model $p(\mathbf{x};\boldsymbol{\theta})$ on log-likelihood using a contrastive data distribution $\mathbf{n}$. Density estimation formally aims to find the optimal parameters $\boldsymbol{\theta}$ of a generative distribution $p(\mathbf{x};\boldsymbol{\theta})$ that maximizes the overall likelihood of the data. As the generative $p(\mathbf{x};\boldsymbol{\theta})$ is a probability density function it is constrained by the equality $\int p(\mathbf{x};\boldsymbol{\theta})d\mathbf{x} = 1$, that is often reformulated

$$p(\mathbf{x}) = \frac{\hat{p}(\mathbf{x};\boldsymbol{\theta})}{Z(\boldsymbol{\theta})}$$

where $Z(\boldsymbol{\theta}) = intp(\mathbf{x};\boldsymbol{\theta})d\mathbf{x}$ is the *partition function* of $\hat{p}(\mathbf{x};\boldsymbol{\theta})$, that is the *unnormalized density function*, or *free energy*, of $p(\mathbf{x};\boldsymbol{\theta})$. This decomposition, that is often used in undirected probabilistic graphs such as Boltzmann Machines [377, 378] (see sec. 2.1.2), allows to model probabilities with arbitrary score functions without caring about the normalization. However, modeling this partition function is often problematic, requiring estimation methods that are often tedious. NCI rather propose to train the parameters of $p(\mathbf{x};\boldsymbol{\theta})$ by using the density ratio trick (see sec. 2.3.2.3) transforming a density estimation to a classification task

$$\mathscr{L}_{NCE} = \mathbf{E}_{\mathbf{x}} \log r(\mathbf{x};\boldsymbol{\theta}) + \log 1 - r(\mathbf{n};\boldsymbol{\theta}) \tag{4.19}$$

where $r(\mathbf{x};\boldsymbol{\theta} = \sigma(\frac{q(\mathbf{x})}{p(\mathbf{x},\boldsymbol{\theta})})$ is the density ratio between $p(\mathbf{x},\boldsymbol{\theta})$ and a contrastive distribution $p(\mathbf{x}$. This framework then allows to optimize

$\hat{p}(\mathbf{x};\boldsymbol{\theta})$ without $Z(\boldsymbol{\theta})$, optimizing the parameters $\boldsymbol{\theta}$ to maximize the density $p(\mathbf{x},\boldsymbol{\theta})$ on data $\mathbf{x}$ while minimizing it when drawn from the contrastive distribution $q(\mathbf{x})$.

The prediction method proposed by CPC is to extract context variables $\mathbf{c}_t$ from a sequence of latent variables $[\mathbf{z}_0...\mathbf{z}_t]$ that is then used to predict latent variables $\mathbf{z}_{t+1}...\mathbf{z}_{t+K}$ with K simple linear predictors (see fig 4.17). The context variables and the predictors are based on InfoNCE, that derives the NCE criterion to maximize the mutual information between latent vectors $[\mathbf{z}_t...\mathbf{z}_{t+k}]$ and contexts $\mathbf{c}_t$

$$\mathbb{L}_{InfoNCE} = -\mathbf{E}_{\mathbf{x}}\frac{f_k(\mathbf{z}_{t+k},\mathbf{c}_t)}{\sum_{\mathbf{z}_j} f_k(\mathbf{z}_j,\mathbf{c}_t)} \tag{4.20}$$

where f_k models the density ratio between variables $\mathbf{z}_{t+k}$ and $\mathbf{c}_t$, and $\mathbf{z}_j$ are randomly taken contrastive sequences. InfoNCE intrisically optimizes the mutual information between predictions $\mathbf{z}_{t+k}$ and the context $\mathbf{c}_t$, as it lower-bounds $\mathbb{I}(\mathbf{z}_{t+k};\mathbf{c}_t) \geq \log N - \mathscr{L}_{InfoNCE}$. Maximizing this mutual information then enforces the context variables $\mathbf{c}_t$ to represent slow features of the evolution of $\mathbf{z}$, such that $\mathbf{c}_t$ represents robust local states that are not sensitive about the latent dynamics. Contexts $\mathbf{c}_t$ are obtained using an auto-regressive model $\mathbf{c}_t = g(\mathbf{z}_{\leq t})$, generally modeled with GRUs (see sec. 4.2.1).

While CPC is a light method to extract slow-feature context vectors from latent variables, the original framework can only be used to extract the context variables $\mathbf{c}_t$ from the hidden vectors $\mathbf{h}$, that can be used to obtain non-invertible representations of opaque models (such as WaveNet). Here, we involve the latent states predicted by the CPC module in the generation process, and evaluate how the predicted states can be used to generate the future steps $\mathbf{x}_{>t}$. This prediction method can thus be understood as modeling an implicit distribution $q(\mathbf{z}_{t>\tau}|\mathbf{z}_{<\tau})$) (see sec. 2.1.3.2). The overall model can then be described with

$$q(\mathbf{z}_{t\leq\tau}|\mathbf{x}_{t\leq\tau}) = \prod_{i=0}^{\tau} q(\mathbf{z}_t|\mathbf{x}_t) \tag{4.21}$$

$$\mathbf{c}_t = g(\mathbf{z}_{<t}) \;\; ; \;\; \mathbf{z}_{t+k} = \mathbf{W}_k\mathbf{c}_t \tag{4.22}$$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{i=0}^{\tau} p(\mathbf{x}_t|\mathbf{z}_t) \tag{4.23}$$

where the full latent sequence is obtained by concatenating the encoded and predicted vectors $\mathbf{z} = [\mathbf{z}_{t\leq\tau};\mathbf{z}_{t+1}...\mathbf{z}_T]$. We then derive the following lower-bound to train the overall model

$$\mathscr{L}_{CPC} = \mathbb{E}_{\mathbf{z}_0...\mathbf{z}_\tau\sim q(\mathbf{z}_{t\leq\tau}|\mathbf{x}_{t\leq\tau})}[p(\mathbf{x}|\mathbf{z})] + D_{KL}\big[q(\mathbf{z}_{t\leq\tau}|\mathbf{x}_{t\leq\tau})\|p(\mathbf{z})\big] \tag{4.24}$$

$$+ \mathscr{L}_{InfoNCE}(\mathbf{c}_t,\mathbf{z}_{t...t+k}) + \mathscr{R}[\mathbf{z}_{t>\tau},p(\mathbf{z})] \tag{4.25}$$

where $\mathscr{R}[\mathbf{z}_{t>\tau}]$ is an optional regularization term enforcing the implicit distribution of $\mathbf{z}_{t>\tau}$ to match the prior $p(\mathbf{z})$, such as MMD or an

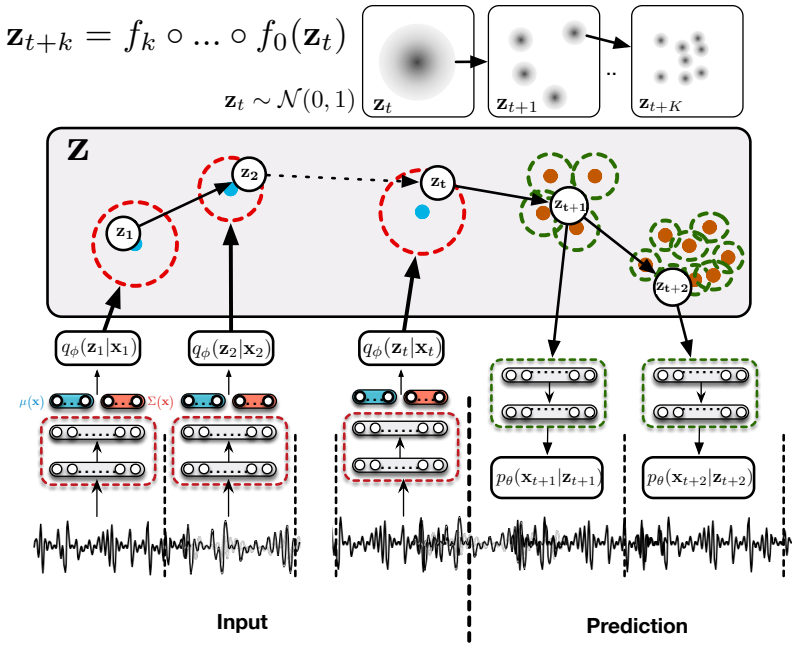


Figure 4.18: With this prediction methods we use normalizing flows to model *filtrations* of the latent space, modeling complex multi-modal distributions to encode the evolution through time.

adversarial criterion.

4.3.2 Filtration learning with temporal normalizing flows

In this section we rather rely on another framework, *normalizing flows* (see sec. 2.3.2.1), to derive a flexible yet powerful method for latent state prediction. This approach is inspired by the concept of *filtrations* of the stochastic process literature, that can be intuited by the idea that the more a stochastic process is observed, the more we can accurately describe its underlying properties. More formally, we consider stochastic process $\{\mathbf{X}(t)\}_{t\in\mathcal{T}}$ ($\mathcal{T}$ having a total order $\leq$) where every $\mathbf{X}(t)$ is a random variable from a probability space $(\Omega, \mathcal{A}, \mathcal{P})$ with sample space Ω, σ-algebra $\mathcal{A}$ and probability measure $\mathcal{P}$. A stochastic is said *filtrated* if the σ-algebra $\mathcal{F}_t$ induced by the successive $\mathbf{X}(t)$ are ordered non-increasingly, i.e. $\mathcal{F}_k \subseteq \mathcal{F}_l \subseteq \mathcal{A}$ for $k \leq l$. The notion of filtration is central in modern probabilities, expressing that the more information we have on a process, the finer we can identify a partition of its sample space Ω.

With this prediction method we aim to represent this concept of *filtration* in the latent space with *normalizing flows*, a sequence of invertible transformations $\mathbf{x}_K = f_K \circ f_{K-1} \circ ... \circ f_1(\mathbf{z})$ that can be used to model tractable expressive posteriors $\mathbf{q}_K(\mathbf{z}_k|\mathbf{x})$ from simple distributions $\mathbf{q}_0(\mathbf{z}|\mathbf{x})$. Here, we rather propose using normalizing flows to model the local temporal deviation of a starting latent point $\mathbf{z}_\tau$ that evolves through time. Therefore, we use all the successive $\mathbf{z}_{\tau+k} = f_{k-1}(\mathbf{z}_k)$ to predict the missing time steps, that we then give to the generative model to recover the data $\mathbf{x}_{t>\tau}$.
This method is related to the idea of filtration as the entrance point $\mathbf{z}_\tau$,

which has an uninformative prior $p(\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{1}))$ will be progressively refined by the flow across time steps, shaping the uniform Gaussian to a complex distribution modelled by the latent dynamics. The locality of the deformations f_k also regularize the temporal consistency of the latent space, such that wide moves in the latent space will consequently model important deformations. Furthermore, normalizing flows are able to transform uni-modal variational distributions $q(\mathbf{z}_\tau|\mathbf{x}_\tau)$ to multi-modal ones, such that flow is able to model the uncertainty of latent dynamics over multiple possible solutions. Normalizing flows can then be a powerful framework for latent prediction, that is able to model complex chains $\mathbf{z}_\tau \rightarrow \mathbf{z}_{\tau+1} \rightarrow ... \rightarrow \mathbf{z}_{\tau+K}$, while preserving the local consistency of the latent space. In this prediction method, we use a simple planar flow for each time step, that is expressed [172]

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^\top \mathbf{z} + b)$$

that has a tractable log-jacobian, and is very cheap to compute. However, applying the flow to a single $\mathbf{z}_\tau$ to obtain the following latent vectors $\mathbf{z}_{t\in[\tau;T]}$ would only give a one-step back prediction of the input sequence, and is thus unlikely to accurately model the information given by the entire sequence. We thus propose to amortize the flow by a feature extractor such as RNNs, such that flow is conditioned by a memory vector that embeds the past sequence

$$\begin{aligned}\mathbf{z}_k &= f_k(\mathbf{h}_\tau) \odot f_{k-1}(\mathbf{h}_\tau) \odot ... \odot f_1(\mathbf{h}_\tau)[\mathbf{z}_\tau] \\ \mathbf{h_t} &= \text{RNN}[\mathbf{z}_t, \mathbf{z}_{t-1}]\end{aligned}$$

Each flow f_k is amortized, such that its parameters are obtained from the last state of the RNN with a single-layer network with a ReLU non-linearity. Flow amortization then allows to condition the flow on past steps $\mathbf{z}_1...\mathbf{z}_\tau$, and is not restricted to a given context size because of the RNN underlying process. Amortized temporal flows can then be naturally integrated in the ELBO, as the distributions $q(\mathbf{z}_t|\mathbf{z}_{t-1})$ are tractable. The complete ELBO is then expressed

$$\mathcal{L}_{flow} = \mathbb{E}_{\mathbf{z}_0...\mathbf{z}_\tau \sim q(\mathbf{z}_{t\leq\tau}|\mathbf{x}_{t\leq\tau})}[p(\mathbf{x}|\mathbf{z})] + D_{KL}\big[q(\mathbf{z}_{t\leq\tau}|\mathbf{x}_{t\leq\tau})||p(\mathbf{z})\big] \tag{4.26}$$

$$+ \sum_{k=1}^{K} \log|\det \frac{\partial f_k(\mathbf{h}_\tau)}{\partial \mathbf{z}_{\tau+k-1}}| \tag{4.27}$$

that is entirely tractable.

4.3.3 Gaussian processes regularization

Finally, the third prediction method is to use Gaussian processes as a prior for $\mathbf{z}_{t>\tau}$, using a continuous time regression to model the trajectory $[\mathbf{z}_0....\mathbf{z}_{\tau-1}]$ and then predict the consecutive steps $\mathbf{z}_{t>\tau}$. Gaussian processes are used to get a probability distribution over functions $\mathbf{f}$ verifying $\mathbf{z}(t) = \mathbf{f}(t)$, where the t are fake inputs representing the index of sequence $\mathbf{z}_{t<\tau}$, normalized between 0 and 1. Remembering section 4.9, we then use a Gaussian prior $\mathbf{GP}(\mathbf{0}, \mathbf{K}_{tt})$ to model the regression

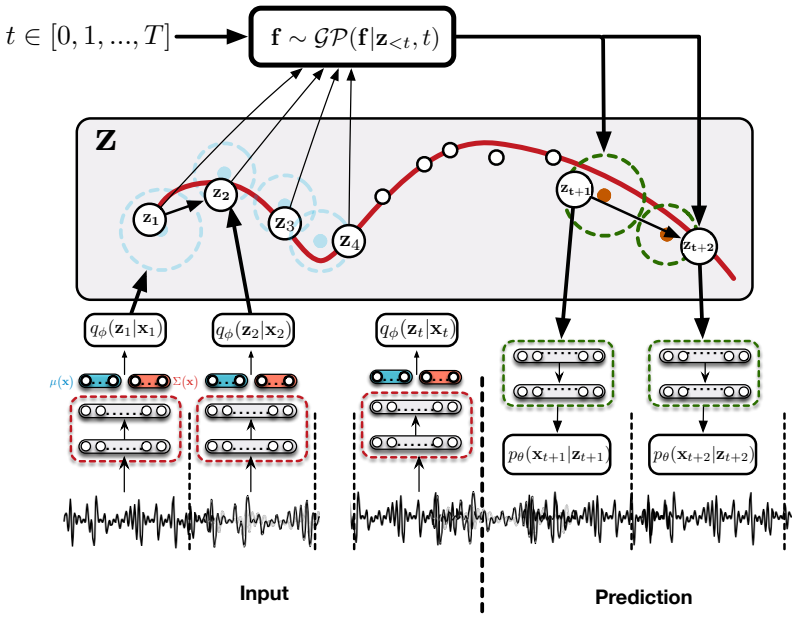


Figure 4.19: We can use the obtained the regression performed by the Gaussian process to infer the following states of the trajectory, using the predictive posterior distribution with non-observed values of t.

function $\mathbf{f}$ conditioned on pairs $\{(\mathbf{z}_t, t)\}$, and obtain the regression over inputs $t^\star$ by using the posterior distribution

$$q(\mathbf{z}|\mathbf{f}, \mathbf{Z}, t^\star) = \mathcal{N}(k(t^\star, t)(\mathbf{K}_{tt} + \sigma_n^2 \mathbb{I})^{-1}\mathbf{z}, \tag{4.28}$$

$$k(t^\star, t^\star) - k(t^\star, t)^\top (\mathbf{K}_{tt} + \sigma_n^2 \mathbb{I})^{-1} k(t^\star, t)^\top) \tag{4.29}$$

to obtain the predictions $\mathbf{z}_{>\tau}$ on prediction times $t^\star = \tau ... \tau + k$.

This Gaussian Processes regression does not suffer from the scalability issues of standard Gaussian processes, at is only involves the inversion of a matrix $\tau \times \tau$, and performs fast as it does not involve any additional module as previous prediction methods. Furthermore, it is also the only method that gives an explicit distribution over $q(\mathbf{z}_{t>\tau}|\mathbf{z}_{t\leq\tau})$, allowing to directly model an overall normal distribution over the full vector $\mathbf{z}$. Introducing the posterior distribution $q(\mathbf{z}|\mathbf{f}, \mathbf{Z}, t^\star)$, we obtain the ELBO

$$\mathcal{L}_{GP} = \mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}[q(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z})] \tag{4.30}$$

$$= \mathbb{E}_{q(\mathbf{z})}[\log p(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}[q(\mathbf{z}_{<\tau}|\mathbf{x}_{<\tau})\|p(\mathbf{z})] \tag{4.31}$$

$$- D_{KL}[q(\mathbf{z}_{>\tau}|\mathbf{f}, \mathbf{Z}, t^\star)\|p(\mathbf{z})] \tag{4.32}$$

that is entirely tractable, as the distribution $q(\mathbf{z}_{>\tau}|\mathbf{f}, \mathbf{Z}, t^\star)$ is a multivariate normal distribution. In this study, we use a simple radial basis function kernel with time length $\ell_{RDF} = \sqrt{3}$ and an input noise variance $\sigma_n^2 = 1e^{-2}$. Note that we could have used the noise variance given from the variational distribution $q(\mathbf{z}|\mathbf{x})$, obtaining a precise estimation of the variances σ_q^2. Moreover, another interesting feature of the proposed GP prediction is to be defined on real time indices $t \in \mathbb{R}$, allowing a sample random latent positions on the predicted curve for continuous time prediction, or to upsample the trajectory to perform time-stretching operations and enforce the temporal consistency of the latent space. Unfortunately, due to a lack of time, we leave this

improvement to a future work.

4.3.4 Learning and evaluation framework.

Database As we want to evaluate the performances of the proposed prediction methods on modeling temporal evolutions, the datasets previously used are not relevant, being mostly composed by static sounds. Hence, we rather rely on the `diva_dataset`, described section 3.2.1, that contains more than 11 thousands of synthesizer sounds with many different dynamical shapes (see fig. 4.20). In this experiment, the models were trained on STFT representations of 60 steps on spectral frames of 2048 bins, pre-processed with a `log1p` non-linearity.

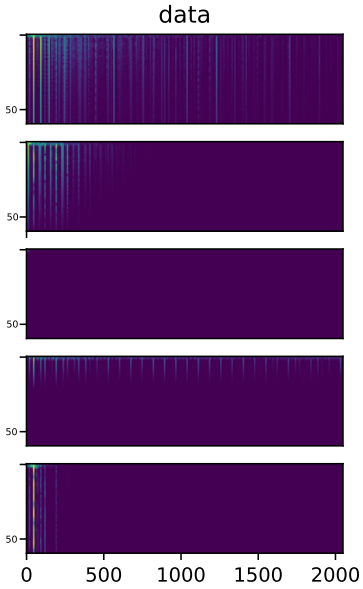


Figure 4.20: Examples taken from the `diva_dataset` database

Models Similarly to previous experiments, we split the dataset between train and tests sets with a balance of 0.8. The architecture of both encoding and decoding functions are defined using convolutional networks of $[64, 32, 16, 8]$ channels and $[3, 5, 7, 9]$ kernels, using ELU non-linearities and batch-norm normalization. For the CPC autoregressive modules, we chose a GRU with one layer of 200 units, and the amortized flows are obtained from simple linear layers with a ReLU non-linear units. The VAEs and prediction modules were trained jointly on 100 epochs with an ADAM optimizer a learning rate of $1e-4$, with a D_{KL} regularization with $\beta = 1$ and a warm-up of 10 epochs. We also found beneficial to add a *teacher warm-up* procedure during the first 5 epochs, initially decoding the latent positions coming from the encoder and progressively replaced by the ones provided by the prediction module. This procedure allows to train the latent space directly on the signal during the first epochs, increasing the robustness of the latent process. The number of predicted steps is 30, for 30 input steps, amounting to a total sequence length of 60.

Table 4.1: Table of the results obtained with the proposed prediction methods on the overall sequence. The error is normalized stepwise, to be compared with the results obtained in section 3. The results out of parenthesis are the obtain with train partition of the dataset, while numbers within are the errors obtained with the test partition.

	$-\log p(\mathbf{x}\|\mathbf{z})$	SC	ISD	$D_{KL}[q\|p]$	$MMD[p,q]$	$TC[p,q]$
CPC prediction	693.61 (*705.45*)	0.671 (*0.682*)	4.22e-5 (*1.58e-4*)	21.49 (*21.87*)	4675 *1075.73*)	347.04 (*360.74*)
Flow prediction	680.84 ***(677.58)***	2.00e-2 (*2.04e-2*)	4.32e-5 ***(1.63e-4)***	**24.45 (23.72)**	5570 *1198.32*)	393.52 (*361.23*)
GP prediction	**677.00** (*692.59*)	**3.16e-2 (3.31e-2)**	**4.29e-5 (1.63e-4)**	24.58 (*24.04*)	**3476 (772.40)**	**231.94 (232.67)**

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^1	ℓ^2
CPC	673.63 (*685.73*)	105.5 (*102.03*)	98.40 (*111.53*)
Flow	**661.06(*658.36*)**	94.67 (***91.90***)	**94.60 (*97.87*)**
GP	674.53 (*702.00*)	**89.83** (*105.9*)	106.93 (*106.76*)

Table 4.2: Table of the results obtained with the proposed prediction methods, only on the predicted part.

Reconstruction evaluation. We show the results obtained with the overall sequence in table 4.1, providing reconstruction and regularization estimators, and the reconstruction results obtained only on the prediction part 4.2.
We observe that the reconstruction results obtained by the three methods are close, flow and GP methods performing slightly better results than the CPC. If we only observe the prediction part, the best performance is achieved by the flow prediction method, even if all the results are also very close. To study qualitatively the reconstruction probabilities, pairs of obtained reconstructions are depicted fig. 4.21. We can see that the predictions manage to catch inner variations of the spectra, showing that the modules efficiently model inner the inner dynamics of each data examples. Though, we can see that CPC has a certain tendency to produce undesired artifacts, as can be seen on the second example that are absent of other prediction methods. Conversely, GP predictions seem to smooth the generated sound, that is coherent with the inner smoothing effect of Gaussian process (see next paragraph). Finally, flow predictions are almost perfect, managing to catch even subtle variations of the data. The qualitative properties of these predictions methods are thus a little different, and can then be optimally chosen depending on the nature of the dataset.

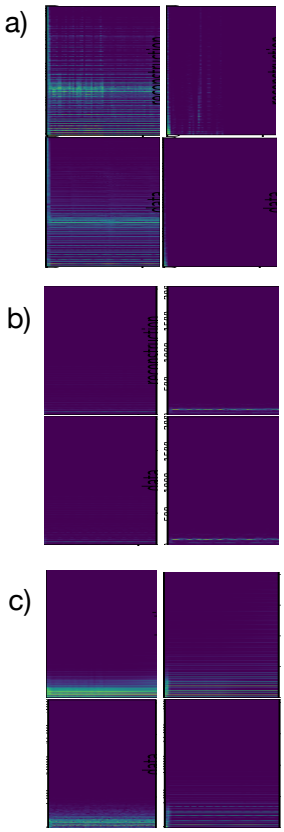


Figure 4.21: reconstruction examples (top) and original spectra (bottom) for (a) CPC (b) Flow (c) GP prediction methods.

Latent evaluation. Regarding latent spaces, we can see from the results table 4.1 that the GP prediction method provides the best regularization result. However, the Kullback-Leibler divergence are low and for the three methods, showing that all of them provide accurately regularized latent spaces. Differences between the methods can be still observed with the MMD and to Total Covariance, the best scores being obtained by the GP while the worst are obtained with the temporal flows. This points that the GP prediction method, the most restrictive in terms of geometrical constraints, manage to successfully regularize the latent space without (or a little) altering the reconstruction abilities of the model.
To get more qualitative insights of the *temporal organization* of the space, we can observe the paths obtained by forwarding two different sounds from the dataset, and to generate from several interpolation steps between the two obtained trajectories. Examples of trajectories for the three methods are depicted fig. 4.23, providing interesting insights on their respective behavior (projected on a PCA extracted from the latent projects of the full dataset). First, we can see that the trajectories obtained with the CPC method are favoring stable latent "nodes", the distribution of projection points corresponding to the stationary part of the signal lying on a narrow latent region. This is enforced by the slow-feature encourages by the CPC training (see fig. 4.22), and then show that CPC influences the topology of the latent space around attractors representing specific spectral distributions. Regarding temporal flows, we can see that the high capacity modeling abilities of NFs allow to model arbitrary trajectories of the latent space, showing that an expressive prediction method does not regularize the representation and rather fits the latent topology ; however, this flexibility allow to provide the best reconstruction results. Finally, we can see that

GP prediction is the one that conditions the most the latent topology, providing very smooth trajectories that correspond to curves sampled from a GP process. This method is then the one that perform the strongest regularization of the latent space, hence explaining its high generalization scores but also its lowest (albeit satisfactory) reconstruction results.

Hence, we can see that these three prediction methods, while providing satisfying results, are qualitatively different, and apply diverse implicit regularization about the temporal behavior of the latent space. These methods can be then wisely chosen depending on the desired structure of the representation, from the less restraining and the more efficient in terms of reconstruction (flows) to the most regularizing (GP), or providing compact latent evolutions (CPC). However, while allowing to regularize the latent space on temporal criteria, prediction methods do not yet allow to *model* the trajectories of the dataset. Hence, an additional modeling is required, that would allow us to extract features from the obtained trajectories.

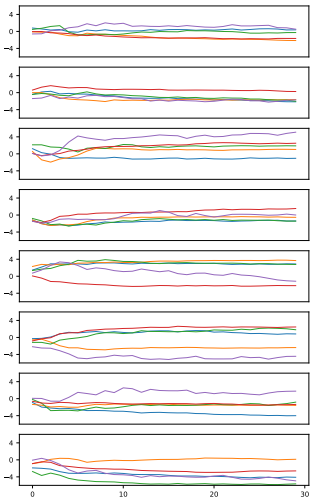


Figure 4.22: CPC curves obtained for 5 random examples, each dimension plotted apart. We can see that the CPC are evolving slowly, contrary to the corresponding latent curves.

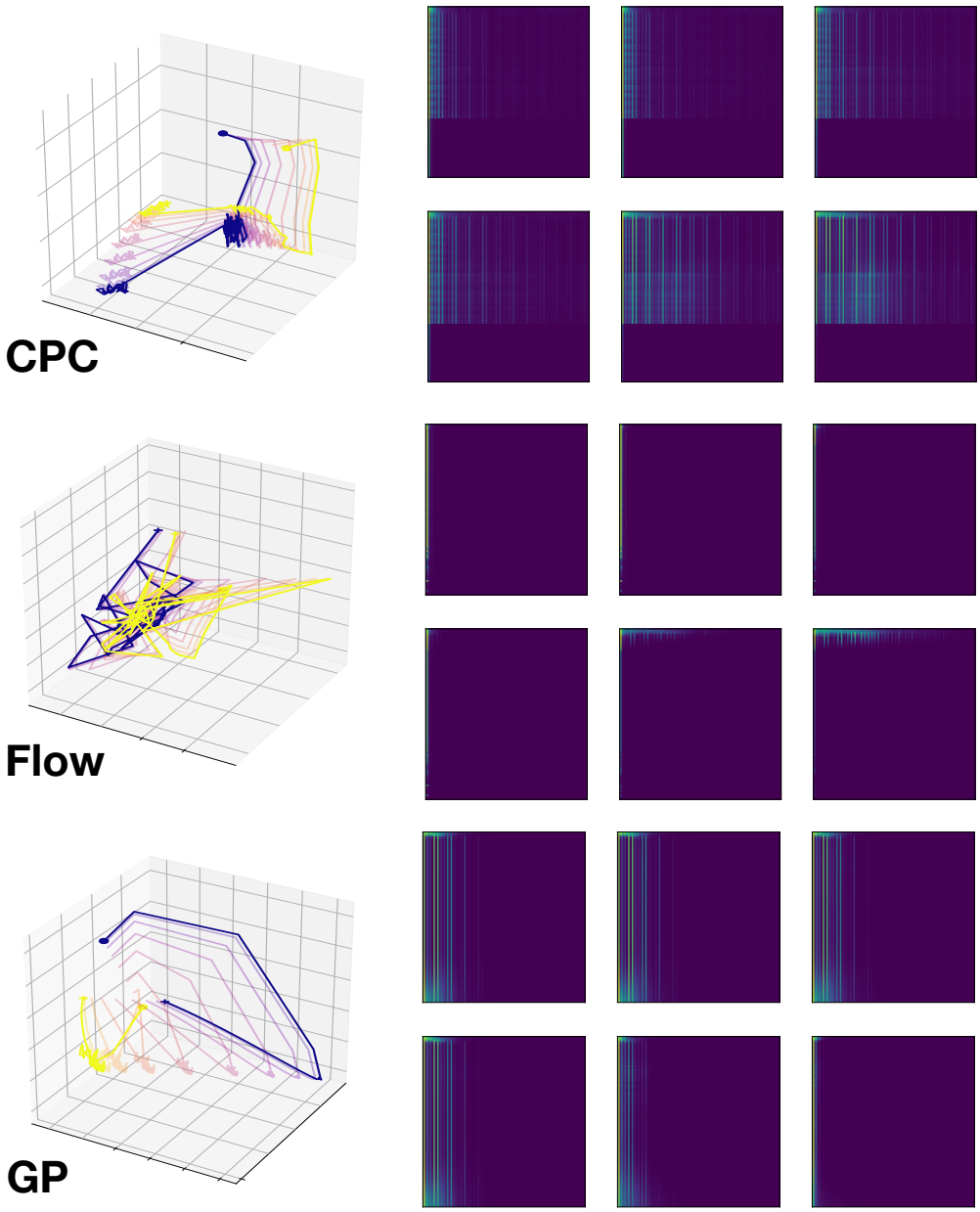


Figure 4.23: Examples of morphing between an original (blue) and target (yellow) trajectories (through PCA) obtained from dataset examples for (top) CPC (middle) Flow (bottom) GP prediction modules.

4.4 Hierarchical latent spaces and raw waveform learning

In the previous section, we investigated three different methods to perform prediction directly on the latent space, modeling dynamics $q(\mathbf{z}_{t>\tau}|p(\mathbf{z}_{t\leq\tau})$ with a broader prediction scope than the one provided by state-space models approaches. The inner idea of these methods was that training jointly the model on inference, generation and prediction objective was constraining the latent space to reflect temporal features of the data. We saw that these three prediction methods was providing substantially different behaviors, that could thus be chosen depending of the target application or dataset. However, these prediction methods do not help us to really extract temporal features of the data, as each state vector $\mathbf{z}_t$ represents a single data step $\mathbf{x}_t$.

Reminding section 4.2.3, there are two different manners to encode temporal features in the latent representation : whether formulating the VAE as a state-space model, then having one latent vector $\mathbf{z}_t$ per data input $\mathbf{x}_t$ and a suitable transition model $p(\mathbf{z}_t|\mathbf{z}_{t-1})$, or using recurrent encoders / decoders to obtain a global latent vector $\mathbf{z}$ for the whole sequence. We investigated the first approach during previous section, showing the ability of latent spaces to incorporate temporal features with a given prediction model. However, the second is also interesting, as it extracts global features from a target sequence and then also implicitly models its dynamics.

We then propose to combine both approaches with hierarchical latent spaces, modeling the trajectories observed in the obtained representation. This approach, called *ShrubVAE*, allows us to represent different temporal scales of the input, encoding the overall structure of the sequence at top-level, and progressively model the local temporal structure by upsampling across layers. This method then allows to considerably extend the scope of prediction process, as a single vector corresponds to several steps of the sequence. Furthermore, this architecture allows us to access the retrieved features corresponding to different time scales, then performing multi-scale analysis. We can then analyze and generate with different temporal scopes by choosing the interaction layer, from a low-level state-space model to a full sequence at the top.

4.4.1 ShrubVAE : Multi-layered models for multi-scale learning

We then propose a hierarchical latent model performing multi-scale analysis and generation of data signals, that is based on a progressive down-sampling of the incoming time series using variational learning. This approach models long-term dependencies with a few latent vectors in the top layer of the model, while intermediate layers extract local dynamical information. We also perform the prediction methods

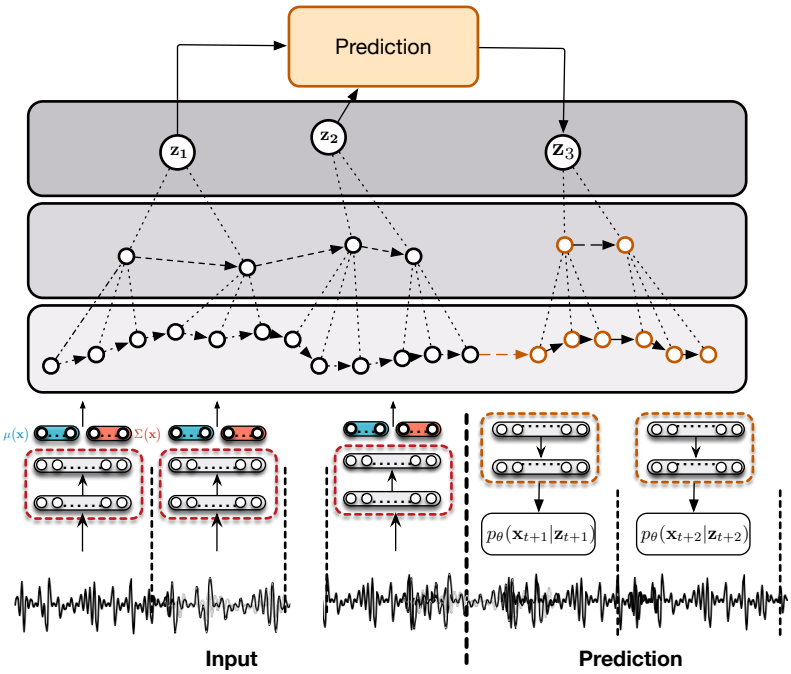


Figure 4.24: Prediction is performed at the higher level of temporality, allowing to significantly reduces the number of prediction states needed to predict the signal.

proposed on the top layer of this model, and requiring less prediction steps as each top latent features correspond to several samples of the sequence. These prediction methods should perform better, as it requires a fewer number of steps and are supposedly performed on features containing high-level temporal information. We then evaluate the proposed model by comparing this model to the previous prediction performances obtained with single-layered VAEs, evaluating it the proposed hierarchical architecture is easing the prediction.

Hierarchical latent spaces. While AEVB convincingly extracts compressed representations from high-dimensional data, the number of latent dimensions is often constrained by the reconstruction capacity of the model. Some approaches then proposed to alleviate this issue by adding stochastic layers on top of the variational models, extracting higher-level latent features directly from $\mathbf{z}_0$. Such approaches where popularized by ladder networks, a deterministic auto-encoder composed from multiple stacks of latent codes, that were trained on a reconstruction error between each top-down and down-top processes [379]. In the domain of variational approaches, this was developed by Rezende & al. with *Deep Latent Gaussian Models*, proposing a multi-layer approach to AEVB using a deterministic encoder across successive layers. A full variational approach to multi-layer models were consequently proposed by Kaae & al., based on the conditional

inference & generation processes with L layers [184]

$$q(\mathbf{z}_L, ...\mathbf{z}^1|\mathbf{x}) = q(\mathbf{z}^1|\mathbf{x}) \prod_{l=2}^{L} q(\mathbf{z}_l|\mathbf{z}_{l-1}) \tag{4.33}$$

$$p(\mathbf{z}_L, ...\mathbf{z}^1, \mathbf{x}) = p(\mathbf{z}^L)p(\mathbf{x}|\mathbf{z}^1) \prod_{l=1}^{L-1} p(\mathbf{z}^{l-1}|\mathbf{z}^l) \tag{4.34}$$

where each conditioned model is parametrized as a diagonal normal distribution, similarly to the VAE. The obtained ELBO is then

$$p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}^1,...\mathbf{z}^L \sim q(\mathbf{z})}[p(\mathbf{x}|\mathbf{z}^1)] + D_{KL}[q(\mathbf{z})||p(\mathbf{z}_L)] \tag{4.35}$$

$$+ \sum_{l=2}^{L-1} D_{KL}[q(\mathbf{z}^l|\mathbf{z}^{l-1})||p(\mathbf{z}^l|\mathbf{z}^{l+1})] \tag{4.36}$$

The L regularization terms $D_{KL}[q(\mathbf{z}^l|\mathbf{z}^{l-1})||p(\mathbf{z}^l|\mathbf{z}^{l+1})]$, that are all tractable, can then be understood as a criterion that matches the layer-wise distributions provided by encoders and decoders, ensuring the consistency of the representation. Unfortunately, further investigations of these hierarchical models showed that these higher-level latent variables were rather inexpressive, and barely used by the low-level decoding model $p(\mathbf{x}|\mathbf{z}^1)$. Information sharing between encoding and decoding processes was consequently proposed by Kaae with Ladder Variational Networks, improving the inference of higher-level latent variables $\mathbf{z}^l$ [380] by explicit information sharing between encoders and decoders. However, the gradient dissipation across higher levels often weakens their representativeness, requiring an additional criterion as semi-supervised task to be non-degenerated.

Multi-scale inference with hierarchical spaces. ShrubVAE rather proposes to use hierarchical latent layers to model different time scales, amounting to successive down-/up-sampling (resp. during inference / generation) steps of the latent sequences. This progressive dilation task prevents the higher-level representations to be data-insensitive, as each feature vector has to represent the temporal information of the incoming sequence. We use simple generative and inference models as above, excepting a window $\mathbf{z}^l_{t_l,...t_l+\nu_l}$ of size ν_l is encoded to one single latent vector $\mathbf{z}^{l+1}_{t_{l+1}}$ using a RNN encoder. The inference distribution can then be formulated as

$$q(\mathbf{z}^0_t|\mathbf{x}_t) = \mathcal{N}(\boldsymbol{\mu}_{q,0}(\mathbf{x}_{t_0}), \boldsymbol{\sigma}_{q,0}(\mathbf{x}_{t_0})) \tag{4.37}$$

$$q(\mathbf{z}^l_{t_l}|\mathbf{z}^{l-1}_{t_l\nu_l:(t_l+1)\nu_0}) = \mathcal{N}(\boldsymbol{\mu}_{q,l}(\mathbf{z}^{l-1}_{t_l\nu_l:(t_l+1)\nu_l}), \boldsymbol{\sigma}_{q,l}(\mathbf{z}^{l-1}_{t_l\nu_l:(t_l+1)\nu_l}) \tag{4.38}$$

$$q(\mathbf{z}^L|\mathbf{z}^{L-1}) = \mathcal{N}\big(\boldsymbol{\mu}_{q,L}(\mathbf{z}^{L-1}), \boldsymbol{\sigma}_{q,L}(\mathbf{z}^{L-1})\big) \tag{4.39}$$

such that a downsampling of factor ν_l is applied at each layer l. We choose to not perform any downsampling on the first layer $\mathbf{z}^0$, modeling it as a state-space representation of the incoming data, that can be accessed directly to generate data elements separately. The parameters

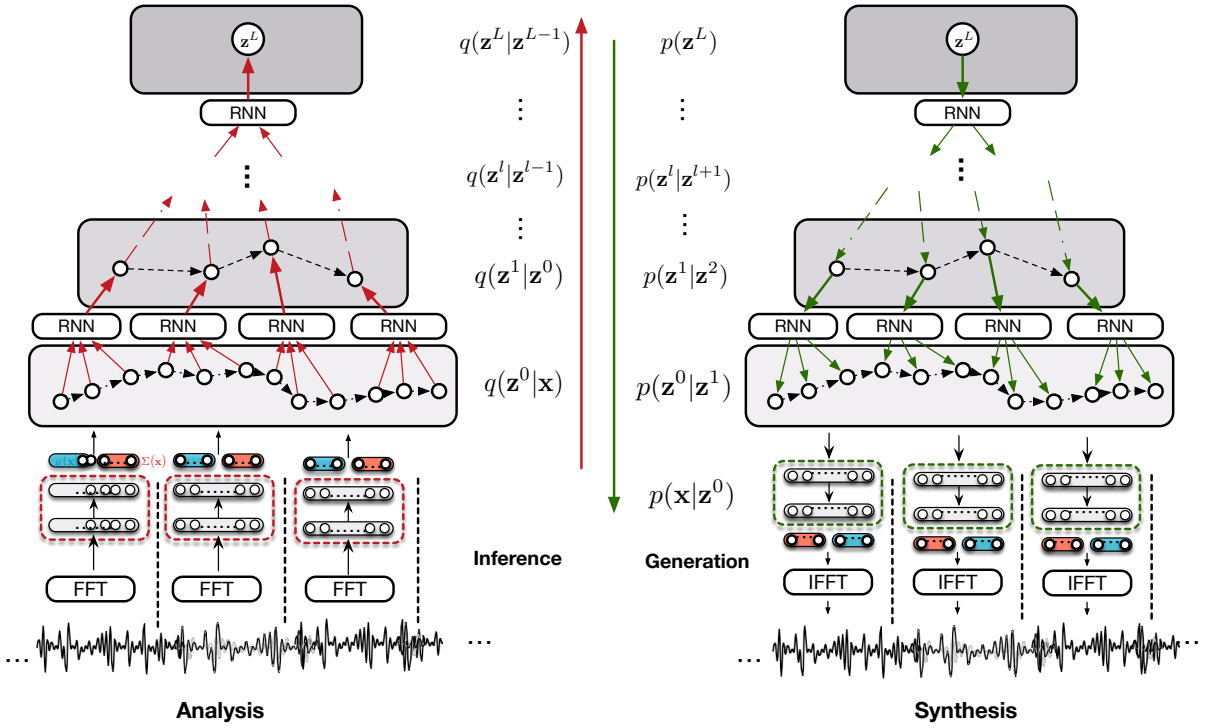


Figure 4.25: Probabilistic graph explaining the hierarchical multi-scale architecture of `ShrubVAE`. This architecture is based on multi-layer dynamical compression, thus performing Bayesian down/up-sampling that allows to efficiently model the temporal structure of the incoming signal.

$\boldsymbol{\mu}_{q,l}(\cdot), \boldsymbol{\sigma}_{q,l}(\cdot)$ are obtained in a similar manner to the RVAE encoding model (see sec. 4.2.3), such that

$$\mathbf{h}_t = RNN(\mathbf{z}_t^{l-1}, \mathbf{h}_{t-1}) \qquad t \in [t_l \nu_l ... (t_l + 1)\nu_l] \tag{4.40}$$

$$\boldsymbol{\mu}_{q,l}\left(\mathbf{z}_t^{l-1}\right) = \mathbf{A}_{\mu}^{\top}\mathbf{h}_{(t_l+1)\nu_l} \tag{4.41}$$

$$\boldsymbol{\sigma}_{q,l}\left(\mathbf{z}_t^{l-1}\right) = \mathbf{A}_{\sigma}^{\top}\mathbf{h}_{(t_l+1)\nu_l} \tag{4.42}$$

so we take the final hidden vector of a RNN to encode $\mathbf{z}^l$, trained on a ν_l-sized window of $\mathbf{z}^{l-1}$. The temporal scope taken by the RNN is quite restricted, such that we do not suffer any of the gradient degeneracy problems evoked in sec. 4.2.1. Intuitively, the vector $\mathbf{z}^l$ can thus bee understood as encoding the local dynamics of the sequence fragment $\mathbf{z}_{t:t+\nu_l}^{l-1}$, projected on a higher-level latent space. Note that, with this system, a window ν_l is held fixed during the training process, but can be arbitrarily set when performing inference or generation.

The generation model is mirroring the encoding process, and *upsamples* an incoming vector $\mathbf{z}_{t_{l+1}}^{l+1}$ of a factor τ_l, obtaining a vector $\mathbf{z}_{t_{l+1}}^{l} ... \mathbf{z}_{t_{l+1}+\tau_l}^{l}$,

through the following generative process

$$p(\mathbf{z}^L) = \mathcal{N}(\mathbf{0}, \mathbb{I}) \tag{4.43}$$

$$p(\mathbf{z}^{l-1}_{t_l \nu_l : (t_l+1)\nu_L} | \mathbf{z}^l_{t_l}) = \prod_{\tau=1}^{\nu_l} \mathcal{N}\left(\boldsymbol{\mu}_{p,l}(\mathbf{z}^l_{t_l}, \tau), \boldsymbol{\sigma}_{p,l}(\mathbf{z}^l_{t_l}, \tau)\right) \tag{4.44}$$

$$p(\mathbf{x}_{t_0} | \mathbf{z}_{t_0}) = \mathcal{N}\left(\boldsymbol{\mu}_{p,0}(\mathbf{z}_{t_0}), \boldsymbol{\sigma}_{p,0}(\mathbf{z}_{t_0})\right) \tag{4.45}$$

$$\tag{4.46}$$

such that the joint distribution of the upsampled vector $\mathbf{z}^{l-1}$ is factorized over time steps t_{l-1}. Similarly, the parameters of successive distributions $\boldsymbol{\mu}_{p,l}(\cdot, \tau), \boldsymbol{\sigma}_{p,l}(\cdot, \tau)$ is obtained with a RNN, such that

$$\mathbf{h}_t = RNN(\mathbf{z}^{l+1}_{t_{l+1}}, \mathbf{h}_{t-1}) \qquad t \in [t_{l+1}\nu_{l+1} ... (t_{l+1}+1)\nu_{l+1}] \tag{4.47}$$

$$\boldsymbol{\mu}_{p,l}\left(\mathbf{z}^{l+1}_t\right) = \mathbf{B}^\top_\mu \mathbf{h}_t \tag{4.48}$$

$$\boldsymbol{\sigma}_{p,l}\left(\mathbf{z}^{l+1}_t\right) = \mathbf{B}^\top_\sigma \mathbf{h}_t \tag{4.49}$$

where the successive distributions for the steps $[t..t+\tau]$ are obtained by iterating a RNN τ times using the replicated input $\mathbf{z}^{l+1}_{t_{l+1}}$. This system then allows to generate several latent variables from a single one, each step being conditioned on the previous with a recurrent mechanism, then generating local dynamics of the curve. Despite the down-/up-sampling procedures performed by stacked latent layers, the ELBO (4.35) is not modified, as the shape of latent trajectories are the same during encoding and decoding processes, and the distributions are factorized across time steps.

4.4.2 Learning procedure and progressive hierarchical warm-up

All the parameters of the ShrubVAE are trained end-to-end using back-propagation algorithms, similarly to regular variational auto-encoders. As both generative and inference models are defined as normal distributions with normal covariances, the reparemetrization trick applied by Kingma & al. is still applicable, such that the convergence of the whole training is fast and guaranteed [22]. In this article, we resort to the ADAM optimization scheme [123], using a learning rate of $10e^{-4}$. However, despite the use a warm-up procedure for the top layer and the use of the MMD at the lower level, the training process of the ShrubVAE can still encounter difficulties, due to the network depth. During experiments, we observed that this was due to the balance between inference and generation process, where the effect of the regularization where canceling the information coming from the data during early-training. While similar to the latent pruning shown in standard and multi-layered VAEs, this effect is emphasized by the down-/up-sampling process of the model.

To counter this unwanted tendency, we add to the top *warm-up* layer an additional procedure, called *progressive hierarchical warm-up*, that

schedules the information fed into the different decoders. This scheme aims at introducing the information provided by higher latent layers progressively during early training, such that each encoder / decoder pairs are suitably trained before being given to the lower levels. This is done by regularly increasing the parameter α of a Bernoulli distribution, that is sampled to select whether the input of the decoder $p(\mathbf{z}^l|\mathbf{z}^{l+1})$ comes from the higher decoding layer or from the corresponding encoding distribution as follows:

$$m_l \sim \text{Bernoulli}(\alpha) \tag{4.50}$$

$$\mathbf{z}_l \sim \begin{cases} q(\mathbf{z}_l|\mathbf{z}_{l-1}) \text{ if } m_l = 0 \\ p(\mathbf{z}_l|\mathbf{z}_{l+1}) \text{ if } m_l = 1 \end{cases} \tag{4.51}$$

This procedure enforces each decoder to use the information coming from the encoder during early training, and are progressively led to use the information from above layers. In this article, we linearly increment the α parameter from 0 to 1 during the first 20 epochs, that corresponds to approximately the mid-term of the training process with the used datasets.

Experimental procedure. To evaluate the ShrubVAE performances, we chose a similar experimental framework than the one used in the previous section. Hence, we first evaluate the generation abilities of the model, comparing it with the results obtained on single-layered VAE (described in the previous section), to ensure that the reconstruction loss does suffer from the down-/up-sampling processes. We then observe the results obtained by each of the three proposed prediction methods above, analyzing how it gains from the temporal hierarchy provided by the model. To compare accurately on these criteria, we then take sequences of the same length than in the previous experiment, such that the total sequence length is 60 with a predictive scope of 30, performed on magnitude spectral bins $\mathbb{R}^{2048}$. All the recurrent modules used to perform dynamical compression are 1-layer units of 100 hidden units, chosen to be very light. We thus trained a total of 6 models, corresponding to the three prediction methods with two different hierarchical architectures:

- a two-layered `ShrubVAE-2l` with 10× down-sampling, such that the top latent vector sequence $\mathbf{z}^1$ has a total step amount of 6 elements, reducing the amount of predicted states to 3
- a three-layered `ShrubVAE-3l` with respective 5× and 3× down-sampling, the first latent sequence $\mathbf{z}^1$ then having a total step amount of 12 and the second latent sequence $\mathbf{z}^2$ a total amount of 4 elements, reducing the amount of predicted states to 2.

we used the same prediction settings than in the previous sections, using a 10 epochs prediction warm-up over a total amount of 80 epochs. A D_{KL} divergence were used in every layer $l > 1$, while a MMD loss was used in the first latent layer to provide the best reconstruction results.

Table 4.3: Full table of results obtained with 1-layer (previous section), 2-layer and 3-layer `ShrubVAEs`, for all models. For regularization losses, the layer-wise divergences are sorted in increasing order.

	$-\log p(\mathbf{x}\|\mathbf{z})$	SC	ISD	$D_{KL}[q\|p]$	$MMD[p,q]$	$TC[p,q]$
CPC prediction						
1-layer	693.61 (*705.45*)	0.671 (*0.682*)	4.22e-5 (*1.58e-4*)	21.49 (*21.87*)	4675 *1075.73*)	347.04 (*360.74*)
`ShrubVAE-2l`	722.76 (*730.17*)	0.394 (*0.391*)	4.28e-5 (*1.61e-4*)	2.81e25 (*2.8e25*) 3.91 (*3.934*)	5158 (*1253.5*) 913.30 (*222.24*)	775.12(*865.78*) 52.83 (*58.05*)
`ShrubVAE-3l`	783.58 (*771.37*)	0.165 (*0.168*)	4.47e-5 (*1.64e-4*)	5.59e25 (*5.51e25*) 1.01e+09 (*80.87*) 1.78 (*1.78*)	12539 (*2775.7*) 98.86 (*21.47*) 303.15 (*68.723*)	1308.3 (*1168.79*) 82.31 (*85.39*) 12.64 (*13.04*)
Flow prediction						
1 -layer	680.84 (*677.58*)	2.00e-2 (*0.204*)	4.32e-5 (*1.63e-4*)	24.45 (*23.72*)	5570 (*1198.32*)	393.52 (*361.23*)
`ShrubVAE-2l`	720.37 (*730.66*)	2.60e-2 (*2.61e-2*)	4.46e-5 (*1.67e-4*)	nan (*nan*) 4.08 (*3.49*)	4289 (*908.56*) 976.48 (*205.68*)	572.23 (*537.48*) 46.87 (*58.05*)
`ShrubVAE-3l`	716.14 (*732.47*)	0.120 (*0.117*)	4.22e-5 (*1.58e-4*)	5.59e25 (*5.51e25*) 1.01e+09 (*80.87*) 1.78 (*1.78*)	12539 (*2775.7*) 98.86 (*21.47*) 303.15 (*68.723*)	1308.3 (*1168.79*) 82.31 (*85.39*) 12.64 (*13.04*)
GP prediction						
1-layer	677.00 (*692.59*)	3.16e-2 (*3.31e-2*)	4.29e-5 (*1.63e-4*)	24.58 (*24.04*)	3476 (*772.40*)	231.94 (*232.67*)
`ShrubVAE-2l`	666.09 (*667.4*)	4.03e-2 (*4.30e-2*)	4.14 e-5 (*1.56e-4*)	nan (*nan*) 4.12 (*4.06*)	4437.7 (*975.94*) 780.61 (*174.0*)	568.3(*491.68*) 39.11 (*35.72*)
`ShrubVAE-3l`	695.00 (*707.97*)	6.44e-2 (*6.26e-2*)	4.34e-5 (*1.62e-4*)	2.32+25(*2.30e25*) 3.69 (*80.87*) 2.86 (*2.88*)	7437 (*2775.7*) 39.8 (*9.01*) 607.3 (*68.723*)	546.79 (*550.02*) 162.07 (*158.92*) 20.63 (*18.49*)

Reconstruction results. The results obtained for the full sequence are listed table 4.3, and the reconstruction of only the predicted vectors are displayed fig. 4.4. We can see that the `ShrubVAE` the down-sampling process across layers does not impact significantly the reconstruction error. The overall uncertainty $-\log p(\mathbf{x}|\mathbf{z})$ of the model globally increase, but the deterministic spectral losses slightly change, and are even decreasing in the case of CPC. A interesting fact is that, while the best reconstruction performance of all the sequence is given by using flow prediction method, similarly to single-layered VAEs, it is the GP prediction method that has the best performances on the predicted sequence, and is barely influenced by the number of added layers. These results then confirm the intuition given previously about the generalization abilities of this method. Nevertheless, for every method the ShrubVAE accurately performs dynamical compression, identifying the overall sequences with only one latent vector without influencing too much the reconstruction loss. However, we do not especially gain in terms of reconstruction from predicting a lower amount of steps, we ensure that the system preserves the overall consistency of the input sequence.

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^1	ℓ^2
CPC			
1l	673.63 (*685.73*)	105.5 (*102.03*)	98.40 (*111.53*)
2l	676.36 (*680.16*)	106.8 (*117.16*)	66.40 (*117.23*)
3l	673.63 (*734.67*)	105.5 (*136.03*)	98.40 (*129.5*)
Flow			
1l	661.06(*658.36*)	94.67 (*91.90*)	94.60 (*97.87*)
2l	666.94 (*660.6*)	106.8 (*98.5*)	99.13 (*99.16*)
3l	697.10(*713.7*)	112.6 (*120.67*)	117.2 (*125.5*)
GP			
1l	674.53 (*702.00*)	89.83 (*105.9*)	106.93 (*106.76*)
2l	642.23 (*646.6*)	93.40 (*93.4*)	89.80 (*92.10*)
3l	707.97 (*662.32*)	99.16 (*105.37*)	99.87 (*107.07*)

Table 4.4: Results obtained with the ShrubVAE with the three prediction methods, only on the predicted part of the sequence.

Regularization results. Latent losses are also depicted table 4.3, allowing us to describe the overall structure of the latent space of each layer. As we are using multi-layer latent architectures, only the last layer is regularized with an isotropic normal distribution $\mathcal{N}(\mathbf{0}, \mathbb{I})$, such that lower level latent losses rather reflect the divergence between encoding and decoding distributions $q(\mathbf{z}^l|\mathbf{z}^{l-1})$ and $p(\mathbf{z}^l|\mathbf{z}^{l+1})$. We can see that, as we are using MMD to regularize the first latent layer, the D_{KL} is very high. By analysing the decoding projections, we found that the decoding distributions $p(\mathbf{z}^1|\mathbf{z}^2)$ had very low variance, while inference distributions $q(\mathbf{z}^1|\mathbf{x})$ where much broader, explaining the very high values taken by the divergence. However, we can see that higher-level latent spaces are well regularized, especially in `ShrubVAE` with three

layers where the top latent space seems to be very close to an isotropic normal distribution, without providing degenerated representations. This can be seen by observing the latent interpolations depicted figure 4.26, where we can see that the trajectories are significantly smoother for each model. This seems to point out that, in addition to down-sampling, hierarchical latent spaces allow to regularize the temporal behavior of the sequences, thus accurately providing a regularized representation of its inner dynamics. This results are very encouraging, and are worth to be investigated on higher temporal scopes.

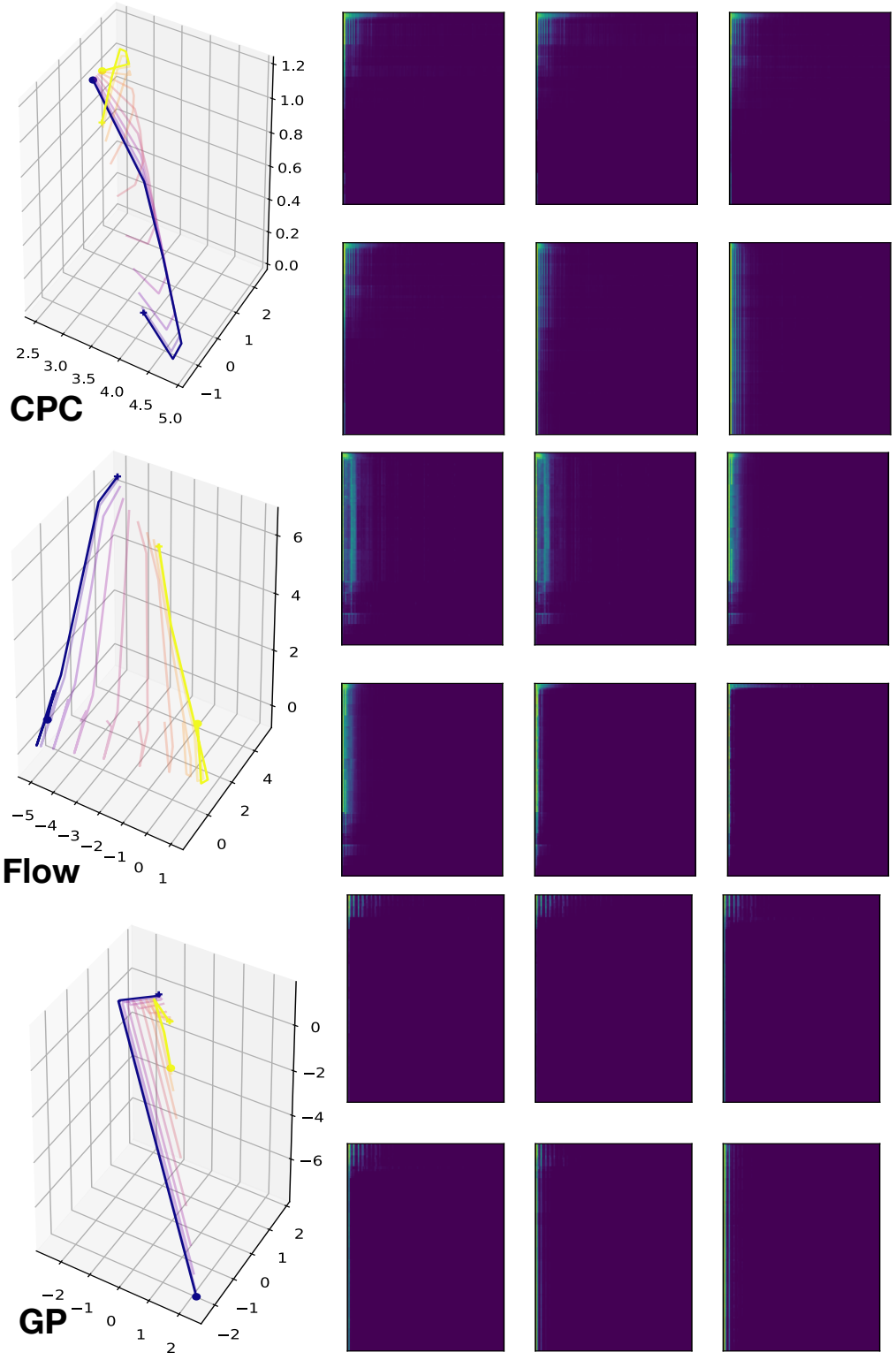


Figure 4.26: Examples of trajectory morphing between two examples of the dataset for (top) CPC (middle) Flow (bottom) GP prediction modules.

Composing and performing with generative spaces

5

In this chapter, we tackle a different aspect of our work that fosters the creative use of the methods developed previously for musical applications. Indeed, while machine-learning techniques are now widely investigated in numerous scientific domains, our main motivation was to develop a framework *design* a generation tool for sound synthesis. Our objective in then two-sided : from one hand, designing machine-learning techniques that are able to opens new technical possibilities for sound synthesis, and from the other hand starting from musical practices to motivate the development of new models. This joint scientific and musical approach is to us seminal in computer music, such that we also aim to develop an usable toolbox, that could be used by both expert and non-export users, and the creation of a musical piece, to explore the creative abilities of the proposed framework.
Thus, in this section we will invest another approach, and adopt a *research & creation* method to the framework development. Indeed, the proposed analysis/synthesis technique differ from synthesis methods developed so far by three aspects : the control space it automatically extract from the data, providing an alternative representation even in the case of sounds synthesized with simple techniques (such as additive synthesis), the diversity sounds it can produce from a limited amount of examples, and the data-centered approach of these methods, integrating the choice of the dataset and of the model architecture in the creative workflow. We will first introduce the developed toolbox, `vsacids`, and the various features it implements. We will then experiment this workflow with the composition of a performance, *ægo*, exploring the latent space with a reinforcement-learning agent interacting with a human performer. This work, done jointly with Hugo Scurto, was accepted at the CMMR2019 conference as both as a musical performance and technical paper.

5.1 `vsacids` : a toolbox for variational generation of musical signals

As we aim to provide an analysis-synthesis framework than can be use by both expert and non-expert users, we have to develop several layers of interaction to allow both customization of the algorithm and high-level usage. We then developed a library, called `vsacids`, that allows easily to train models and to use it real-time or non real-time use cases. This library was conceived in a modular way, allowing deeper customization of models, easily implementation of model improvements. The library also provide methods audio data import, model definitions, and training routines, such that the implemented models can be easily trained by just defining a signature. It also implements high-level methods for non-expert usage, automatically proposing an

architecture for a given dataset with high-level descriptors. This library is using the Python language, and is based on the open-source framework PyTorch* [381], an elegant machine learning framework that allows dynamical definitions of machine-learning models. This library is distributed as a python package, that is open-source be available here : https://github.com/domkirke/vschaos_package.

5.1.1 Architecture and design

The vschaos library is based on three seminal classes, Dataset, Model, and Loss, that can be sub-classed to implement new models or specific behaviors for particular applications. The architecture of vsacids is built around three main classes:

Data format. The Dataset class is an high-level class for collecting data that allow useful routines for data import, transformation, metadata processing, and asynchronous data loading for big datasets. This object assumes a formatted dataset format, that allows it to build a data structure that holds various formats of metadata, so it naturally integrates with the other modules. This dataset structure is defined as shown in figure 5.1, sorted in three folders : a data folder, that contains the audio files, the analysis folder, that contains a list of audio transforms used for training, and a *metadata* folder, that contains a list of tasks. This simple structure allows the Dataset object to organize the dataset, and to automatically retrieve an efficient data structure for model training. We also provide a toy dataset generator, based on the grid sampling of parameters of a digital synthesizer, to provide a fast method methods to evaluate the emerging properties of a model configuration.

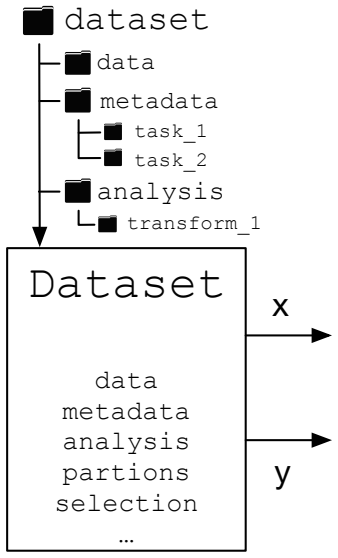


Figure 5.1: The Dataset class, organising data and metadata from audio transforms

Model definition. The Model class is the base of all the models trainable by the library, and implements the three methods encode, decode and forward that are used to train and use the model. All the models of the library can be defined by a *signature* [input_params, latent_parms] , plus model dependent arguments, that defines the variational and generative distributions of the model. These two specifications are defined by an attribute dim, that specifies their dimensions, and dist, that specifies the distribution of the layers. The model can be easily multi-layered by defining a list of latent parameters, and every input or layer can be split in an arbitrary number of parts. Each encoding and decoding functions are defined as the conjugation of an hidden module and of a distribution module, such that each transformation is defined on the encoded distribution (see figure 5.2).
The model is also attached two additional objects : a list of optimizers, that are used to optimize the model during training, and a list of *manifolds*, that are low-dimensional transformations saved during interaction. Model also provides specific load and save functions, such that the model can be saved with the performance parameters and other diverse useful information.

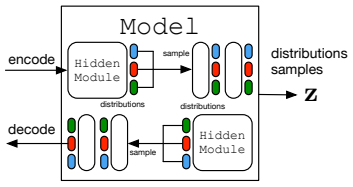


Figure 5.2: The Model class, encoding and decoding the input data

* https://pytorch.org/

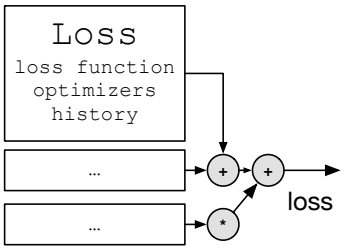


Figure 5.3: The `Loss` class implements different loss functions, providing additional features such as loss history tracking, and simple algebra.

Algebraic losses. The `Loss` object is used for defining the objective criteria of the training. These objects are also defined as models, allowing to embed discriminative criteria such that *DRE* or adversarial regularization directly in the loss object, then lightening the model. In the case of trainable losses, the optimizer is then embedded in the `Loss` object. Also, as the training criterion is a determinant parameter for the obtained model, we wanted to make manipulation of losses convenient and intuitive. We therefore implemented basic arithmetic operators between loss objects (sum, subtraction, multiplication, division, power), such that an additional criterion to the `ELBO` object can be easily formulated by `ELBO + Discriminator[task]`, and can then be experimented by the user without to much effort 5.3. The `Loss` object also provides automatic loss tracking, such that each individual loss can write the values into its memory.

Overall training process. The overall process given a `Dataset`, a `Model` and a `Loss` objects is performed by a `Trainer`, that also implements common machine-learning routines to train the given model. The `Trainer` object takes care of processing the usual steps of training such as data loading, loss optimization, model saving, loss tracking, and test / train procedures. It can also regularly plot the state of the model using an intern `Monitor` object, that synthesizes audio files and plots specific to variational learning regularly during training to provides an accurate monitoring of the process. The plot routines are also conceived hierarchically from low-level to high-level functions, to ease the definition of new monitoring methods for specific models.

Max interface for real-time generation To experiment the capabilities of the proposed framework, we implemented a real-time interface with the `vsacids` library using the *Max* software*, a audio programming widely used in the computer music community. Max is then used a both as a sound engine, inverting in real-time the magnitude spectrograms sent by the generative model, and as a user interface, sending to the model the generative coordinates sent by the user (see fig. 5.4). The communication between Max & `vsacids` is done with the OSC protocol using a dedicated Python server, that interfaces the `Model` instance with Max. As we argued, the lightness of obtained models allows us to interact in real time with the latent space, and can thus be efficiently used as a generative synthesis algorithm. Real-time generation also provides a way to evaluate the properties of extracted spaces directly through *interaction*. Furthermore, the recent coupling between Max and the digital audio workstation Ableton Live † allows to use `vsacids` directly into a Live work session, and then enter the creative workflow of most musicians. Navigation in the latent space can thus be processed with automation and MIDI controlling, transforming the `vsacids` models in real synthesizers.

* cycling74.com
† https://www.ableton.com/

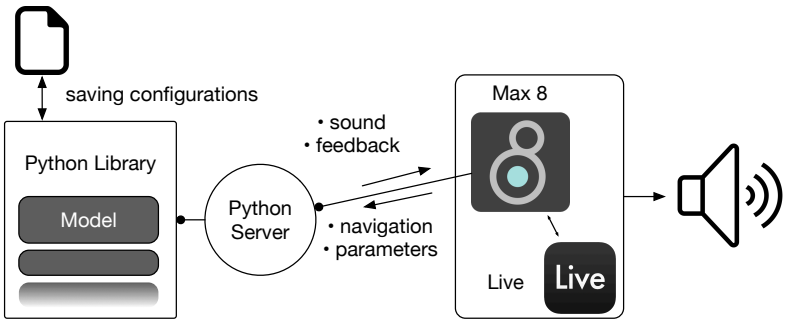


Figure 5.4: Schematics of the vsacids real-time environment.

5.1.2 Workflow

The creative flow of the vsacids framework then decomposes in three steps : *training*, *exploration*, and *exploitation*. Indeed, as the various parameters and models of the framework are of prime importance for the obtained properties of the representation, we involve the training of the algorithm as part of the framework's practical use. Some machine learning algorithms focused on interaction and usabiliity, such as the Wekinator of Fiebrink & al., opened a new way to investigate the development of this techniques under a different perspective [382]. Similarly, the use of variational methods could allow such uses, as it scales well to little datasets and models that train quite fast, even on CPU. We then separate the exploration of the representation, using the real-time interface developed with Max, and the exploitation of the model, based on several techniques that exploits the representation offline.

Training options. As the training step is involved in the creative process, the choice of the model in terms of architecture, losses, and of some training criterion parameters are determinant for the interaction with the developed synthesis method. While our library is opened to developers and researchers that may want to investigate the proposed synthesis method, we also have to model a user-compliant procedure for non-expert users. We define these conception steps as follows :

- fix a dataset $\mathcal{D}$, that can be taken in a sample bank or generated using the toy dataset generator
- fix the desired dimensionality of the latent space, and the resolution of the STFT (trading efficiency and sound quality)
- define the desired regularization, and some additional criteria to shape the latent space (semi-supervised learning, conditioning, geometric constraints)
- train the model on a defined number of epochs

This process allows for each user to develop heuristics on the training methods, that he will then validate though experimentation. The input parameters are the most relevant for the final shape of the representation, filtrating the parameters of less importance to avoid the over-parametrization of the tool. Thus, grounded by the theoretical analysis of section 1 and by the benchmark performed section 2, we

can design a set of heuristics to set the training parameters according to high-level features

- *narrowness*, that defines the capacity of the involved neural networks, trading between *bottleneck* and *sparse* modelling of the encoding / decoding fuctions,
- *coarseness*, that can trade between explicit (Kullback-Leibler, Renyi or Jensen-Shannon divergences) and implicit regularizations (MMD or adversarial), and tuning the β parameter, to implicitly define on the amount of mutual information between examples in the representation
- *influence*, setting the relative weight of the external criteria to the unsupervised loss.

These compliant high-level features can then allow non-expert users to intuition the underlying of these parameters, and still experiment with the system without being enforced to enter the details of the methods.

Interaction methods. The interface with Max software then opens the framework to online interaction, such that user can experiment in real-time with the previously trained model. The most straightforward method for latent navigation is to access separately every dimensions of the input space, with a set of sliders that it can perform during playing. However, the number of latent dimensions can be high, and then provide an over-parametrized interface that can make its use tedious, not helped by the non-linear behavior of the latent space. Hence, the user can rather navigate on sub-sets of the latent space, whether by dimension selection or with dimensionality reduction techniques such as PCA or ICA. The creative process is then leaded by the choice of these representations, that the user can save with the model.

As blind navigation in this space can be an unsatisfactory, additional visualization of the space can provide a useful feedback for bettering the performance. We can then provide a global representation of the currently explored subspace, annotated with training points to inform the user if he is exploring trained or untrained zone of the representation, and possibly with labelling information, provided its availability. The descriptor plots used previously can also provide precious information, allowing the user to target some zones with desired perceptual properties. If such visual feedback is provided, we can thus provide additional controllers to allow the user to navigate more freely into the space, as for example with a rotation knob that allows to rotate the explored representation. Alternatively, we can also rely on collaborative human-machine interaction techniques to explore the space over its full span, using reinforcement learning methods to reduce the number of parameters to a single reward. This was explored with *ægo*, that is presented below.

Offline methods. Finally, another way to use the framework is to consider it as a offline synthesizer, that can generate some sounds

based on some requests. The user then explores the possibility of the algorithm by experimenting the several generation methods provided by the flexibility of the framework. We summarize these possible offline interactions below :

- **trajectory generation** : using a *trajectory generator* to obtain a multi-dimensional curve in the latent space, and sample the corresponding audio distribution
- **sound morphing** : taking two or more target points, performing an interpolation in the latent space, and generate the corresponding audio distribution.

- **full sound morphing** : is the same than below, but with hierarchical temporal models, then performing "full sound" interpolation
- **cross-synthesis** : encoding and decoding an out-of-domain sound, and listen how it is "filtered" by the model 5.5
- **cross-modal translation** : if the model has a symbolic counterpart (see sec. 3.3), performed transfer between different domains
- **context translation** : if the model is conditioned by external information, regenerate a sound with an different context

- **auto-run** : if a model has a prediction model, let the prediction run over several time spans
- **prediction translation** : performs prediction of a model with a different temporal context (as a different context vector in CPC)

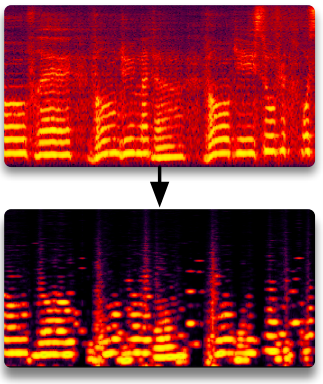


Figure 5.5: An example of cross-synthesis, with a sample of voice passed through a model trained on `toy_additive` dataset. We can see that the voice is somehow "recomposed" by the features learned by the model.

we see that the possibilities of generation with the proposed method are numerous, and are quite different from other synthesis techniques. We then hope that this framework could be found inspiring in the computer music community, and maybe can motivate other applications we did not suppose or being deviated from its original purpose.

5.2 A step towards research and creation process : *aego*

To experiment the creative framework proposed above we presented *ægo*, an improvisational piece with interactive sound and image for one performer. One of our intentions was to explore the representations extracted with AEVB by focusing on the emerging properties of its internal organization, without focusing on the technical aspects of the generating model but rather on the feedback loop created by the performer, the interactive agent, and the synthesis engine.
To represent this idea of exploration, the dimensionality of the navigated projection is increasing through time, making the growing complexity of the space perceptible, such that the audience is directly witnessing the progressive discovery of the space. In addition, we projected the generated sound spectra on stage to provide the audience with a visual representation that accentuate, not disrupt, the sonic perception of the piece, and emphasize the impression of movement.
Another intention was to display the *exploration behavior* of the reinforcement learning agent in front of the audience. To do this, we

wanted to challenge the skills and abilities usually at stake in performance, by summoning an ecological approach and evoking a sense of reciprocal interaction between the human and the machine. In this sense, rather than using it for control purposes, we used the body of the performer to convey kinesthetic information about how machine exploration may be internally experienced by a human. In parallel, we added raw textual information about the exploration agent's internal state at top left of the image projection to emphasize the machine's encoded perception of the performer.
Finally, this piece was created to make a first step towards a research & creation approach to the development of the proposed framework. Hence, we aimed to use it in a real-time performance, identifying first the technical & compositional challenges of using such models in creative applications, then to propose a musical realization of this method to an audience, allowing us to get some feedback about the global æsthetics emerging from the proposed system. While this "creativity-oriented" part of the evaluation would clearly demand a more extensive methodology, involving external artists and performers, and the development of well-defined usability studies, the creation of this piece was very beneficial to us, and helped us to focus more on these aspects in our respective works.

5.2.1 Reinforcement Learning for Sonic Exploration

Sonic exploration is a central task in music creation [383]. Specifically, exploration of digital sound synthesis consists in taking multiple steps and iterative actions through a large number of technical parameters to move from an initial idea to a final outcome. Yet, the mutually-dependent technical functions of parameters, as well as the exponential number of combinations, often hinder interaction with the underlying sound space. A way to support exploration of large sound synthesis spaces can be found in *reinforcement learning*, that defines a statistical framework for the interaction between a learning agent and its environment [27]. The agent can learn how to act in its environment by iteratively receiving some representation of the environment's state S, taking an action A on it, and receiving a numerical reward R. The agent's goal, roughly speaking, is to maximize the cumulative amount of reward that it will receive from its environment.

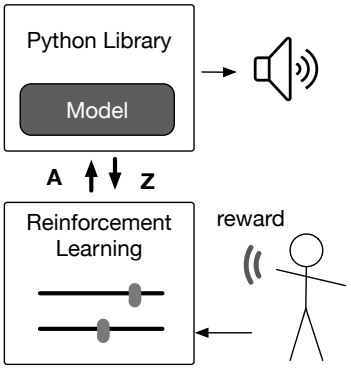


Figure 5.6: Reinforcement learning for sonic exploration. The agent learns which actions to take on a sound synthesis environment based on reward given by the musician. The agent implements an exploration method to foster discovery along interaction.

For our case of sonic exploration, we propose that the musician would listen to the agent exploring the space, and teach it how to explore by giving reward data (Fig. 5.6). Formally, the environment's state is constituted by the numerical values of all synthesis parameters. The agent's actions are to move one of the parameters up or down at constant frequency. Finally, the musician communicates *positive or negative reward* to the agent as a subjective feedback to agent actions. We implemented a deep reinforcement learning model to support learning from human reward signal in high-dimensional parametric spaces [384]. A crucial requirement for reinforcement learning agents is to *autonomously explore their environment*, to keep on discovering which actions would yield the most reward. We developed a statistical

method, based on intrinsic motivation, which pushes the agent to "explore what surprises it". The resulting interactive learning workflow was found to be useful to relax musicians' control over all synthesis parameters, while also provoking discoveries by exploring uncharted parts of the sound space.

5.2.2 Instrument Design

For this piece, we conceived a dedicated musical instrument combining latent-based generative models and reinforcement-based parameter exploration, and leverages their learning capabilities from a design perspective. This instrument is based both on software, interfacing the system presented in the first section and the parameter exploration framework proposed by Scurto & al. [385, 386], and on hardware, using a movement recognition device. The exploration agent then acts as an expressive partner [387], that adapts to the feedback given by the user and navigates in the high-dimensional variational space. Hence, the latent space can be seen as a creative interface, that allows the user to experiment with high the non-linearity of the representation topology.

Workflow. The interactive workflow of the instrument is then twofold, as shown in Fig. 5.7. The *setup* phase allows musicians to configure the instrument, create a customized sound dataset for the unsupervised learning model, experiment with various training parameters, or also load a previously-built latent sound space. They can also change dimensionality of the reinforcement learning agent to explore specific dimensions of the latent sound space, as well as the frequency at which it would take actions inside the latent space. During this setup phase, we followed the creative workflow developed in the first section. We thus trained several models on different datasets, varying the architecture, and tested them in our real-time implementation to select which one was the most satisfactory in terms of interaction. Then, the *playing* phase allows musicians to improvise with the agent by means of feedback. The agent produces a continuous layer of sound from the spectrum output of the VAE. Musicians can either cooperate with its learning by giving consistent feedback data to attain a sonic goal. Or, they can obstruct its learning by giving inconsistent feedback data to improvise through sonic exploration.

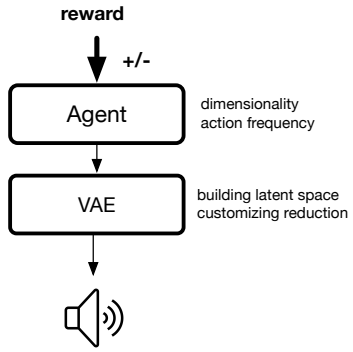


Figure 5.7: Interaction diagram of the *ægo* instrument. The reinforcement module monitors the reward given by the user and the current latent state to decide a future action A, that is used to control the navigation.

5.2.3 Engineering

Technically (see Fig. 5.7), the reinforcement learning agent receives a representation of the environment's state S as a position in the latent space $\mathbf{z}$. Then, it takes an action A corresponding to a displacement along some dimension of the latent space. The resulting position has the unsupervised learning model generate a sound spectrum $\mathbf{x}$. Based on the sound, the musician would communicate reward R to the agent. The latter would progressively learn to explore the latent space in

relation to the musician's feedback data.

Hardware. To support embodied musical interaction, we designed a hardware prototype that consists in two velcro rings, each of them equipped with a wireless inertial measurement unit*. We took each unit angular rotation about each forearm axis and summed them to compute a single, normalized numerical reward signal. This, combined with the lightweight, nonintrusive velcro rings, lets musicians experiment with a wide range of gesture vocabulary [388] to communicate positive or negative feedback to the agent.

Software. Both machine learning models are implemented in Python, based on `vsacids` and the `coexplorer`† library provided by Scurto & al. We developed a Max/MSP patch to implement a user interface for the setup phase, as well as a hardware data converter for the playing phase. The communication between the synthesis engine, the exploration agent and the audio renderer is leveraged with the OSC protocol.

5.2.4 Aesthetics and writing

Our artistic motivation for *ægo* was to open a sensitive reflection on what may actually be learned on a musical level through interaction with machine learning, both by the human and its artificial alter ego—the machine. To share this reflection with members of an audience, we opted for a performance format that displays a human and a machine mutually learning to interact with each other—on an embodied level for the human, and on a computational level for the machine—through live improvization.

The learning machine possesses a distinctive musical behavior, as well as two latent sound spaces, that are all originally unknown to the human performer. The latter will expressively negotiate control of these spaces with the machine, communicating positive or negative feedback using our instrument and its motion sensors placed in both hands. The slowly-evolving spectromorphologies, synthesized and projected in real-time on stage, create a contemplative, minimalist atmosphere intended to let members of the audience freely consider potential learnings of musical qualities by the human and the machine.

Composition. The piece is based on the exploration of two different latent spaces, the first obatined with the `toy_additive` and `toy_fm` datasets, built from synthetic sounds obtained with additive and FM synthesis, and the second with the `acidsInstrument-ordinario` dataset obtained with the sounds of classical instruments [281]. The representations were obtained using simple VAEs, using 3-layered convolutional networks, and have a total amount of 8 dimensions. The

* http://ismm.ircam.fr/riot/
† https://github.com/Ircam-RnD/coexplorer

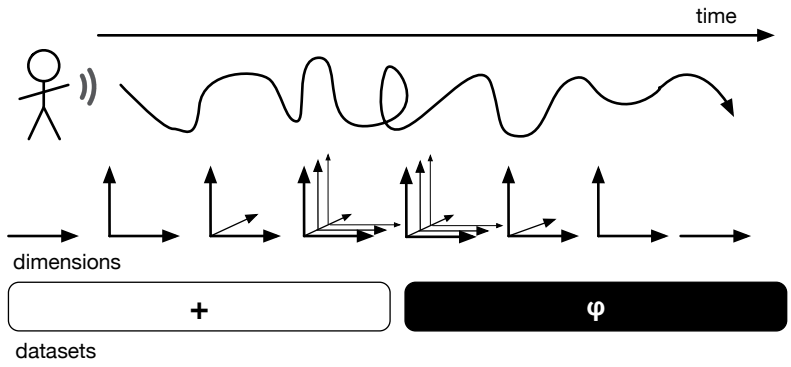


Figure 5.8: Temporal structure composed for the piece. The dimensions are progressively increased and then decreased when the dataset is changed.

performers explore a sub-plane to this latent space whose dimensions are increased through time (1, 2, 4 and 8), whether by the performer or by an external operator. We empirically found that ICA was more satisfactory than PCA (see section 2.2.1), as it progressively opens axis that are orthogonal to the previously explored ones (see fig. 5.8).

Once the full dimensionality of the additive latent space is reached, we change the representation, and reversely progressively shrink the number of projected dimensions to 1. This enables to write form within different soundscapes, allowing the building of a narrative from elementary sinusoidal spectra to richer instrumental timbres. The narration of the piece then distinguish two different phases. An *exploration* phase, that makes perceptible the exploration of an unknown space of increasing complexity, and a *decaying* phase, where the performer is suddenly projected in a richer space, whose complexity is progressively reduced to be fixed in one dimension. The performance is stopped when the performer manages to stay in a fixed point, concluding the performance. The exploration then consists in the improvisational paths taken by the reinforcement learning agent following the performer's feedback data. We set the frequency of agent actions between 30 and 100 milliseconds. This choice allowed for slow, continuous evolution of spectromorphologies, which enables to grasp the behaviour of the agent inside the latent spaces.

Performance. While the piece is intended to be improvised, our sole instruction toward the stage performer is that he or she globally performs with the machine with an overall sense of attentiveness*. We propose that the performer would start the piece facing the audience, relaxed, using the instrument with small forearm rotations only. As the piece would unfold over time, the performer would be free to adapt its gestures in response to the slowly evolving complexity of the explored spaces, focusing on embodied interaction with the machine. A second contributor is required to manage the two remaining temporal scales of the piece—*i.e.*, changing dimensionalities, and switching latent spaces.

* See the following video recording of a rehearsal: https://youtu.be/gCzOoNChlJQ

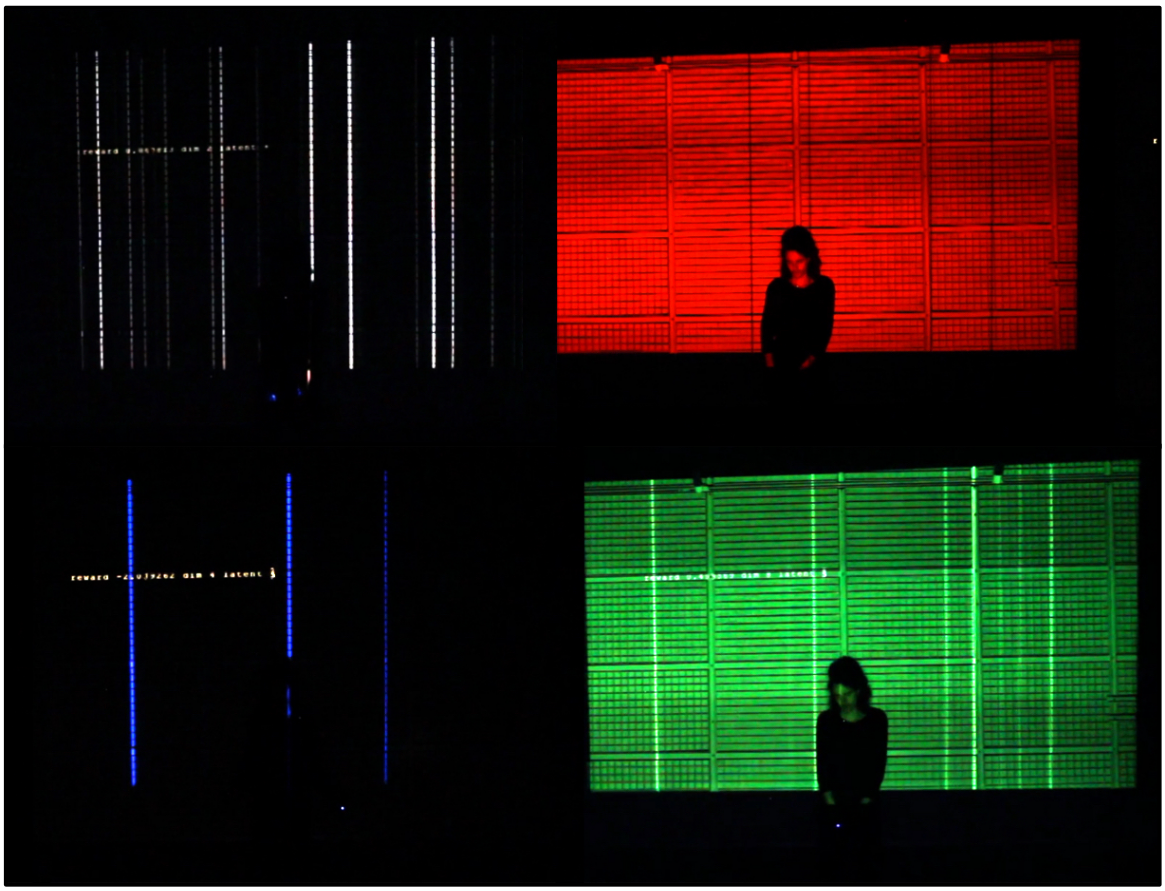


Figure 5.9: Pictures taken from the live setup of the piece, showing different moments of the performance.

Conclusion

The current work investigated the use of unsupervised / semi-supervised methods for audio synthesis and generation using *Auto-Encoding Variational Bayes*, a hybrid approach based on deep representation learning and Bayesian inference. The development of these methods thus allowed to jointly address the problem of *high-level feature extraction*, focusing on the automatic discovery of the underlying structure of a given dataset, and the problem of *generation*, inverting this representation to generate back the data in the audio domain. To this end, we leveraged expressive probability distribution modelers, based on neural networks, to model complex dependencies between the audio domain and the obtained representation, that we trained using black-box variational inference methods.
Starting from this ground, the work achieved in this PhD first targeted the development of latent regularization strategies specifically addressed to the audio domain, providing first steps of using methods derived from AEVB in the audio signal processing domain. First, we proposed a method to use this latent space as a *translation* space between audio and symbols, allowing to associate sounds and custom vocabularies in the latent representation. This regularization method then allowed us to perform in parallel symbol extraction and constrained audio generation, that we experienced by matching sounds from orchestral instruments with their corresponding pitches and dynamics. Then, we proposed a method to topologically enforce the latent space to reflect high-level audio properties, using dissimilarity distances obtained from perceptual measurements. This regularization method allowed us to propose a new approach of *timbre spaces*, a fundamental notion of audio perception, allowing the perceptual inference of sounds coming from instruments unknown to the system, and audio generation based on high-level properties, opening the way to *descriptor-based synthesis*.

Consequently, we addressed the temporality of audio signals, proposing to use this latent representation as expressive *state-space models*. We first modeled the dynamics of these audio signals by performing a prediction task in the latent space, that is trained jointly with overall model to intrinsically regularize its organization on temporal constraints. We provided three different prediction methods : a method based on *contrastive predictive coding*, that allows the extraction of *slow features* from the latent space, a method based on *normalizing flows*, allowing to model complex latent dynamics grounded on the idea of stochastic *filtrations*, and finally a method based on *Gaussian processes*, performing a Bayesian regression task that allows the continuous modeling of the latent trajectories. After showing the efficiency and respective properties of these prediction methods, we also proposed a Bayesian multi-scale architecture, `ShrubVAE`, that performs progressive downsampling of incoming data, then allowing to hierarchically

model the temporal structure of the learned sequence and to generate a signal up to custom temporal scales. The hierarchical refinement of the temporal information represented by the latent layers were shown to not reduce the reconstruction of the signals, opening the way to fast long-term analysis and generation of audio time series, that is still missing in the current state of the art.

Finally, we addressed the creative use of the developed methods by proposing a full audio synthesis framework, allowing off-line and on-line exploration of the extracted latent spaces, that could be hopefully used in music production or in real-time performances. Therefore, we provided first steps for a *research and creation* approach, by developing this new instrument and by the composition of a new musical piece, *ægo*, based on multi-dimensional latent space navigation using a reinforcement-learning exploratory agent. The creation of this piece allowed us to experience the musical use of these methods and to develop a specific creative workflow, that we implement in an framework usable for experts and non-expert users. Doing that, we addressed the *intrinsic* exploration of machine-learning generation methods, focusing on the structural behavior of the models rather than on their performance towards an explicit task, and allowing an experimental approach similar to digital audio synthesis practices.

Future works and perspectives. We understand this thesis as a preliminary work of using deep generative models both for high-level feature analysis, extracting representations reflecting underlying properties of the data that could be used for visualization or explicit tasks, and for an intrinsic creative use of these models, discovering their underlying properties to allow a fruitful creative interaction with a human user. We now that most of the works initiated in this PhD could be beneficially detailed and precised, and that the number of possible ameliorations and further investigations are numerous. We will here shortly describe the points that we find the most important, and that we target to develop in the future.

Phase and raw-waveform learning. The main issue that prevent the generation of involved models to be entirely satisfactory from a audio quality perspective is the use of magnitude spectra, that require phase reconstruction algorithms during inversion. This problem could be alleviated in two ways : whether also learning the phase of the spectrogram, resorting to complex neural networks or complex distributions, to prevent the use of extensive Griffin-Lim iterations, or directly model the raw waveform of the signal. While both solutions are non-trivial, the second seems to us be the best option, and is one of our main motivation for addressing the temporal aspect of AEVB methods.

Regularization strategies. The *regularization* aspect of these models is seminal, as it allows to enforce the latent properties of the model to be organized for a specific analytic or creative purpose. The two

regularization methods we proposed verified the flexibility of Auto-Encoding Variational Bayes to shape custom constraints, showing the flexibility of these methods. Hence, the development of additional regularization constraints addressing different motivations could allow further creative uses of this model, as for example to model high-level synthesis macro-parameters of existing synths (a parallel work was recently proposed by Esling & al., that also involves this work [389]), custom vocabularies for specific creative uses, explicit axis disentanglement of desired audio properties, and of course detailing the temporal regularizations that we initiated in this work.

Temporality. Due to time constraints, we could not address the modeling of long-time dependencies, that could allow to extend the approaches described above to significant temporal scopes. Furthermore, the efficient downsampling performed by the proposed `ShrubVAE` could be investigated more intensively, as the obtained generation results are surprisingly good regarding the temporal scope of the top latent representation (approximatively between one and two seconds per latent vector). Even the simple use of ShrubVAE developed in this thesis multiplies by three times the temporal scope of algorithms such as WaveNet, while using only two layers and involving neural networks of reasonable shapes. Also, the developed multi-scale approach to time series modeling could be used to perform constrained generation on multiple temporal scales, and thus be able to efficiently perform analysis / generation based on high-level musical features (several seconds). With such systems, we can then hope to be able to generate at the raw waveform level, hopefully in real time, then preventing to use spectral transforms and allowing us to extract dynamical features from audio signals.

Usabiliity. Despite being only approached in the last section of this manuscript, the development of a *research & creation* method, that is a flourishing dialog between scientific topics and musical creation, seems to us mandatory as we aim to develop creative tools from the studied models. Therefore, further developments should also investigate more deeply the usability of the proposed framework, involving the collaboration with external artists, composers, and performers, as well as well-defined usability studies that could opening this work to design and human-computer interaction. Also, the integration of alternative media (image, video) and *movement* in the process was also very beneficial, allowing to provide a physicality to the system that is also mandatory for live performance purposes. Finally, the parallel development of a framework usable in dedicated software for music production could also spread this new method onto musical practices, and could in turn point out specific scientific, design and musical questions on the use of machine-learning based generation processes for creativity.

List of publications.

- [1] Jérôme Nika, Ken Déguernel, Axel Chemla–Romeu-Santos, Emmanuel Vincent, Gérard Assayag. *DYCI2 agents: merging the" free"," reactive", and" scenario-based" music generation paradigms*. hal.archives-ouvertes.fr, 2017.

- [2] Philippe Esling, Axel Chemla–Romeu-Santos, Adrien Bitton. *Bridging Audio Analysis, Perception and Synthesis with Perceptually-regularized Variational Timbre Spaces*. ISMIR, 2018.

- [3] Philippe Esling, Axel Chemla–Romeu-Santos, Adrien Bitton. *Generative timbre spaces: regularizing variational auto-encoders with perceptual metrics*. DaFX, 2018. https://arxiv.org/abs/1805.08501

- [4] Adrien Bitton, Philippe Esling, Axel Chemla–Romeu-Santos. *Modulated Variational auto-Encoders for many-to-many musical timbre transfer.* ISMIR, 2018. https://arxiv.org/abs/1810.00222

- [5] Philippe Esling, Naotaké Masuda, Adrien Bardet, Romeéo Despres, Axel Chemla–Romeu-Santos. *Universal audio synthesizer control with normalizing flows*. ISMIR, 2019. https://arxiv.org/abs/1907.00971

- [6] Axel Chemla–Romeu-Santos, Stavros Ntalampiras, Philippe Esling, Goffredo Haus, Gérard Assayag. *Cross-Modal Variational Inference for Bijective Signal-Symbol Translation.* DAFX, 2019.

- [7] Axel Chemla–Romeu-Santos, Hugo Scurto. *Machine Learning for Computer Music Multidisciplinary Research: A Practical Case Study.* CMMR, 2019.

- [8] Axel Chemla–Romeu-Santos, Hugo Scurto. *ægo.* CMMR, 2019.

List of acronyms

AE	Auto-Encoder 38
AEVB	Auto-Encoding Variational Bayes
ALI	Adversarially Learned Inference
ANN	Artificial Neural Network
AR	Auto-Regressive
ARMA	Auto-Regressive Moving Average
CAE	Contractive Auto-Encoder
CPC	Contrastive Predictive Coding
DAE	Denoising Auto-Ecoder
DFT	Discrete Fourier Transform
DNN	Deep Neural Networks
DRE	Density Ratio Estimation
DSP	Digital Signal Processing
ELBO	Evidence Lower-BOund
EM	Expectation-Maximization
GAN	Generative Adversarial Networks
GP	Gaussian Processes
HMM	Hidden Markov Models
MA	Moving Average
MIR	Music Information Retrieval
MLP	Multi-Layer Perceptron
MMD	Maximum Mean Discrepency
NCI	Noise Constrastive Estimation
NF	Normalizing Flows
NN	Neural Networks (same as ANN)
OT	Optimal Transport
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SSM	State-Space Models
STFT	Short-Term Fourier Transform
SVD	Singluar Value Decomposition
SVI	Stochastic Variational Inference
VAE	Variational Auto-Encoder
VI	Variational Inference

Figures and tables for baseline results

A

In this appendix we show all the complete results tables for the preliminary results of section 2. The number in parenthesis are obtained with the test set, while the results in regular characters are obtained with train tests.

A.1 Benchmark reconstruction results.

A.1.1 Dimensions benchmark

A.1.1.1 `toy_additive` table

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^2	SC	ISD	log-ℓ^1
2d					
VAE	910.64 (*910.10*)	340.71 (*340.40*)	92.75 (*90.57*)	3.435e-4 (*1.27e-3*)	217.21 (*216.93*))
PCA	-	368.17	111.68	nan	253.27
ICA	-	368.17	111.68	nan	253.27
4d					
VAE	550.13 (*551.74*)	49.25 (*50.14*)	31.99 (*32.02*)	3.78e-4 (*1.38e-4*)	85.25 (*86.61*)
PCA	-	338.72	117.94	nan	251.66
ICA	-	338.72	117.83	nan	251.66
8d					
VAE	510.78 (*510.56*)	25.52 (*25.45*)	13.60 (*14.03*)	**3.16e-4** (***1.18e-3***)	64.82 (*64.46*)
PCA	-	289.12	127.40	nan	245.70
ICA	-	289.12	127.34	nan	245.71
16d					
VAE	**488.37** (***488.37***)	**13.33** (***13.21***)	**11.26** (***11.07***)	1.08e-3 (*2.92e-4*)	**47.79** (***48.20***)
PCA	-	137.29	160.42	nan	139.35
ICA	-	137.29	160.42	nan	139.35

A.1.1.2 `toy_fm` table

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^2	SC	ISD	log-ℓ^1
2d					
VAE	1816.64 (*1820.59*)	1070.21 (*1073.57*)	347.40 (*349.29*)	3.72e-4 (*1.40e-3*)	617.31 (*619.38*)
PCA	-	1087.64	469.06955	nan	650.80
ICA	-	1087.64	469.1	nan	650.80
4d					
VAE	1996.2 (*1989.7*)	1187.8 (*1179.9*)	2003.321 (*2036.61*)	4.18e-4 (*1.57e-3*)	837.52 (*833.86*)
PCA	-	1039.62	469.07	nan	641.61
ICA	-	1087.64	469.1	nan	650.8
8d					
VAE	1003.5 (*994.03*)	427.81 (*419.91*)	54.24 (*54.40*)	4.32e-4 (*1.62e-3*)	289.95 (*285.81*)
PCA	-	1039.62	453.68	nan	641.61
ICA	-	950.35	453.7558	nan	650.8
16d					
VAE	**820.670 (*825.75*)**	**258.13 (*262.16*)**	**15.59 (*15.817*)**	**3.70e-4 (*1.37e-3*)**	**190.71 (*193.48*)**
PCA	-	859.12	527.28	nan	608.13
ICA	-	859.00	527.74	nan	608.13

A.1.1.3 `acidsInstruments-ordinario` table

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^2	SC	ISD	log-ℓ^1
2d					
VAE	334.41 (*334.34*)	68.12 (*68.12*)	21.72 (*21.5*)	8.53e-5 (*3.15e-4*)	84.14 (*84.03*)
PCA	-	104.76	16.47	nan	107.22
ICA	-	104	16.48	nan	107.22
4d					
VAE	286.55 (*287.0*)	31.90 (*31.93*)	17.45 (*18.21*)	8.53e-5 (*3.15e-4*)	66.27 (*66.17*)
PCA	-	92.7	17.02	nan	104.82
ICA	-	92.7	17.03	nan	104.82
8d					
VAE	270.1 (*269.9*)	20.23 (*20.11*)	12.10 (*12.28*)	8.30e-5 (*3.06e-4*)	51.56 (*51.28*)
PCA	-	71.54	16.79	nan	98.40
ICA	-	71.54	16.79	nan	98.40
16d					
VAE	276.56 (*276.69*)	23.49 (*23.67*)	22.96 (*20.13*)	1.31e-3 (*4.79e-3*)	56.74 (*56.72*)
PCA	-	48.65	15.16	nan	219.70
PCA	-	48.65	15.16	nan	85.34

A.1.2 Divergences benchmark

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^2	SC	ISD	log-ℓ^1
`toy_additive_mini`					
D_{KL}	488.37 (*488.37*)	13.33 (*13.21*)	11.26 (*11.07*)	1.08e-3 **(*2.92e-4*)**	47.79 (*48.20*)
MMD	**474.46 (*474.75*)**	**5.90 (6.06)**	**6.78 (*6.67*)**	3.12e-4 (*1.16e-3*)	**33.66 (*34.39*)**
$D_{\alpha=2}$	487.89 (*487.84*)	13.035 (*13.01*)	9.03 (*9.03*)	2.95e-4 (*1.09e-3*)	46.75 (*46.72*)
JSD	519.41 (*519.1379*)	28.07 (*24.36*)	24.34 (*7.60*)	3.22e-4 (*1.20e-3*)	70.70 (*70.36*)
`Adv`	485.78 (*485.88*)	11.69(*10.18*)	10.25(*10.0*)	**2.91e-4** (*1.075e-3*)	46.44 (*46.49*)
PCA	-	137.29	160.42	nan	139.35
ICA	-	137.29	160.42	nan	139.35
`toy_fm`					
D_{KL}	910.64 (*910.10*)	340.71 (*340.40*)	15.59 (*15.81*)	3.435e-4 (*1.27e-3*)	**190.71 (*193.48*)**
MMD	**771.6 (*766.8*)**	**225.08 (*221.78*)**	62.14 (*62.19*)	4.00e-4 (*1.50e-3*)	205.46 (*202.95*)
$D_{\alpha=2}$					
JSD	792.52 (*259.07*)	244.99 (*242.90*)	27.83 (*27.60*)	**8.28e-05 (*3.06e-4*)**	199.34 (*0.63*)
`Adv`	1273.90 (*789.56*)	524.03 (*525.16*)	**6.28 (*6.46*)**	4.24e-4 (*1.59e-3*)	253.13 (*197.16*)
PCA	-	859.12	527.28	nan	608.13
ICA	-	859.95	527.28	nan	608.13
`aIns-o`					
D_{KL}	276.56 (*276.69*)	23.49 (*23.67*)	22.96 (*20.13*)	1.31e-3 (*4.79e-3*)	56.74 (*56.72*)
MMD	248.19 (*248.19*)	8.43 (*8.43*)	5.25 (*5.31*)	8.25e-05 (*3.04e-4*)	35.37 (*35.26*)
$D_{\alpha=2}$	255.66 (*255.74*)	12.33 (*12.38*)	10.23 (*10.53*)	8.27e-05 (*3.05e-4*)	43.39 (*43.40*)
JSD	259.36 (*259.07*)	7.75 (*14.22*)	7.74 (*7.60*)	8.28e-05 (*3.06e-4*)	43.19 (*43.07*)
`Adv`	**246.8 (*246.7*)**	7.76 (*7.71*)	9.97 (*10.0*)	8.26e-5 (*000*)	35.83 (*35.83*)
PCA	-	48.65	15.16	nan	219.70
ICA	-	48.65	15.16	nan	85.34

A.1.3 β benchmarks

	$-\log p(\mathbf{x}\|\mathbf{z})$	ℓ^2	SC	ISD	log-ℓ^1
`toy_additive`					
$\beta = 1$	**276.56 (*276.69*)**	**23.49 (*23.67*)**	22.96 (*20.13*)	1.31e-3 (*4.79e-3*)	56.74 (*56.72*)
$\beta = 4$	518.45 (*518.12*)	31.57 (*31.43*)	**19.33 (*19.46*)**	3.11e-4 (*1.15e-3*)	68.58 (*68.03*)
$\beta = 10$	554.01(*553.48*)	58.18 (*57.91*)	37.34 (*37.85*)	3.09e-4 (*1.14e-3*)	89.51 (*88.79*)
PCA	-	137.29	160.42	nan	139.35
ICA	-	137.29	160.42	nan	139.35
`toy_fm`					
$\beta = 1$	**820.670 (*825.75*)**	**258.13 (*262.16*)**	**15.59 (*15.817*)**	**3.70e-4 (*1.37e-3*)**	**190.71 (*193.48*)**
$\beta = 4$	901.00 (*897.3193*)	319.03 (*316.24*)	17.094276 (*15.61*)	3.96e-4 (*1.47e-3*)	211.48 (*208.22*)
$\beta = 10$	1098.0 (*1099.0*)	451.97 (*453.92*)	16.15 (*15.97*)	4.24e-4 (*1.59e-3*)	248.34 (*250.49*)
PCA	-	859.12	527.28	nan	608.13
ICA	-	859.95	527.28	nan	608.13
`aIns-o`					
$\beta = 1$	261.37 (*261.37*)	15.24 (*15.34*)	14.384189 (*13.74*)	8.294699e-5 (*3.06e-4*)	56.74 (*47.81*)
$\beta = 4$	279.98 (*279.83688*)	27.82 (*27.77*)	17.26 (*16.56*)	8.30e-05 (*3.07e-4*)	56.96 (*56.77*)
$\beta = 10$	303.88 (*304.09*)	45.06 (*45.21*)	41.46 (*44.54*)	8.3674e-05 (*3.09e-04*)	72.98 (*73.11*)
PCA	-	859.12	527.28	nan	608.13
ICA	-	859.12	527.28	nan	608.13

A.2 Regularization benchmarks.

A.2.1 Dimensions benchmark

A.2.1.1 `toy_additive`

	$D_{KL}[q\|p]$	$D_{KL}[p\|q]$	$D_2^{\alpha}[q\|p]$	$JSD_{0.5}[p\|q]$	$MMD[p,q]$	$TC[p,q]$
2d						
VAE	6.78 (*6.78*)	745.24 (*744.37*)	7.80 (*7.80*)	184.83 (*184.61*)	5287.61 (*1338.22*)	8.29 (*7.56*)
PCA	-	-	-	-	48961	32.85
ICA	-	-	-	-	20020	1.65e-4
4d						
VAE	13.87(*13.89*)	44620.4 (*4449.24*)	15.87 (*15.89*)	179.54 (*1109.28*)	21510 (*125.44*)	11.72 (*10.78*)
PCA	-	-	-	-	22944	56.711
ICA	-	-	-	-	4587.23	1.0826e-4
8d						
VAE	20.64 (*20.64*)	1658.67 (*1647.13*)	24.63 (*24.62*)	410.29 (*407.40*)	840.89(*212.56*)	19.00 (*18.83*)
PCA	-	-	-	-	141890	118.63
ICA	-	-	-	-	13725	2.66e-4
16d						
VAE	35.15 (*35.20*)	438.08 (*440.41*)	49.45 (*49.50*)	103.48 (*104.05*)	673.49 (*159.52*)	22.75 (*22.89*)
PCA	-	-	-	-	759.39	400.13
ICA	-	-	-	-	34.193	4e-2

A.2.1.2 `toy_fm`

	$D_{KL}[q\|\|p]$	$D_{KL}[p\|\|q]$	$D_2^{\alpha}[q\|\|p]$	$JSD_{0.5}[p\|\|q]$	$MMD[p,q]$	$TC[p,q]$
2d						
VAE	6.31 (*6.31*)	612.88 (*629.59*)	7.32 (*7.32*)	151.85 (*156.02*)	3648.16 (*869.68*)	29.82 (*23.33*)
PCA	-	-	-	-	12620	19.62
ICA	-	-	-	-	4209.7	1.04e-3
4d						
VAE	15.47 (*15.46*)	11962 (*11883*)	17.47 (*17.46*)	2987 (*2967*)	2987 *1470*)	30.53 (*31.69*)
PCA	-	-	-	-	43748	158.63
ICA	-	-	-	-	2871.0	1.65e-3
8d						
VAE	28.44 (*28.46*)	9854.3 (*9735.94*)	32.44 (*32.46*)	2457.2 (*2427.7*)	5010.1 (*1252.14*)	40.31 (*43.71*)
PCA	-	-	-	-	102061	86.44
ICA	-	-	-	-	15727	3.22e-3
16d						
VAE	48.98 (*49.20*)	8632 (*8750.4*)	56.98 (*57.20*)	2147.54 (*2176*)	10081 (*2406*)	71.43 (*65.32*)
PCA	-	-	-	-	207162	406.33
ICA	-				7628	1.76e-3

A.2.1.3 `acidsInstruments-ordinario`

	$D_{KL}[q\|\|p]$	$D_{KL}[p\|\|q]$	$D_2^{\alpha}[q\|\|p]$	$JSD_{0.5}[p\|\|q]$	$MMD[p,q]$	$TC[p,q]$
2d						
VAE	5.97 (*5.97*)	1042 (*1042*)	6.97 (*6.98*)	259.2 (*259.2*)	16054 (*4025*)	25.68 (*24.46*)
PCA	-	104.76	16.47	nan	nan	107.42
ICA	-	104	16.48	nan	nan	107.42
4d						
VAE	-694.1 (*9.84*)	726.39 (*000*)	11.83 (*11.81*)	179.54 (*180.04*)	110.60 (*5549*)	57.23 (*56.79*)
PCA	-	-	-	nan	161016	148.2
ICA	-	-	-	-	65115	6.12e-5
8d						
VAE	13.88 (*13.9*)	353.27 (*351.9*)	17.734 (*17.72*)	85.62 (*85.29*)	26389 (*6529*)	81.51 (*84.46*))
PCA	-	-	-	nan	188622	148.2
ICA	-	-	-	-	48852	1e-4
16d						
VAE	-694.1 (*000*)	0.093 (*000*)	-416.6 (*000*)	0.177 (*000*)	0.177 (*000*)	0.177 (*000*)
PCA	-	-	-	-	39409	204.82
ICA	-	48.68	15.19		7881	3.9e-4

A.2.2 Divergence regularization results

	$D_{KL}[q\|p]$	$D_{KL}[p\|q]$	$D_2^\alpha[q\|p]$	$JSD_{0.5}[p\|q]$	$MMD[p,q]$	$TC[p,q]$
toy_additive						
D_{KL}	**35.15 (*35.20*)**	**438.08 (*440.41*)**	**49.45 (*49.50*)**	**103.48** (*104.05*)	**673.49 (*159.52*)**	22.75 (*22.89*)
MMD	279.43 (*279.86*)	22.49e11 (*86.10e9*)	295.43 (*295.86*)	16.46e9 (*521.5e9*)	729.60 (*588.83*)	**20.13 (*22.67*)**
$D_{\alpha=2}$	35.40(*35.42*)	456.00 (*458.25*)	49.77 (*49.78*)	107.87 (*108.42*)	689.97 (*174.62*)	21.82 (*22.25*)
JSD	341.51 (*340.71*)	13.67e6 (*13.22e6*)	357.52 (*356.72*)	3418281 (*3.30e6*)	43200 (*3.30e6*)	206.66 (*205.71*)
Adv	947.45 (*948.73*)	3.70e32 (*3.53e32*)	955.44 (*956.72*)	9.24e31 (*8.81e31*)	470097 (*117231*)	213.88 (*211.75*)
PCA	-	-	-	-	759.39	400.13
ICA	-	-	-	-	34.193359375	4e-2
toy_fm						
D_{KL}	**48.98 (*49.20*)**	8632 (*8750.4*)	56.98 (*57.20*)	2147.54 (*2176*)	10081 (*2406*)	71.43 (*65.32*)
MMD	359.16 (*359.16*)	3.91e28 (*3.87e28*)	367.16 (*295.86*)	9.77e27 (*9.67e27*)	**1659.7 (*405.44*)**	**36.33 (*35.98*)**
$D_{\alpha=2}$	210.4 (*210.86*)	3.31e16 (*3.40e16*)	218.40 (*218.86*)	8.26e15 (*8.51e15*)	63233 (*15523*)	157.91 (*155.37*)
JSD	153.88 (*153.00*)	**277.95 (*276.26*)**	243.02 (*241.64*)	61.25 (*60.88*)	178043 (*43708*)	200.24 (*197.73*)
Adv	2178.50 (*2179.06*)	4.55e29 (*4.51e29*)	2186.50 (*2187.07*)	1.14e29 (*1.13e+29*)	585633 (*143165*)	411.74 (*409.4*)
PCA	-	-	-	-	207162	406.33
ICA	-	48.68	15.19		7628	1.76e-3
acidsInstruments-o						
D_{KL}	18.69 (*18.82*)	247.97(*252.12*)	25.65 (*25.80*)	58.63 (*59.63*)	155.60 (*35.72*)	25.35 (*23.36*)
MMD	141.91 (*141.70*)	22.49e11 (*26.39e12*)	149.92 (*149.70*)	5.622e11 (*65.98e12*)	2399.50 (*588.83*)	20.13 (*19.35*)
$D_{\alpha=2}$	18.12 (*18.08*)	398.89 (*399.56*)	24.41 (*24.35*)	96.35 (*96.52*)	4124.59 (*1012.24*)	27.83 (*28.28*)
JSD	32.92 (*32.84*)	49.64 (*49.53*)	57.69 (*57.55*)	11.144 (*11.12*)	105911 (*26344*)	130.77 (*130.11*)
Adv	1302.11 (*1303.22*)	1e+31 (*1.75+31*)	1310.11 (*1311.22*)	4.7e30 (*4.37e30*)	1.88e6 (*000*)	542.7 (*535*)
PCA	-	-	-	-	39409	204.82
ICA	-	48.68	15.19		7881	3.9e-4

A.2.3 β-benchmarks.

	$D_{KL}[q\|p]$	$D_{KL}[p\|q]$	$D_2^\alpha[q\|p]$	$JSD_{0.5}[p\|q]$	$MMD[p,q]$	$TC[p,q]$
toy_additive						
$\beta = 1$	35.15 (*35.20*)	438.08 (*440.41*)	49.45 (*49.50*)	103.48 (*104.05*)	673.49 (*159.52*)	22.75 (*22.89*)
$\beta = 4$	18.30 (*18.28*)	155.44 (*155.22*)	25.72 (*25.71*)	35.75 (*35.70*)	605.64(*143.10*)	20.34 (*19.83*)
$\beta = 10$	12.48 (*12.47*)	60.37 (*60.04*)	19.30 (*19.29*)	13.30 (*13.22*)	550.72 (*140.6*)	19.93 (*20.10*)
PCA	-	-	-	-	759.39	202.13
ICA	-	-	-	-	8875.41	6.77e-4
toy_fm						
$\beta = 1$	34.94 (*49.20*)	8632 (*8750*)	56.97 (*57.20*)	2147 (*2176*)	10081 (***2406***)	**71.43 (*65.33*)**
$\beta = 4$	34.94 (*34.77*)	1791.2 (*1736.4*)	43.06 (*42.9*)	440.84 (*427.29*)	10163 (*2534*)	85.73 (*89.76*)
$\beta = 10$	**24.86 (*24.87*)**	**500.77 (*504.15*)**	**33.09 (*33.09*)**	**121.5 (*121.8*)**	**10053** (*2419.8*)	108.52 (*113.10*)
PCA	-	-	-	-	205598	405.70
ICA	-	-	-	-	7594.93	6.10e-4
acidsInstruments-o						
D_{KL}	19.19 (*19.28*)	282.62 (*284.09*)	26.29 (*26.39*)	67.20 (*67.54*)	11557 (*2840.25*)	51.95 (*50.27*)
$\beta = 4$	9.36 (*9.36*)	57.24 (*57.50*)	14.67 (*14.66*)	13.00 (*12.98*)	10473 (*2527.10*)	51.63 (*50.80*)
$\beta = 10$	5.424(*5.423*)	26.63 (*26.67*)	8.78 (*8.78*)	5.90 (*5.912*)	7539.1 (*1807.7*)	45.781 (*45.52*)
PCA	-	-	-	-	39409	204.82
ICA	-	-	-	-	7881.27	3.92e-4

A.3 Disentangling.

A.3.1 `toy_additive`

	f_0	random	**n_partials**	random	**decay**	random
2d	4.90	4.97	2.29	2.26	1.75	1.75
4d	4.79	4.87	2.10	2.15	**1.69**	1.68
8d	**4.75**	4.81	**2.011**	2.09	1.70	1.71
D_{KL}	4.97	4.79	2.22	2.12	1.73	**1.72**
MMD	**4.77**	4.82	2.14	2.13	2.50	2.63
$D_{\alpha=2}$	4.79	4.78	2.17	**2.12**	**1.72**	**1.72**
JSD	6.99	7.13	3.63	3.49	2.91	2.91
Adv	12.02	7.52	6.97	6.38	5.31	5.32
$\beta = 4$	4.86	4.78	2.10	2.09	1.72	1.70
$\beta = 10$	4.73	4.78	2.06	2.09	1.69	1.70
PCA	5.55	7.13	3.94	4.02	2.51	2.51
ICA	4.61	4.61	1.95	1.95	1.61	1.61

A.3.2 `toy_fm`

	f_carrier	random	**f_multiplier**	random	**fm_ratio**	random
2d	3.54	3.54	3.93	3.90	3.44	3.48
4d	3.31	3.31	3.60	3.63	3.23	3.25
8d	3.49	3.49	3.77	3.76	3.42	3.46
D_{KL}	3.58	3.55	3.82	3.85	3.54	3.58
MMD	3.46	3.43	3.78	3.71	3.29	3.37
$D_{\alpha=2}$						
JSD	5.68	5.60	5.68	6.00	5.97	5.50
Adv	14.32	14.34	16.12	16.40	14.44	15.49
$\beta = 4$	3.62	3.52	3.82	3.82	3.33	3.46
$\beta = 10$	3.43	3.48	3.90	3.81	3.61	3.48
PCA	6.40	6.04	5.81	7.02	4.84	6.34
ICA	**3.22**	3.22	**3.54**	3.53	**3.17**	3.18

A.3.3 acidsInstruments-ordinario

	octave	random	**pitch**	random	**dynamics**	random	**instrument**	random
2d	2.52	2.46	2.72	2.74	1.73	1.69	2.74	2.72
4d	2.55	2.42	2.41	2.67	1.77	1.98	2.79	2.76
8d	2.51	2.47	2.60	2.65	1.70	1.66	2.75	2.65
D_{KL}	2.77	2.47	2.76	2.66	1.70	1.72	2.46	2.69
MMD	2.4	2.35	2.50	2.64	2.50	2.63	1.68	1.75
$D_{\alpha=2}$	2.27	2.33	2.61	2.60	1.74	1.72	2.73	2.61
JSD	2.733	2.6653	3.02	2.97	1.97	2.06	3.12	3.07
Adv	6.94	7.52	10.112	10.03	6.08	7.42	8.74	8.25
$\beta = 4$	2.64	2.45	2.75	2.62	1.80	1.73	2.38	2.61
$\beta = 10$	2.33	2.33	2.65	2.58	1.65	1.68	2.78	2.57
PCA	3.00	3.04	3.35	3.39	2.04	2.04	3.27	3.19
ICA	2.41	2.265	2.53	2.53	2.53	1.65	1.61	2.53

What doth life ?
- Xavier Renegade Angel

Bibliography

[1] Max V Mathews. 'The digital computer as a musical instrument'. In: *Science* 142.3592 (1963), pp. 553–557 (cited on page 2).

[2] Jean-Claude Risset and David L Wessel. 'Exploration of timbre by analysis and synthesis'. In: *The psychology of music* 2 (1982), p. 151 (cited on pages 2, 88).

[3] John M Chowning. 'The synthesis of complex audio spectra by means of frequency modulation'. In: *Journal of the audio engineering society* 21.7 (1973), pp. 526–534 (cited on pages 2, 58).

[4] Noam Chomsky. *Aspects of the Theory of Syntax*. Tech. rep. MASSACHUSETTS INST OF TECH CAMBRIDGE RESEARCH LAB OF ELECTRONICS, 1964 (cited on page 2).

[5] Richard Ohmann. 'Generative grammars and the concept of literary style'. In: *Word* 20.3 (1964), pp. 423–439 (cited on page 2).

[6] Grzegorz Rozenberg and Arto Salomaa. *The mathematical theory of L systems*. Vol. 90. Academic press, 1980 (cited on page 2).

[7] Robert M Haralick, Stanley R Sternberg, and Xinhua Zhuang. 'Image analysis using mathematical morphology'. In: *IEEE transactions on pattern analysis and machine intelligence* 4 (1987), pp. 532–550 (cited on page 2).

[8] Martin Norgaard. 'How Jazz Musicians Improvise'. In: *Music Perception: An Interdisciplinary Journal* 31.3 (2014), pp. 271–287. DOI: 10.1525/mp.2014.31.3.271 (cited on page 2).

[9] Mark J Steedman. 'A generative grammar for jazz chord sequences'. In: *Music Perception: An Interdisciplinary Journal* 2.1 (1984), pp. 52–77 (cited on page 2).

[10] E.R. Miranda. 'Evolving cellular automata music: From sound synthesis to composition'. In: *Proceedings of the Workshop on Artificial Life Models for Musical Applications* (2001), p. 12 (cited on page 2).

[11] Dave Burraston et al. 'Cellular Automata in MIDI based Computer Music'. In: *Proceedings of the International Computer Music Conference* 4 (2004), pp. 71–78 (cited on page 2).

[12] Fleischer Manufacturing. '(12) United States Patent'. In: 1.12 (2002) (cited on page 2).

[13] Adriano Baratè, Goffredo Haus, and Luca A Ludovico. 'Music analysis and modeling through Petri nets'. In: *International Symposium on Computer Music Modeling and Retrieval*. Springer. 2005, pp. 201–218 (cited on page 2).

[14] C. E. Shannon and W. Weaver. 'The Mathematical Theory of Communication'. In: *The mathematical theory of communication* 27.4 (1949), p. 117. DOI: 10.2307/3611062 (cited on pages 2, 23).

[15] Luciano Floridi. 'What is the Philosophy of Information ?' In: (1997) (cited on page 3).

[16] Antti Honkela and Harri Valpola. 'Variational learning and bits-back coding: an information-theoretic view to Bayesian learning'. In: *IEEE Transactions on Neural Networks* 15.4 (2004), pp. 800–810 (cited on page 3).

[17] Jie Cheng et al. 'Learning Bayesian networks from data: an information-theory based approach'. In: *Artificial intelligence* 137.1-2 (2002), pp. 43–90 (cited on page 3).

[18] Jürgen Schmidhuber. 'Formal theory of creativity, fun, and intrinsic motivation (1990–2010)'. In: *IEEE Transactions on Autonomous Mental Development* 2.3 (2010), pp. 230–247 (cited on pages 3, 4).

[19] Peter D. Grünwald. *The Minimum Description Length Principle*. 2007, p. 736 (cited on pages 3, 27).

[20] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. 'Testing the manifold hypothesis'. In: *Journal of the American Mathematical Society* 29.4 (2016), pp. 983–1049. DOI: 10.1090/jams/852 (cited on pages 3, 29).

[21] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 'Representation Learning: A Review and New Perspectives'. In: 1993 (2012), pp. 1–30. DOI: 10.1145/1756006.1756025 (cited on page 4).

[22] Diederik P Kingma and Max Welling. 'Auto-encoding variational bayes'. In: *arXiv preprint arXiv:1312.6114* (2013) (cited on pages 4, 42, 43, 47, 127).

[23] Ian Goodfellow et al. 'Generative adversarial nets'. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680 (cited on pages 4, 59).

[24] Diederik P. Kingma and Prafulla Dhariwal. 'Glow: Generative Flow with Invertible 1x1 Convolutions'. In: 2 (2018), pp. 1–10 (cited on page 4).

[25] Margaret A Boden. 'of Creativity'. In: (2009), pp. 23–34 (cited on page 4).

[26] Philippe Pasquier et al. 'An introduction to musical metacreation'. In: *Computers in Entertainment (CIE)* 14.2 (2016), p. 2 (cited on page 4).

[27] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cited on pages 4, 138).

[28] Simon Colton, John Charnley, and Alison Pease. 'Computational Creativity Theory : The FACE and IDEA Descriptive Models'. In: (1992), pp. 90–95 (cited on page 4).

[29] Jesse Engel et al. 'Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders'. In: (2017) (cited on page 5).

[30] Adam Roberts et al. 'A hierarchical latent vector model for learning long-term structure in music'. In: *arXiv preprint arXiv:1803.05428* (2018) (cited on page 5).

[31] Diederik P. Kingma et al. 'Semi-Supervised Learning with Deep Generative Models'. In: (2014), pp. 1–9 (cited on page 9).

[32] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 'Stochastic Backpropagation and Approximate Inference in Deep Generative Models'. In: (2014). DOI: 10.1051/0004-6361/201527329 (cited on pages 9, 44, 51).

[33] Ronald Aylmer Fisher. *Statistical methods for research workers*. Genesis Publishing Pvt Ltd, 2006 (cited on page 11).

[34] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer. 2006 (cited on page 13).

[35] G Cooper. 'Computational Complexity of probabilistic inference using Bayesian belief networks (research note)'. In: *Machine Learning* 42.1990 (1990), pp. 393–405 (cited on page 14).

[36] Judea Pearl. 'Fusion, propagation, and structuring in belief networks'. In: *Artificial Intelligence* 29.3 (1986), pp. 241–288. DOI: 10.1016/0004-3702(86)90072-X (cited on page 14).

[37] Andriy Mnih and Karol Gregor. 'Neural Variational Inference and Learning in Belief Networks'. In: 32 (2014). DOI: 10.1117/12.526813 (cited on page 14).

[38] Geoffrey E Hinton and Ruslan R Salakhutdinov. 'Reducing the dimensionality of data with neural networks'. In: *science* 313.5786 (2006), pp. 504–507 (cited on page 15).

[39] Martin J Wainwright, Tommi S Jaakkola, and Alan S Willsky. 'A new class of upper bounds on the log partition function'. In: *IEEE Transactions on Information Theory* 51.7 (2005), pp. 2313–2335 (cited on page 15).

[40] Gerasimos G Potamianos and John K Goutsias. 'Partition function estimation of Gibbs random field images using Monte Carlo simulations'. In: *IEEE Transactions on Information Theory* 39.4 (1993), pp. 1322–1332 (cited on page 15).

[41] Frank R Kschischang, Brendan J Frey, Hans-Andrea Loeliger, et al. 'Factor graphs and the sum-product algorithm'. In: *IEEE Transactions on information theory* 47.2 (2001), pp. 498–519 (cited on page 15).

[42] Jingwei Zhang. 'An Optimal Transport View on Generalization'. In: () (cited on page 15).

[43] Raddford Neal and Geoffrey E. Hinton. 'A view of the EM algorithm that justifies incremental, sparse, and other variants'. In: *Learning in graphical models*. Ed. by Springer. 1998, pp. 355–368 (cited on page 16).

[44] Matthew J Beal. 'Variational algorithms for approximate bayesian inference'. In: *PhD Thesis* May (2003), pp. 1–281 (cited on page 17).

[45] Tommi S Jaakkola and Michael I Jordan. 'Bayesian parameter estimation via variational methods'. In: *Statistics and Computing* 10.1 (2000), pp. 25–37 (cited on page 17).

[46] Tommi S Jaakkola. 'Tutorial on Variational Approximation Methods'. In: *In Advanced Mean Field Methods: Theory and Practice* (2000), 129{\textendash}159. DOI: 10.1.1.31.8989 (cited on page 17).

[47] Hagai Attias. 'Inferring Parameters and Structure of Latent Variable Models by Variational Bayes'. In: *Journal of Chemical Information and Modeling* 53.9 (2013), pp. 1689–1699. DOI: 10.1017/CBO9781107415324.004 (cited on page 17).

[48] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 'Variational inference: A review for statisticians'. In: *Journal of the American Statistical Association* just-accepted (2017) (cited on page 17).

[49] Carsten Peterson. 'A mean field theory learning algorithm for neural networks'. In: *Complex systems* 1 (1987), pp. 995–1019 (cited on page 17).

[50] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 'Variational Inference: A Review for Statisticians'. In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877. DOI: 10.1080/01621459.2017.1285773 (cited on page 17).

[51] Michael I. Jordan et al. 'Introduction to variational methods for graphical models'. In: *Machine Learning* 37.2 (1999), pp. 183–233. DOI: 10.1023/A:1007665907178 (cited on page 17).

[52] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006 (cited on page 17).

[53] Martin J Wainwright, Michael I Jordan, et al. 'Graphical models, exponential families, and variational inference'. In: *Foundations and Trends®in Machine Learning* 1.1–2 (2008), pp. 1–305 (cited on page 17).

[54] Iñigo Urteaga and Chris H Wiggins. 'Variational inference for the multi-armed contextual bandit arXiv : 1709 . 03163v1 [stat . ML] 10 Sep 2017'. In: 1933 (2017), pp. 1–13 (cited on page 17).

[55] Dustin Tran, Rajesh Ranganath, and David M. Blei. 'The Variational Gaussian Process'. In: (2015), pp. 1–14 (cited on pages 17, 19).

[56] Roger Frigola, Yutian Chen, and Carl E. Rasmussen. 'Variational Gaussian process state-space models'. In: *Advances in Neural Information Processing Systems* 4.January (2014), pp. 3680–3688 (cited on pages 17, 105).

[57] Lawrence Saul and Michael Jordan. 'Exploiting tractable substructures in intractable networks'. In: (1996), pp. 486–492 (cited on page 18).

[58] Xing E T Al, Eric P Xing, and Michael I Jordan. 'A generalized mean field algorithm for variational inference m exponential families'. In: (1999), pp. 583–591 (cited on page 19).

[59] Rajesh Ranganath et al. 'Deep exponential families'. In: *Artificial Intelligence and Statistics*. 2015, pp. 762–771 (cited on pages 19, 44).

[60] David M Blei, Michael I Jordan, et al. 'Variational inference for Dirichlet process mixtures'. In: *Bayesian analysis* 1.1 (2006), pp. 121–143 (cited on page 19).

[61] Anirban Roychowdhury and Brian Kulis. 'Gamma Processes, Stick-Breaking, and Variational Inference'. In: *arXiv preprint* 1994 (2014), pp. 1–19 (cited on page 19).

[62] Chong Wang and David M Blei. 'Variational inference for the nested Chinese restaurant process'. In: *Advances in Neural Information Processing Systems*. 2009, pp. 1990–1998 (cited on page 19).

[63] Samuel Gershman, Matt Hoffman, and David Blei. 'Nonparametric variational inference'. In: *arXiv preprint arXiv:1206.4665* (2012) (cited on page 19).

[64] Rajesh Ranganath, Dustin Tran, and David Blei. 'Hierarchical variational models'. In: *International Conference on Machine Learning*. 2016, pp. 324–333 (cited on page 19).

[65] Felix Agakov and David Barber. 'An Auxiliary Variational Method'. In: April (2004) (cited on page 19).

[66] Shakir Mohamed and Balaji Lakshminarayanan. 'Learning in implicit generative models'. In: *arXiv preprint arXiv:1610.03483* (2016) (cited on page 19).

[67] Steffen Bickel, Michael Brückner, and Tobias Scheffer. 'Discriminative learning for differing training and test distributions'. In: *ACM International Conference Proceeding Series* 227 (2007), pp. 81–88. DOI: 10.1145/1273496.1273507 (cited on page 19).

[68] Vincent Dumoulin et al. 'Adversarially Learned Inference'. In: (2017), pp. 1–18 (cited on page 19).

[69] Alireza Makhzani et al. 'Adversarial Autoencoders'. In: (2015) (cited on pages 19, 73).

[70] Lars Mescheder et al. 'Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks'. In: () (cited on page 19).

[71] Nikolaos Gianniotis. 'Mixed Variational Inference'. In: (2019), pp. 1–15 (cited on page 19).

[72] Mingzhang Yin and Mingyuan Zhou. 'Semi-Implicit Variational Inference'. In: (2018) (cited on page 19).

[73] Dmitry Molchanov et al. 'Doubly Semi-Implicit Variational Inference'. In: Vi (2018) (cited on page 19).

[74] Jiaxin Shi, Shengyang Sun, and Jun Zhu. 'Implicit Variational Inference with Kernel Density Ratio Fitting'. In: 1 (2017) (cited on page 20).

[75] Arthur Gretton et al. 'A kernel method for the two-sample-problem'. In: *Advances in neural information processing systems*. 2007, pp. 513–520 (cited on pages 20, 21, 50).

[76] Tim Van Erven and Peter Harremo. 'Renyi Divergence and Kullback-Leibler Divergence'. In: 6.1 (2007), pp. 1–24 (cited on pages 20, 25).

[77] Yingzhen Li and Richard E Turner. 'Renyi Divergence Variational Inference'. In: (), pp. 1–20 (cited on page 20).

[78] Ayananendran Bassu, I A N R Harris, and M C Jones. 'Robust and Efficient Estimation by Minimising a Density Power Divergence'. In: (), pp. 1–26 (cited on page 20).

[79] Jean-Baptiste Regli and Ricardo Silva. 'Alpha-Beta Divergence For Variational Inference'. In: (2018) (cited on page 20).

[80] Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. 'Generalized Variational Inference'. In: 1 (2019), pp. 1–61 (cited on page 21).

[81] Arshia Cont, Shlomo Dubnov, and Gérard Assayag. 'On the information geometry of audio streams with applications to similarity computing'. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (2010), pp. 837–846 (cited on page 21).

[82] Yujia Li, Kevin Swersky, and Richard Zemel. 'Generative Moment Matching Networks'. In: (2014) (cited on page 21).

[83] Gintare Karolina Dziugaite and Daniel M Roy. 'Training generative neural networks via Maximum Mean Discrepancy optimization'. In: () (cited on page 21).

[84] Ilya Tolstikhin et al. 'Wasserstein Auto-Encoders'. In: (2017) (cited on pages 21, 50, 72).

[85] Luca Ambrogioni et al. 'Wasserstein Variational Inference'. In: () (cited on page 21).

[86] C Villani. 'Optimal transport: old and new'. In: (2008) (cited on page 21).

[87] Trevor Henderson and Justin Solomon. 'Audio Transport: A Generalized Portamento via Optimal Transport'. In: *arXiv preprint arXiv:1906.06763* (2019) (cited on page 21).

[88] Espen Bernton et al. 'On parameter estimation with the Wasserstein distance'. In: (2017), pp. 1–47 (cited on page 22).

[89] Antoine Rolet, Marco Cuturi, and Gabriel Peyré. 'Fast dictionary learning with a smoothed Wasserstein loss'. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016* 51 (2016), pp. 630–638 (cited on page 22).

[90] Olivier Bousquet et al. 'From optimal transport to generative modeling: the VEGAN cookbook'. In: *arXiv preprint arXiv:1705.07642* (2017) (cited on page 22).

[91] Alireza Makhzani, Brendan Frey, and Ian Goodfellow. 'Adversarial Autoencoders'. In: (2014) (cited on page 22).

[92] Alfréd Rényi. 'On Measures of Entropy and Information'. In: *Journal of Physics* 114.49 (1961), pp. 16184–16188. DOI: 10.1021/jp106846b (cited on page 25).

[93] Eshed Ram. 'On Rényi Entropy Power Inequalities'. In: *IEEE Transactions on Information Theory* (2016) (cited on page 25).

[94] Bruce G. Lindsay. 'The Geometry of Mixture Likelihoods: A General Theory'. In: *The Annals of Statistics* 11.1 (1983), pp. 86–94. DOI: 10.1214/aos/1176348654 (cited on page 25).

[95] Naftali Tishby and Noga Zaslavsky. 'Deep learning and the information bottleneck principle'. In: *2015 IEEE Information Theory Workshop, ITW 2015* (2015). DOI: 10.1109/ITW.2015.7133169 (cited on page 26).

[96] Alexander A Alemi et al. 'An Information-Theoretic Analysis of Deep Latent-Variable Models'. In: *arXiv preprint arXiv:1711.00464* (2017) (cited on page 26).

[97] Jorma Rissanen. 'A Universal Prior for Integers and Estimation by Minimum Description Length'. In: *Annals of Statistics* 14.2 (1986), pp. 590–606 (cited on page 27).

[98] Geoffrey E Hinton and Richard S Zemel. 'Autoencoders, minimum description length and Helmholtz free energy'. In: *Advances in neural information processing systems*. 1994, pp. 3–10 (cited on pages 27, 28).

[99] Karol Gregor et al. 'Deep AutoRegressive Networks'. In: 32 (2013) (cited on page 28).

[100] Hariharan Narayanan and Sanjoy Mitter. 'Sample complexity of testing the manifold hypothesis'. In: *Nips* (2010), pp. 1–9. DOI: 10.1590/0074-02760150454 (cited on page 29).

[101] Shuyin Xia et al. 'Effectiveness of the Euclidean distance in high dimensional spaces'. In: *Optik* 126.24 (2015), pp. 5614–5619. DOI: 10.1016/j.ijleo.2015.09.093 (cited on page 29).

[102] Svante Wold, Kim Esbensen, and Paul Geladi. 'Principal component analysis'. In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52 (cited on page 30).

[103] Jonathon Shlens. 'A tutorial on principal component analysis'. In: *arXiv preprint arXiv:1404.1100* (2014) (cited on page 30).

[104] Emmanuel J. Candes et al. 'Robust Principal Component Analysis?' In: (2009). DOI: 10.1145/1970392.1970395 (cited on page 30).

[105] Hui Zou, Trevor Hastie, and Robert Tibshirani. 'Sparse principal component analysis'. In: *Journal of computational and graphical statistics* 15.2 (2006), pp. 265–286 (cited on page 30).

[106] M. E. Tipping and C. M. Bishop. *Probabilistic Principal Component Analysis*. 19991. DOI: 10.1111/1467-9868.00196 (cited on page 30).

[107] Aapo Hyvarinen and Erkki Oja. 'Independent component analysis by general nonlinear Hebbian-like learning rules'. In: *Signal Processing* 64 (1998), pp. 301–313. DOI: 10.1016/s0165-1684(97)00197-7 (cited on page 30).

[108] Harri Valpola et al. 'Nonlinear Independent Component Analysis Using Ensemble Learning: Experiments and Discussion'. In: *Proc. 2ˆ{\;nd} Int. Workshop on Independent Component Analysis and Blind Signal Separation – (ICA 2000)* (2000), pp. 351–356 (cited on page 30).

[109] Aapo Hyvärinen and Petteri Pajunen. 'Nonlinear independent component analysis: Existence and uniqueness results'. In: *Neural Networks* 12.3 (1999), pp. 429–439 (cited on page 31).

[110] Joseph B Kruskal and Myron Wish. *Multidimensional scaling*. Vol. 11. Sage, 1978 (cited on pages 31, 88).

[111] Geoffrey E Hinton and Sam T Roweis. 'Stochastic neighbor embedding'. In: *Advances in neural information processing systems*. 2003, pp. 857–864 (cited on page 31).

[112] Laurens van der Maaten and Geoffrey Hinton. 'Visualizing data using t-SNE'. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605 (cited on page 32).

[113] S. Mika et al. 'Kernel PCA and de-noising in feature spaces'. In: *Advances in Neural Information Processing Systems* i (1999) (cited on page 32).

[114] S. T. Roweis. 'Nonlinear Dimensionality Reduction by Locally Linear Embedding'. In: *Science* 290.5500 (2000), pp. 2323–2326. DOI: 10.1126/science.290.5500.2323 (cited on page 33).

[115] Mikhail Belkin and Partha Niyogi. 'Laplacian Eign Map'. In: *Neural Computation* 15 (2003), pp. 1373–1396 (cited on page 33).

[116] David L Donoho and Carrie Grimes. 'PNAS_Hessen eigenmap.pdf'. In: (2003). DOI: http://www.pnas.orgcgidoi10.1073pnas.1031596100/ (cited on page 33).

[117] Z Zhang and H Zha. *Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment (Dept of Computer Science, Pennsylvania State Univ, University Park, PA)*. Tech. rep. Tech Rep CSE-02-019, 2002 (cited on page 33).

[118] Joshua B Tenenbaum, Vin De Silva, and John C Langford. 'A global geometric framework for nonlinear dimensionality reduction'. In: *science* 290.5500 (2000), pp. 2319–2323 (cited on page 33).

[119] Leland McInnes, John Healy, and James Melville. 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction'. In: (2018) (cited on page 33).

[120] Herbert Robbins and Sutton Monro. 'A stochastic approximation method'. In: *The annals of mathematical statistics* (1951), pp. 400–407 (cited on page 35).

[121] John Duchi, Elad Hazan, and Yoram Singer. 'Adaptive subgradient methods for online learning and stochastic optimization'. In: *Journal of Machine Learning Research* 12.Jul (2011), pp. 2121–2159 (cited on page 35).

[122] Tijmen Tieleman and Geoffrey Hinton. 'Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude'. In: *COURSERA: Neural networks for machine learning* 4.2 (2012), pp. 26–31 (cited on page 35).

[123] Diederik P Kingma and Jimmy Ba. 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980* (2014) (cited on pages 35, 70, 127).

[124] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 'Multilayer feedforward networks are universal approximators'. In: *Neural networks* 2.5 (1989), pp. 359–366 (cited on page 36).

[125] Kurt Hornik. 'Approximation Capabilities of Multilayer Neural Network'. In: *Neural Networks* 4.1991 (1991), pp. 251–257 (cited on page 36).

[126] Ken-Ichi Funahashi. 'On the approximate realization of continuous mappings by neural networks'. In: *Neural networks* 2.3 (1989), pp. 183–192 (cited on page 36).

[127] Zhou Lu et al. 'The expressive power of neural networks: A view from the width'. In: *Advances in Neural Information Processing Systems* 2017-Decem.Nips (2017), pp. 6232–6240 (cited on page 36).

[128] Boris Hanin. 'Universal Function Approximation by Deep Neural Nets with Bounded Width and ReLU Activations'. In: *arXiv preprint arXiv:1708.02691* (2017) (cited on page 36).

[129] Ronen Eldan and Ohad Shamir. 'The power of depth for feedforward neural networks'. In: *Conference on learning theory*. 2016, pp. 907–940 (cited on page 36).

[130] Monica Bianchini and Franco Scarselli. 'On the complexity of shallow and deep neural network classifiers'. In: *22nd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2014 - Proceedings* April (2014), pp. 371–376 (cited on page 36).

[131] Guido Montúfar et al. 'On the number of linear regions of deep neural networks'. In: *Advances in Neural Information Processing Systems* 4.January (2014), pp. 2924–2932 (cited on page 36).

[132] Samuel S Schoenholz et al. 'Deep Information Propagation'. In: *arXiv preprint arXiv:1611.01232* (2016) (cited on page 36).

[133] Kaiming He et al. 'Identity mappings in deep residual networks'. In: *European conference on computer vision*. Springer. 2016, pp. 630–645 (cited on page 36).

[134] Emmanuel Abbe and Colin Sandon. 'Provable limitations of deep learning'. In: (2018) (cited on page 37).

[135] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 'Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps'. In: (2013), pp. 1–8 (cited on page 37).

[136] Sebastian Bach et al. 'On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation'. In: *PLoS ONE* 10.7 (2015), pp. 1–46. DOI: 10.1371/journal.pone.0130140 (cited on page 37).

[137] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 'Learning important features through propagating activation differences'. In: *34th International Conference on Machine Learning, ICML 2017* 7 (2017), pp. 4844–4866 (cited on page 37).

[138] Marylou Gabrié et al. 'Entropy and mutual information in models of deep neural networks'. In: (2018). DOI: arXiv:1805.09785v1 (cited on page 37).

[139] Ziv Goldfeld et al. 'Estimating Information Flow in Neural Networks arXiv : 1810 . 05728v1 [cs . LG] 12 Oct 2018'. In: () (cited on page 37).

[140] Andrew Saxe et al. 'On The Information Bottleneck of Deep Learning'. In: *Iclr 2018* 24.3 (2018), pp. 3–6. DOI: 10.1108/eb040537 (cited on page 37).

[141] Henry W Lin, Max Tegmark, and David Rolnick. 'Why does deep and cheap learning work so well?' In: *Journal of Statistical Physics* 168.6 (2017), pp. 1223–1247 (cited on page 37).

[142] Chiyuan Zhang et al. *Theory of deep learning iii: Generalization properties of sgd*. Tech. rep. Discussion paper, Center for Brains, Minds and Machines (CBMM). Preprint, 2017 (cited on page 37).

[143] Nitish Srivastava et al. 'Dropout: a simple way to prevent neural networks from overfitting'. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958 (cited on page 37).

[144] Alex Hernández-García and Peter König. 'Do deep nets really need weight decay and dropout?' In: (2018), pp. 1–5 (cited on page 37).

[145] Radford M. Neal. 'Bayesian Learning for Neural Networks'. In: 118 (1996). DOI: 10.1007/978-1-4612-0745-0 (cited on page 37).

[146] Nir Friedman and Iftach Nachman. 'Gaussian process networks'. In: *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 2000, pp. 211–219 (cited on page 37).

[147] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. 'Learning Represenations by Back Propogation Errors'. In: *Letters to Nature* 5.3 (323), pp. 533–536 (cited on page 38).

[148] Pierre Baldi and Kurt Hornik. 'Neural networks and principal component analysis: Learning from examples without local minima'. In: *Neural Networks* 2.1 (1989), pp. 53–58. DOI: 10.1016/0893-6080(89)90014-2 (cited on page 38).

[149] Marc'Aurelio Ranzato et al. 'Efficient Learning of Sparse Representations with an Energy-Based Model Marc'Aurelio'. In: *Advances in neural information processing systems* (2007), pp. 1137–1144 (cited on page 38).

[150] Alessandro Achille and Stefano Soatto. 'Emergence of Invariance and Disentanglement in Deep Representations '. In: 18 (2018), pp. 1–34 (cited on pages 39, 52, 73).

[151] Pascal Vincent et al. *Extracting and Composing Robust Features with Denoising Autoencoders*. Tech. rep. 2008 (cited on page 39).

[152] Pascal Vincent et al. 'Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion'. In: *Journal of Machine Learning Research* 11.Dec (2010), pp. 3371–3408 (cited on page 39).

[153] Guillaume Alain and Yoshua Bengio. 'What Regularized Auto-Encoders Learn from the Data Generating Distribution'. In: (2011), pp. 1–31 (cited on page 39).

[154] Salah Rifai et al. 'Contractive auto-encoders: Explicit invariance during feature extraction'. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 833–840 (cited on page 39).

[155] Matthew D Hoffman et al. 'Stochastic variational inference'. In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 1303–1347 (cited on page 41).

[156] Rajesh Ranganath, Sean Gerrish, and David M Blei. 'Black Box Variational Inference'. In: () (cited on page 41).

[157] George Casella and Christian P Robert. 'Rao-Biackwellization of Sampling Schemes Rao-Blackwellization of Sampling Schemes'. In: July (1994) (cited on page 41).

[158] Sheldon M Ross and Kyle Y Lin. 'Applying variance reduction ideas in queuing simulations'. In: *Probability in the Engineering and Informational Sciences* 15.4 (2001), pp. 481–494 (cited on page 41).

[159] Yann LeCun et al. 'Backpropagation applied to handwritten zip code recognition'. In: *Neural computation* 1.4 (1989), pp. 541–551 (cited on pages 42, 66).

[160] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 'Stochastic backpropagation and approximate inference in deep generative models'. In: *arXiv preprint arXiv:1401.4082* (2014) (cited on pages 43, 51).

[161] Samuel R Bowman et al. 'Generating Sentences from a Continuous Space'. In: (2015) (cited on pages 43, 44, 48).

[162] Chris Cremer, Xuechen Li, and David Duvenaud. 'Inference Suboptimality in Variational Autoencoders'. In: (2018) (cited on page 43).

[163] Xi Chen et al. 'Variational Lossy Autoencoder'. In: (2016), pp. 1–17 (cited on page 44).

[164] Serena Yeung et al. 'Tackling Over-pruning in Variational Autoencoders'. In: *arXiv preprint arXiv:1706.03643* (2017) (cited on pages 44, 48, 49).

[165] Jakub M. Tomczak and Max Welling. 'Improving Variational Auto-Encoders using Householder Flow'. In: 2 (2016) (cited on page 44).

[166] Ke Sun and Xiangliang Zhang. 'Coarse Grained Exponential Variational Autoencoders'. In: *arXiv preprint arXiv:1702.07904* (2017) (cited on page 44).

[167] Shun-Ichi Amari. *Information Geometry and Its Applications (front matter)*. Vol. 194. 2000, p. 736 (cited on page 44).

[168] Prasoon Goyal et al. 'Nonparametric Variational Auto-encoders for Hierarchical Representation Learning'. In: *arXiv preprint arXiv:1703.07027* (2017) (cited on page 45).

[169] E G Tabak and Cristina V Turner. 'A family of nonparametric density estimation algorithms'. In: *Communications on Pure and Applied Mathematics* 66.2 (2000), pp. 145–164 (cited on page 45).

[170] Esteban G. Tabak and Eric Vanden-Eijnden. 'Density estimation by dual ascent of the log-likelihood'. In: *Communications in Mathematical Sciences* 8.1 (2009), pp. 217–233 (cited on page 45).

[171] Diederik P. Kingma and Prafulla Dhariwal. 'Glow: Generative Flow with Invertible 1x1 Convolutions'. In: 2 (2018), pp. 1–10 (cited on pages 45, 59).

[172] Danilo Jimenez Rezende and Shakir Mohamed. 'Variational Inference with Normalizing Flows'. In: 37 (2015) (cited on pages 45, 117).

[173] Diederik P Kingma et al. 'Improved variational inference with inverse autoregressive flow'. In: *Advances in Neural Information Processing Systems*. 2016, pp. 4743–4751 (cited on pages 46, 49).

[174] George Papamakarios, Theo Pavlakou, and Iain Murray. 'Masked Autoregressive Flow for Density Estimation'. In: *arXiv preprint arXiv:1705.07057* (2017) (cited on page 46).

[175] Micha Livne and David Fleet. 'TzK: Flow-Based Conditional Generative Model'. In: NeurIPS (2019), pp. 1–7 (cited on page 46).

[176] Prince Zizhuang Wang. 'Riemannian Normalizing Flow on Variational Wasserstein Autoencoder for Text Modeling'. In: (2018) (cited on page 46).

[177] Chris Cremer, Quaid Morris, and David Duvenaud. 'Reinterpreting Importance-Weighted Autoencoders'. In: *arXiv preprint arXiv:1704.02916* (2017) (cited on page 47).

[178] Andriy Mnih and Danilo Rezende. 'Variational inference for monte carlo objectives'. In: *International Conference on Machine Learning*. 2016, pp. 2188–2196 (cited on page 47).

[179] Chris J Maddison et al. 'Filtering Variational Objectives'. In: *arXiv preprint arXiv:1705.09279* (2017) (cited on page 47).

[180] Bin Dai et al. 'Veiled Attributes of the Variational Autoencoder'. In: *arXiv preprint arXiv:1706.05148* (2017) (cited on pages 48, 51).

[181] Andrea Asperti. 'Sparsity in Variational Autoencoders'. In: 2 () (cited on page 48).

[182] Michal Rolinek, Dominik Zietlow, and Georg Martius. 'Variational Autoencoders Pursue PCA Directions (by Accident)'. In: (2018), pp. 1–22 (cited on page 48).

[183] Samuel R. Bowman et al. 'Generating Sentences from a Continuous Space'. In: (2015). DOI: 10.18653/v1/K16-1002 (cited on pages 49, 109).

[184] Casper Kaae Sønderby et al. 'How to train deep variational autoencoders and probabilistic ladder networks'. In: *arXiv preprint arXiv:1602.02282* (2016) (cited on pages 49, 125).

[185] Xi Chen et al. 'Variational lossy autoencoder'. In: *arXiv preprint arXiv:1611.02731* (2016) (cited on page 49).

[186] Ali Razavi et al. 'Preventing Posterior Collapse with delta-VAEs'. In: (2019) (cited on page 49).

[187] Matthew D Hoffman and Matthew J Johnson. 'Elbo surgery: yet another way to carve up the variational evidence lower bound'. In: *Workshop in Advances in Approximate Bayesian Inference, NIPS*. 2016 (cited on page 49).

[188] Giorgio Patrini et al. 'Sinkhorn AutoEncoders'. In: (2018) (cited on page 50).

[189] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 'InfoVAE: Information Maximizing Variational Autoencoders'. In: (2017) (cited on page 50).

[190] Hiroshi Takahashi et al. 'Variational Autoencoder with Implicit Optimal Priors'. In: (2018) (cited on page 50).

[191] Ehsan Abbasnejad, Anthony Dick, and Anton van den Hengel. 'Infinite Variational Autoencoder for Semi-Supervised Learning'. In: *arXiv preprint arXiv:1611.07800* (2016) (cited on page 51).

[192] Finale Doshi-Velez and Been Kim. 'Towards A Rigorous Science of Interpretable Machine Learning'. In: Ml (2017), pp. 1–13 (cited on page 51).

[193] Ilyes Khemakhem, Diedrik Kingma, and Aapo Hyvärinen. 'Variational Autoencoders and Nonlinear ICA: A Unifying Framework'. In: () (cited on page 51).

[194] Irina Higgins et al. 'beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework'. In: *Iclr* July (2017), pp. 1–13 (cited on pages 51, 73).

[195] Yan Zhang et al. 'Information Potential Auto-Encoders'. In: (2017), pp. 1–15 (cited on page 52).

[196] Sherjil Ozair, Corey Lynch, and Yoshua Bengio. 'Wasserstein Dependency Measure for Representation Learning'. In: (2015) (cited on page 52).

[197] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. 'Latent Space Oddity: on the Curvature of Deep Generative Models'. In: 2 (2017), pp. 1–15 (cited on pages 52, 77).

[198] P. Thomas Fletcher Hang Shao, Abhishek Kumar. 'The Riemannian Geometry of Deep Generative Models'. In: (2017) (cited on page 53).

[199] Ricky T Q Chen et al. 'Neural Ordinary Differential Equations'. In: NeurIPS (2018), pp. 1–19 (cited on page 53).

[200] Vlado Menkovski and Jacobus W Portegies. 'Diffusion Variational Autoencoders'. In: (2019) (cited on page 53).

[201] Ivan Ovinnikov. 'Poincaré Wasserstein Autoencoder'. In: NeurIPS (2018) (cited on page 53).

[202] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 'Hyperbolic Neural Networks'. In: (2018) (cited on page 53).

[203] Gaëtan Hadjeres, Frank Nielsen, and François Pachet. 'GLSR-VAE: Geodesic Latent Space Regularization for Variational AutoEncoder Architectures'. In: (2017), pp. 1–11 (cited on page 53).

[204] Joseph Redmon and Ali Farhadi. 'YOLO9000: Better, Faster, Stronger'. In: (2016). DOI: 10.1142/9789812771728_0012 (cited on page 53).

[205] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 'ImageNet Classification with Deep Convolutional Neural Networks Alex'. In: *Handbook of Approximation Algorithms and Metaheuristics* (2007), pp. 60–1–60–16. DOI: 10.1201/9781420010749 (cited on page 53).

[206] Shlomo E Chazan, Sharon Gannot, and Jacob Goldberger. 'Deep clustering based on a mixture of autoencoders'. In: () (cited on page 53).

[207] Klaus Greff et al. 'Multi-Object Representation Learning with Iterative Variational Inference'. In: (2019) (cited on page 53).

[208] Karol Gregor et al. 'DRAW: A Recurrent Neural Network For Image Generation'. In: (2015) (cited on page 53).

[209] S M Ali Eslami et al. 'Attend, infer, repeat: Fast scene understanding with generative models'. In: *Advances in Neural Information Processing Systems*. 2016, pp. 3225–3233 (cited on page 53).

[210] Jason Tyler Rolfe. 'Discrete Variational Autoencoders'. In: (2016), pp. 1–33 (cited on page 54).

[211] Akash Srivastava and Charles Sutton. 'Autoencoding variational inference for topic models'. In: *arXiv preprint arXiv:1703.01488* (2017) (cited on page 54).

[212] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 'Neural Discrete Representation Learning'. In: Nips (2017) (cited on page 54).

[213] Abubakar Abid, Muhammad Fatih, and Balin James. 'Concrete Autoencoders for Differentiable Feature Selection and Reconstruction'. In: (1997) (cited on page 54).

[214] Eric Jang, Shixiang Gu, and Ben Poole. 'Categorical Reparameterization with Gumbel-Softmax'. In: (2016), pp. 1–13 (cited on page 54).

[215] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 'The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables'. In: (2016), pp. 1–20 (cited on page 54).

[216] Jiacheng Xu and Greg Durrett. 'Spherical Latent Spaces for Stable Variational Autoencoders'. In: (2018) (cited on page 55).

[217] Nat Dilokthanakul et al. 'Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders'. In: 2016 (2016), pp. 1–12 (cited on page 55).

[218] Lars Maaløe et al. 'Auxiliary deep generative models'. In: *arXiv preprint arXiv:1602.05473* (2016) (cited on page 55).

[219] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. 'Multi-Level Variational Autoencoder: Learning Disentangled Representations from Grouped Observations'. In: *arXiv preprint arXiv:1705.08841* (2017) (cited on page 56).

[220] Philippe Esling and Carlos Agon. 'Multiobjective time series matching for audio classification and retrieval'. In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.10 (2013), pp. 2057–2072 (cited on page 57).

[221] J Markel and A Gray. 'On autocorrelation equations as applied to speech analysis'. In: *IEEE Transactions on audio and Electroacoustics* 21.2 (1973), pp. 69–79 (cited on page 57).

[222] Olivier Lartillot, Petri Toiviainen, and Tuomas Eerola. 'A matlab toolbox for music information retrieval'. In: *Data analysis, machine learning and applications*. Springer, 2008, pp. 261–268 (cited on page 58).

[223] Jaap Haitsma and Ton Kalker. 'Speed-change resistant audio fingerprinting using auto-correlation'. In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).* Vol. 4. IEEE. 2003, pp. IV–728 (cited on page 58).

[224] MIReS Consortium. *Roadmap for Music Information Research*. Tech. rep. 2013 (cited on page 58).

[225] Rainer Typke, Frans Wiering, and Remco C Veltkamp. 'A survey of music information retrieval systems'. In: *Proc. 6th International Conference on Music Information Retrieval*. Queen Mary, University of London. 2005, pp. 153–160 (cited on page 58).

[226] Meinard Muller et al. 'Signal processing for music analysis'. In: *IEEE Journal of Selected Topics in Signal Processing* 5.6 (2011), pp. 1088–1110 (cited on page 58).

[227] Alain De Cheveigné and Hideki Kawahara. 'YIN, a fundamental frequency estimator for speech and music'. In: *The Journal of the Acoustical Society of America* 111.4 (2002), pp. 1917–1930 (cited on page 58).

[228] Gaël Richard and Roland Badeau. 'Détection de fréquences fondamentales multiples'. In: () (cited on page 58).

[229] Geoffroy Peeters. 'Music pitch representation by periodicity measures based on combined temporal and spectral representations'. In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 5. IEEE. 2006, pp. V–V (cited on page 58).

[230] Miguel A Alonso Arevalo. 'Extraction d'information rythmique à partir d'enregistrements musicaux'. PhD thesis. Télécom ParisTech, 2006 (cited on page 58).

[231] Gilbert Nouno. 'Suivi de tempo appliqué aux musiques improvisées, à la recherche du temps perdu...' PhD thesis. Paris 6, 2008 (cited on page 58).

[232] Peter Grosche, Meinard Müller, and Frank Kurth. 'Cyclic tempogram—A mid-level tempo representation for musicsignals'. In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2010, pp. 5522–5525 (cited on page 58).

[233] Juan P Bello. 'Measuring structural similarity in music'. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.7 (2011), pp. 2013–2025 (cited on page 58).

[234] Mark Levy and Mark Sandler. 'Structural segmentation of musical audio by constrained clustering'. In: *IEEE transactions on audio, speech, and language processing* 16.2 (2008), pp. 318–326 (cited on page 58).

[235] Goffredo Haus, Luca A Ludovico, and Giorgio Presti. 'Automatic Annotation of Timbre Variation for Musical Instruments'. In: *Computer Music Multidisciplinary Research*. Les éditions de PRISM. 2017, pp. 493–504 (cited on page 58).

[236] Thomas Kemp et al. 'Strategies for automatic segmentation of audio data'. In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*. Vol. 3. IEEE. 2000, pp. 1423–1426 (cited on page 58).

[237] Yi-Hsuan Yang and Homer H Chen. 'Machine recognition of music emotion: A review'. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 3.3 (2012), p. 40 (cited on page 58).

[238] Jean-Julien Aucouturier et al. 'Covert digital manipulation of vocal emotion alter speakers' emotional states in a congruent direction'. In: *Proceedings of the National Academy of Sciences* 113.4 (2016), pp. 948–953 (cited on page 58).

[239] Bob L Sturm. 'A survey of evaluation in music genre recognition'. In: *International Workshop on Adaptive Multimedia Retrieval*. Springer. 2012, pp. 29–66 (cited on page 58).

[240] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. 'Automatic genre classification of music content: a survey'. In: *IEEE Signal Processing Magazine* 23.2 (2006), pp. 133–141 (cited on page 58).

[241] Marius Kaminskas and Francesco Ricci. 'Contextual music information retrieval and recommendation: State of the art and challenges'. In: *Computer Science Review* 6.2-3 (2012), pp. 89–119 (cited on page 58).

[242] Peter Knees and Markus Schedl. 'A survey of music similarity and recommendation from music context data'. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 10.1 (2013), p. 2 (cited on page 58).

[243] Geoffray Bonnin and Dietmar Jannach. 'Automated generation of music playlists: Survey and experiments'. In: *ACM Computing Surveys (CSUR)* 47.2 (2015), p. 26 (cited on page 58).

[244] Jan Schlüter and Sebastian Böck. 'Improved musical onset detection with convolutional neural networks'. In: *2014 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE. 2014, pp. 6979–6983 (cited on page 58).

[245] Hong Su et al. 'Convolutional neural network for robust pitch determination'. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 579–583 (cited on page 58).

[246] Eric J Humphrey and Juan P Bello. 'Rethinking automatic chord recognition with convolutional neural networks'. In: *2012 11th International Conference on Machine Learning and Applications*. Vol. 2. IEEE. 2012, pp. 357–362 (cited on page 58).

[247] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. 'An end-to-end neural network for polyphonic piano music transcription'. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.5 (2016), pp. 927–939 (cited on page 58).

[248] Antoine Liutkus, Roland Badeau, and Gäel Richard. 'Gaussian processes for underdetermined source separation'. In: *IEEE Transactions on Signal Processing* 59.7 (2011), pp. 3155–3167 (cited on page 58).

[249] Benoit Fuentes et al. 'Probabilistic model for main melody extraction using constant-Q transform'. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2012, pp. 5357–5360 (cited on page 58).

[250] Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent. 'Multichannel audio source separation with deep neural networks'. In: *IEEE/ACM Transactions on Audio* 24.10 (2016), pp. 1652–1664. DOI: 10.1109/TASLP.2016.2580946> (cited on pages 58, 64).

[251] Andreas Jansson et al. 'Singing voice separation with deep U-Net convolutional networks'. In: (2017) (cited on page 58).

[252] Curtis Hawthorne et al. 'Onsets and frames: Dual-objective piano transcription'. In: *arXiv preprint arXiv:1710.11153* (2017) (cited on page 58).

[253] Jan Schlüter and Thomas Grill. 'Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks.' In: *ISMIR*. 2015, pp. 121–126 (cited on page 58).

[254] Jean-Claude Risset and David L Wessel. 'Exploration of timbre by analysis and synthesis'. In: *The psychology of music*. Elsevier, 1999, pp. 113–169 (cited on page 58).

[255] Curtis Roads, John Strawn, et al. *The computer music tutorial*. MIT press, 1996 (cited on page 58).

[256] Philippe Depalle, Guillermo Garcia, and Xavier Rodet. 'Tracking of partials for additive sound synthesis using hidden Markov models'. In: *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. IEEE. 1993, pp. 225–228 (cited on page 58).

[257] Piotr Kleczkowski. 'Group additive synthesis'. In: *Computer Music Journal* 13.1 (1989), pp. 12–20 (cited on page 58).

[258] Curtis Roads. 'Introduction to granular synthesis'. In: *Computer Music Journal* 12.2 (1988), pp. 11–13 (cited on page 58).

[259] Julius O Smith. 'Physical modeling using digital waveguides'. In: *Computer music journal* 16.4 (1992), pp. 74–91 (cited on page 58).

[260] Davide Rocchesso, Roberto Bresin, and Mikael Fernstrom. 'Sounding objects'. In: *IEEE MultiMedia* 10.2 (2003), pp. 42–52 (cited on page 58).

[261] Stefan D Bilbao. *Numerical sound synthesis*. Wiley Online Library, 2009 (cited on page 58).

[262] Aaron van den Oord et al. 'Wavenet: A generative model for raw audio'. In: *arXiv preprint arXiv:1609.03499* (2016) (cited on pages 59, 62).

[263] Mathieu Germain et al. 'MADE: masked autoencoder for distribution estimation'. In: *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 2015, pp. 881–889 (cited on page 59).

[264] Mehdi Mirza and Simon Osindero. 'Conditional generative adversarial nets'. In: *arXiv preprint arXiv:1411.1784* (2014) (cited on page 59).

[265] Xi Chen et al. 'Infogan: Interpretable representation learning by information maximizing generative adversarial nets'. In: *Advances in Neural Information Processing Systems*. 2016, pp. 2172–2180 (cited on page 59).

[266] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 'WaveGlow: A Flow-based Generative Network for Speech Synthesis'. In: (2018) (cited on page 59).

[267] Huadong Liao, Jiawei He, and Kunxian Shu. 'Generative Model with Dynamic Linear Flow'. In: (2019) (cited on page 59).

[268] Roland Badeau, Gaël Richard, and Bertrand David. 'Fast adaptive esprit algorithm'. In: *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005*. IEEE. 2005, pp. 289–294 (cited on page 59).

[269] Sadaoki Furui. 'Cepstral analysis technique for automatic speaker verification'. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29.2 (1981), pp. 254–272 (cited on page 59).

[270] Xavier Rodet and Philippe Depalle. 'Spectral envelopes and inverse FFT synthesis'. In: *Audio Engineering Society Convention 93*. Audio Engineering Society. 1992 (cited on page 64).

[271] Jonathan Allen. 'Short term spectral analysis, synthesis, and modification by discrete Fourier transform'. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25.3 (1977), pp. 235–238 (cited on page 64).

[272] Judith C Brown. 'Calculation of a constant Q spectral transform'. In: *The Journal of the Acoustical Society of America* 89.1 (1991), pp. 425–434 (cited on page 64).

[273] Gino Angelo Velasco et al. 'Constructing an invertible constant-Q transform with non-stationary Gabor frames'. In: *Proceedings of DAFX11, Paris* (2011), pp. 93–99 (cited on page 64).

[274] Hans Georg Zimmermann, Alexey Minin, and Victoria Kusherbaeva. 'Comparison of the Complex Valued and Real Valued Neural Networks Trained with Gradient Descent and Random Search Algorithms'. In: April (2011), pp. 27–29 (cited on page 64).

[275] Christian Berg. 'Complex Analysis'. In: (2012), pp. 1–101 (cited on page 64).

[276] Chiheb Trabelsi et al. 'Deep Complex Networks'. In: 2016 (2017), pp. 1–19. DOI: 10.1159/000448939 (cited on page 64).

[277] Bernard Picinbono, Second-order Complex Random, and Normal Distributions Ieee Trans-. 'Second-Order Complex Random Vectors and Normal Distributions To cite this version : HAL Id : hal-01736682 Second-Order Complex Random Vectors and Normal Distributions'. In: (2018) (cited on page 64).

[278] Daniel Griffin and Jae Lim. 'Signal estimation from modified short-time Fourier transform'. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.2 (1984), pp. 236–243 (cited on page 65).

[279] Nathanaël Perraudin, Peter Balazs, and Peter L Søndergaard. 'A fast Griffin-Lim algorithm'. In: *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE. 2013, pp. 1–4 (cited on page 65).

[280] Zdenek Pruša and Peter L Søndergaard. 'Real-time spectrogram inversion using phase gradient heap integration'. In: *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*. 2016, pp. 17–21 (cited on page 65).

[281] Guillaume Ballet et al. 'Studio online 3.0: An internet" killer application" for remote access to ircam sounds and processing tools'. In: *Journée d'Informatique Musicale (JIM)* (1999) (cited on pages 69, 140).

[282] John S Gero and Udo Kannengiesser. 'The situated function–behaviour–structure framework'. In: *Design studies* 25.4 (2004), pp. 373–391 (cited on page 69).

[283] Li-chia Yang Alexander Lerch. 'On the evaluation of generative models in music'. In: (2018) (cited on page 69).

[284] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 'Fast and accurate deep network learning by exponential linear units (elus)'. In: *arXiv preprint arXiv:1511.07289* (2015) (cited on page 70).

[285] Sergey Ioffe and Christian Szegedy. 'Batch normalization: Accelerating deep network training by reducing internal covariate shift'. In: *International Conference on Machine Learning*. 2015, pp. 448–456 (cited on page 70).

[286] Sercan O. Arik, Heewoo Jun, and Gregory Diamos. 'Fast spectrogram inversion using multi-head convolutional neural networks'. In: *IEEE Signal Processing Letters* 26.1 (2019), pp. 94–98. DOI: 10.1109/LSP.2018.2880284 (cited on page 71).

[287] Fumitada Itakura. 'Analysis synthesis telephony based on the maximum likelihood method'. In: *The 6th international congress on acoustics, 1968*. 1968, pp. 280–292 (cited on page 71).

[288] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. 'The Numerics of GANs'. In: Nips (2017) (cited on page 73).

[289] Hyunjik Kim and Andriy Mnih. 'Disentangling by Factorising'. In: (2018) (cited on page 73).

[290] Sangchul Hahn and Heeyoul Choi. 'Disentangling Latent Factors with Whitening'. In: (2018) (cited on page 73).

[291] Cian Eastwood and Christopher K.L. Williams. 'A Framework for the Quantitative Evaluation of Disentangled Representations'. In: *ICLR* (2018) (cited on page 74).

[292] Jeremy Marozeau and Alain de Cheveigné. 'The effect of fundamental frequency on the brightness dimension of timbre'. In: *The Journal of the Acoustical Society of America* 121.1 (2007), pp. 383–387. DOI: 10.1121/1.2384910 (cited on pages 76, 88).

[293] Anssi Klapuri and Manuel Davy. *Signal processing methods for music transcription*. Springer Science & Business Media, 2007 (cited on page 76).

[294] Shlomo Dubnov. 'Generalization of spectral flatness measure for non-Gaussian linear processes'. In: *IEEE Signal Processing Letters* 11.8 (2004), pp. 698–701. DOI: 10.1109/LSP.2004.831663 (cited on page 76).

[295] Diederik P Kingma et al. 'Semi-supervised learning with deep generative models'. In: *Advances in Neural Information Processing Systems*. 2014, pp. 3581–3589 (cited on pages 79, 80).

[296] Xinchen Yan et al. 'Attribute2image: Conditional image generation from visual attributes'. In: *European Conference on Computer Vision*. Springer. 2016, pp. 776–791 (cited on page 79).

[297] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 'A Learned Representation For Artistic Style'. In: 2016 (2016) (cited on page 79).

[298] Ethan Perez et al. 'FiLM : Visual Reasoning with a General Conditioning Layer'. In: (2017) (cited on page 79).

[299] Adrien Bitton, Philippe Esling, and Axel Chemla-Romeu-Santos. 'Modulated Variational auto-Encoders for many-to-many musical timbre transfer'. In: 2015 (2018), pp. 1–13 (cited on page 79).

[300] Yang Li et al. 'Disentangled Variational Auto-Encoder for Semi-supervised Learning'. In: (2017) (cited on page 79).

[301] Ramakrishna Vedantam et al. 'Generative Models of Visually Grounded Imagination'. In: *arXiv preprint arXiv:1705.10762* (2017) (cited on page 80).

[302] Irina Higgins et al. 'SCAN: Learning Abstract Hierarchical Compositional Visual Concepts'. In: *arXiv preprint arXiv:1707.03389* (2017) (cited on pages 80, 82).

[303] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. 'Joint Multimodal Learning with Deep Generative Models'. In: *arXiv preprint arXiv:1611.01891* (2016) (cited on page 81).

[304] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 'Unsupervised Image-to-Image Translation Networks'. In: 2 (2017) (cited on page 81).

[305] Elena Agullò Cantos. 'Flute audio labelled database for Automatic Music Transcription'. In: () (cited on pages 83, 85).

[306] John M Grey and John W Gordon. 'Perceptual effects of spectral modifications on musical timbres'. In: *The Journal of the Acoustical Society of America* 63.5 (1978), pp. 1493–1500 (cited on page 88).

[307] J. Krimphoff, S. McAdams, and S. Winsberg. 'Caracterisation du timbre des sons complexes. II. analyses acoustiques et quantification psychophysique'. In: *Journal De Physique* 4.5 pt 1 (1994), pp. 625–628 (cited on page 88).

[308] A. M. Noll and M. R. Schroeder. 'Short-Time "Cepstrum" Pitch Detection'. In: *The Journal of the Acoustical Society of America* 36.5 (1964), pp. 1030–1030. DOI: 10.1121/1.2143271 (cited on page 88).

[309] William Brent. 'Cepstral analysis tools for percussive timbre identification'. In: *Proceedings of the 3rd International Pure Data Convention, Sao Paulo, Brazil*. sn. 2009 (cited on page 88).

[310] Antti Eronen and Anssi Klapuri. 'Musical instrument recognition using cepstral coefficients and temporal features'. In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*. Vol. 2. IEEE. 2000, pp. II753–II756 (cited on page 88).

[311] Olivier Lartillot and Petri Toiviainen. 'A Matlab toolbox for musical feature extraction from audio'. In: *International conference on digital audio effects*. Bordeaux. 2007, pp. 237–244 (cited on page 88).

[312] Stephen McAdams and Bruno L Giordano. 'The perception of musical timbre'. In: *The Oxford handbook of music psychology* (2009), pp. 72–80 (cited on page 88).

[313] Kai Siedenburg, Ichiro Fujinaga, and Stephen McAdams. 'A Comparison of Approaches to Timbre Descriptors in Music Information Retrieval and Music Psychology'. In: *Journal of New Music Research* 45.1 (2016), pp. 27–41. DOI: 10.1080/09298215.2015.1132737 (cited on page 88).

[314] Anne Caclin et al. 'Acoustic correlates of timbre space dimensions: A confirmatory study using synthetic tones'. In: *The Journal of the Acoustical Society of America* 118.1 (2005), pp. 471–482. DOI: 10.1121/1.1929229 (cited on page 88).

[315] Stephen McAdams and Jean-Christophe Cunible. 'Perception of Timbral Analogies'. In: *The Royal Society* 367.1587 (1993), pp. 430–438 (cited on page 88).

[316] Reinier Plomp. 'Auditory analysis and timbre perception'. In: *Auditory analysis and perception of speech* (1975), pp. 7–22 (cited on page 89).

[317] J Douglas Carroll and Jih-Jie Chang. 'Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition'. In: *Psychometrika* 35.3 (1970), pp. 283–319 (cited on page 89).

[318] Arnab Ghosh et al. 'Multi-Agent Diverse Generative Adversarial Networks'. In: *arXiv preprint arXiv:1704.02906* (2017) (cited on page 89).

[319] Paul Iverson and Carol L Krumhansl. 'Isolating the dynamic attributes of musical timbrea'. In: *The Journal of the Acoustical Society of America* 94.5 (1993), pp. 2595–2603 (cited on page 89).

[320] Stephen McAdams et al. 'Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes'. In: *Psychological Research* 58.3 (1995), pp. 177–192. DOI: 10.1007/BF00419633 (cited on page 89).

[321] Stephen Lakatos. 'A common perceptual space for harmonic and percussive timbres'. In: *Perception & psychophysics* 62.7 (2000), pp. 1426–1439 (cited on page 89).

[322] Philippe Esling, Axel Chemla-Romeu-Santos, and Adrien Bitton. 'Bridging Audio Analysis, Perception and Synthesis with Perceptually-regularized Variational Timbre Spaces.' In: *ISMIR*. 2018, pp. 175–181 (cited on page 91).

[323] Douglas O'Shaughnessy. 'Linear predictive coding'. In: *IEEE potentials* 7.1 (1988), pp. 29–32 (cited on page 102).

[324] W Bastiaan Kleijn. 'Continuous representations in linear predictive coding'. In: *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*. IEEE. 1991, pp. 201–204 (cited on page 102).

[325] HS Peter Yue and Rafi Rabipour. *Methods and apparatus for noise conditioning in digital speech compression systems using linear predictive coding*. US Patent 5,642,464. June 1997 (cited on page 102).

[326] Masayuki Nishiguchi and Jun Matsumoto. 'Harmonic and noise coding of LPC residuals with classified vector quantization'. In: *1995 International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. IEEE. 1995, pp. 484–487 (cited on page 102).

[327] Kevin Patrick Murphy. 'Dynamic Bayesian Networks: Representation, Inference and Learning'. In: (1994) (cited on page 103).

[328] Clare A McGrory and DM Titterington. 'Variational Bayesian analysis for hidden Markov models'. In: *Australian & New Zealand Journal of Statistics* 51.2 (2009), pp. 227–244 (cited on page 103).

[329] Nick Foti et al. 'Stochastic variational inference for hidden Markov models'. In: *Advances in neural information processing systems*. 2014, pp. 3599–3607 (cited on page 103).

[330] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning*. Vol. 1. MIT press Cambridge, 2006 (cited on page 104).

[331] Matthias Seeger. 'Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations'. In: (2004). DOI: 10.1162/153244303765208377 (cited on page 104).

[332] Michalis K. Titsias. 'Variational Learning of Inducing Variables in Sparse Gaussian Processes'. In: *Artificial Intelligence and Statistics* 5 (2009), pp. 567–574. DOI: 10.1117/12.591679 (cited on page 104).

[333] Andrew Gordon Wilson, David A. Knowles, and Zoubin Ghahramani. 'Gaussian process regression networks'. In: *Proceedings of the 29th International Conference on Machine Learning, ICML 2012* 1 (2012), pp. 599–606 (cited on page 104).

[334] Neil Lawrence. 'Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models'. In: *Journal of Machine Learning Research* 6 (2005), pp. 1783–1816 (cited on page 105).

[335] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. 'Gaussian process dynamical models'. In: *Advances in Neural Information Processing Systems* (2005), pp. 1441–1448 (cited on page 105).

[336] Stefanos Eleftheriadis et al. 'Identification of Gaussian process state space models'. In: *Advances in Neural Information Processing Systems* 2017-Decem.2014 (2017), pp. 5310–5320 (cited on page 105).

[337] Andreas Doerr et al. 'Probabilistic Recurrent State-Space Models'. In: *35th International Conference on Machine Learning, ICML 2018* 3 (2018), pp. 2060–2075 (cited on page 105).

[338] Lincoln C Mattos et al. 'Recurrent Gaussian Processes'. In: 3 (2016), pp. 1–12 (cited on page 105).

[339] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. 'Learning representations by back-propagating errors'. In: *Cognitive modeling* 5.3 (1988), p. 1 (cited on page 106).

[340] Jeffrey L. Elman. 'Finding structure in time'. In: *Cognitive Science* 14.2 (1990), pp. 179–211. DOI: 10.1016/0364-0213(90)90002-E (cited on page 106).

[341] Randall D. Beer. 'On the Dynamics of Small Continuous-Time Recurrent Neural Networks'. In: *Adaptive Behavior* 3.4 (1995), pp. 469–509. DOI: 10.1177/105971239500300405 (cited on page 106).

[342] Kenji Doya. 'Bifurcations of recurrent neural networks in gradient descent learning'. In: *IEEE Transactions on neural networks* 1.75 (1993), p. 164 (cited on page 106).

[343] Herbert Jaeger and Harald Haas. 'Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication'. In: *Science* 304.5667 (2004), pp. 78–80. DOI: 10.1126/science.1091277 (cited on page 106).

[344] Alex M Lamb et al. 'Professor forcing: A new algorithm for training recurrent networks'. In: *Advances In Neural Information Processing Systems*. 2016, pp. 4601–4609 (cited on page 106).

[345] Salah El Hihi and Yoshua Bengio. 'Hierarchical Recurrent Neural Networks for Long-Term Dependencies.' In: *Nips* (1995), pp. 493–499 (cited on page 106).

[346] Sepp Hochreiter and Jürgen Schmidhuber. 'Long short-term memory'. In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cited on page 107).

[347] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. 'Learning to forget: Continual prediction with LSTM'. In: *Neural Computation* 12.10 (2000), pp. 2451–2471. DOI: 10.1162/089976600300015015 (cited on page 107).

[348] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 'Bidirectional LSTM networks for improved phoneme classification and recognition'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3697 LNCS (2005), pp. 799–804 (cited on page 107).

[349] Ashish Vaswani et al. 'Attention Is All You Need'. In: Nips (2017). DOI: 10.1017/S0952523813000308 (cited on page 107).

[350] Sneha Chaudhari et al. 'An Attentive Survey of Attention Models'. In: (2019) (cited on page 107).

[351] Kazuki Irie et al. 'Language Modeling with Deep Transformers'. In: () (cited on page 107).

[352] Mostafa Dehghani et al. 'Universal Transformers'. In: (2018), pp. 1–23 (cited on page 107).

[353] Kyunghyun Cho et al. 'On the properties of neural machine translation: Encoder-decoder approaches'. In: *arXiv preprint arXiv:1409.1259* (2014) (cited on page 107).

[354] Moritz Wolter and Angela Yao. 'FOURIER RNNS FOR SEQUENCE ANALYSIS AND PREDICTION'. In: (2018) (cited on page 107).

[355] Marta Garnelo et al. 'Neural Processes'. In: (2018) (cited on page 107).

[356] Timon Willi et al. 'Recurrent Neural Processes'. In: (2019) (cited on page 107).

[357] Chao Ma, Yingzhen Li, and José Miguel Hernández-Lobato. 'Variational Implicit Processes'. In: (2018) (cited on page 107).

[358] Hugo Larochelle and Iain Murray. 'The neural autoregressive distribution estimator'. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, pp. 29–37 (cited on page 108).

[359] Tapani Raiko et al. 'Iterative neural autoregressive distribution estimator nade-k'. In: *Advances in neural information processing systems*. 2014, pp. 325–333 (cited on page 108).

[360] Soroush Mehri et al. 'SampleRNN: An Unconditional End-to-End Neural Audio Generation Model'. In: (2016), pp. 1–11 (cited on page 108).

[361] Albert Haque, Michelle Guo, and Prateek Verma. 'Conditional End-to-End Audio Transforms'. In: i (2018), pp. 1–5 (cited on page 109).

[362] Jesse Engel et al. 'Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders'. In: *arXiv preprint arXiv:1704.01279* (2017) (cited on page 109).

[363] Evan Archer et al. 'Black box variational inference for state space models'. In: (2015), pp. 1–11. DOI: 10.1038/nature08121 (cited on page 110).

[364] Rahul G. Krishnan, Uri Shalit, and David Sontag. 'Deep Kalman Filters'. In: 2000 (2015), pp. 1–17 (cited on page 110).

[365] Otto Fabius and Joost R. van Amersfoort. 'Variational Recurrent Auto-Encoders'. In: (2014), pp. 1–5 (cited on pages 110, 111).

[366] Justin Bayer and Christian Osendorfer. 'Variational inference of latent state sequences using recurrent networks'. In: *arXiv preprint arXiv:1406.1655* (2014), pp. 1–12 (cited on pages 110, 111).

[367] Iulian Vlad Serban et al. 'A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues.' In: *AAAI*. 2017, pp. 3295–3301 (cited on page 110).

[368] Karol Gregor et al. 'Temporal Difference Variational Auto-Encoder'. In: (2018), pp. 1–17 (cited on page 111).

[369] Marco Fraccaro et al. 'Sequential neural models with stochastic layers'. In: *Advances in Neural Information Processing Systems* (2016), pp. 2199–2207 (cited on page 111).

[370] Anirudh Goyal, Alessandro Sordoni, and Nan Rosemary Ke. 'Z-Forcing : Training Stochastic Recurrent Networks'. In: Nips (2017) (cited on page 111).

[371] Samira Shabanian et al. 'Variational Bi-LSTMs'. In: (2017), pp. 1–12 (cited on page 111).

[372] Justin Bayer and Christian Osendorfer. 'Variational inference of latent state sequences using recurrent networks'. In: *stat* 1050 (2014), p. 6 (cited on pages 111, 113).

[373] Junyoung Chung et al. 'A recurrent latent variable model for sequential data'. In: *Advances in neural information processing systems*. 2015, pp. 2980–2988 (cited on pages 111, 113).

[374] Yingzhen Li and Stephan Mandt. 'Disentangled Sequential Autoencoder'. In: (2018) (cited on page 112).

[375] Will Grathwohl and Aaron Wilson. 'Disentangling Space and Time in Video with Hierarchical Variational Auto-encoders'. In: *arXiv preprint arXiv:1612.04440* (2016) (cited on page 112).

[376] Aaron Van Den Oord. 'Representation Learning with Contrastive Predictive Coding'. In: () (cited on page 114).

[377] Geoffrey E Hinton, Terrance Sejnowski, and David Ackley. *Boltzman Machines*. 1984 (cited on page 114).

[378] Ruslan Salakhutdinov and Geoffrey Hinton. 'Deep Boltzmann machines'. In: *Journal of Machine Learning Research* 5.3 (2009), pp. 448–455 (cited on page 114).

[379] Antti Rasmus et al. 'Semi-supervised learning with ladder networks'. In: *Advances in Neural Information Processing Systems*. 2015, pp. 3546–3554 (cited on page 124).

[380] Casper Kaae Sønderby et al. 'Ladder variational autoencoders'. In: *Advances in Neural Information Processing Systems*. 2016, pp. 3738–3746 (cited on page 125).

[381] Adam Paszke et al. 'PyTorch: An imperative style, high-performance deep learning library'. In: *Advances in Neural Information Processing Systems*. 2019, pp. 8024–8035 (cited on page 133).

[382] Rebecca Fiebrink and Perry R Cook. 'The Wekinator: a system for real-time, interactive machine learning in music'. In: *Proceedings of The Eleventh International Society for Music Information Retrieval Conference (ISMIR 2010)(Utrecht)*. 2010 (cited on page 135).

[383] Sølvi Ystad, Mitsuko Aramaki, and Richard Kronland-Martinet. *Timbre from Sound Synthesis and High-Level Control Perspectives*. 2019, pp. 361–389 (cited on page 138).

[384] Garrett Warnell et al. 'Deep TAMER: Interactive agent shaping in high-dimensional state spaces'. In: *32nd AAAI Conference on Artificial Intelligence, AAAI 2018* (2018), pp. 1545–1553 (cited on page 138).

[385] Hugo Scurto, Frédéric Bevilacqua, and Baptiste Caramiaux. 'Perceiving Agent Collaborative Sonic Exploration In Interactive Reinforcement Learning'. In: 2018 (cited on page 139).

[386] Hugo Scurto et al. 'Designing Deep Reinforcement Learning for Human Parameter Exploration'. In: *arXiv preprint arXiv:1907.00824* (2019) (cited on page 139).

[387] Gérard Assayag et al. 'Omax brothers: a dynamic yopology of agents for improvization learning'. In: *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*. ACM. 2006, pp. 125–132 (cited on page 139).

[388] Atau Tanaka and Marco Donnarumma. 'The body as musical instrument'. In: *The Oxford Handbook of Music and the Body* (2019), p. 79 (cited on page 140).

[389] Philippe Esling et al. 'Universal audio synthesizer control with normalizing flows'. In: *CoRR* abs/1907.00971 (2019) (cited on page 145).