Bridging the inference gap in Mutimodal Variational Autoencoders

Agathe Senellart, Stéphanie Allassonnière, Agathe Senellart

Highlights

Bridging the inference gap in Mutimodal Variational Autoencoders

Agathe Senellart, Stéphanie Allassonnière

- Development of two novels methods for modeling and generating multimodal data, based on the Variational Autoencoder framework, Normalizing Flows and Self-Supervised Learning.
- Comprehensive evaluation showing superior performance compared to state-of-the-art models on several benchmark datasets.

Bridging the inference gap in Mutimodal Variational Autoencoders

Agathe Senellart[a], Stéphanie Allassonnière[a]

[a] *UMR1346, Université de Paris Cité, Inria Paris, Inserm,*

Abstract

From medical diagnosis to autonomous vehicles, critical applications rely on the integration of multiple heterogeneous data modalities. Multimodal Variational Autoencoders offer versatile and scalable methods for generating unobserved modalities from observed ones. Recent models using mixtures-of-experts aggregation suffer from theoretically grounded limitations that restrict their generation quality on complex datasets. In this article, we propose a novel interpretable model able to learn both joint and conditional distributions without introducing mixture aggregation. Our model follows a multistage training process: first modeling the joint distribution with variational inference and then modeling the conditional distributions with Normalizing Flows to better approximate true posteriors. Importantly, we also propose to extract and leverage the information shared between modalities to improve the conditional coherence of generated samples. Our method achieves state-of-the-art results on several benchmark datasets.

Keywords:
Multimodality, Variational Autoencoders, Normalizing Flows, Contrastive Learning

1. Introduction

In many cases, information is conveyed through multiple heterogeneous modalities. In the medical field, a patient's status is comprehensively described through various analyses: sonograms, MRI, blood analysis, textual reports, etc ... Taking different views into account jointly leads to richer representations and a better understanding of the modalities' interactions. Two important challenges in multimodal machine learning are the tasks of *learn-*

ing relevant joint representations and *generating realistic data*, either *from one modality to another or in all modalities jointly.*

Multimodal Variational Autoencoders are latent generative models that can be used to tackle both challenges at the same time. In recent years, several approaches have been proposed to extend the Variational Autoencoder [1] (VAE) to efficiently model multimodal data. Some of them suggest training *coordinated* VAEs where the latent spaces for all modalities are constrained to be similar [2, 3, 4]. In other works, a single latent space is used to jointly represent all modalities [5, 6, 7]. Among these models, one popular approach is to aggregate modalities using a simple function such as Product-of-Experts [6], or Mixture-of-Experts [7], [8]. Aggregation has the advantage of requiring fewer parameters and therefore being easily scalable. However, recent works show that it can limit the quality and diversity of generated samples [9, 8].

In this article, we propose a new flexible VAE-based framework that can model the **joint and conditional distributions across any number of modalities**. In particular, our main contributions are:

- Development of two novel multimodal VAE-based methods for modeling and generating multimodal data. Unlike recent approaches, we do not use aggregation, which enables us to improve generation, particularly by integrating Normalizing Flows and leveraging shared information across modalities.

- Comprehensive evaluation on several benchmark datasets demonstrating that the proposed models outperform recent methods.

2. Background

Mathematically speaking, we assume that we observe multimodal samples $X = (x_1, x_2, \dots, x_M)$ with M modalities from an unknown distribution $p(X)$. We aim to approximate this joint distribution as well as the conditional distributions with parametric ones $p_\theta(X)$, $p_\theta(x_j|x_i)$ for any $1 \leq i \neq j \leq M$. $p_\theta(x_j|x_i)$ is the distribution of one modality x_j *given* x_i.

In the VAE framework, one assumes that there exists a shared latent representation z, from which all modalities can be generated with parametric distributions $(p_\theta(x_j|z))_{1\leq j\leq M}$ called *decoders*. For instance, for an image modality x_1, $p_\theta(x_1|z)$ can be a Gaussian distribution $\mathcal{N}(\mu_\theta(z), \Sigma_\theta(z))$ whose mean and variance are given by a neural network. In most cases [6, 10, 5],

each modality is supposed to be conditionally independent of the others given z, such that the joint model writes:

$$p_\theta(X, z) = p_\theta(X|z)p_\theta(z) = p_\theta(z)\prod_{j=1}^{M} p_\theta(x_j|z)\,, \tag{1}$$

where $p_\theta(z)$ is a *prior* distribution over the latent variables and θ refers to all parameters used to model the prior and the decoders. In that framework, the two goals mentioned above (model the joint and conditional distributions) translate as follows: first, we want to learn the best possible θ to model the observations. Secondly, we want to approximate the *inference distributions* $p_\theta(z|(x_j)_{j\in S})$ to infer the latent variable from any given subset of modalities $S \in \mathcal{P}(M)$ where $\mathcal{P}(M) = \{S|S \subset [|1, M|] \text{ and } S \neq \emptyset\}$. If we can infer z from observed modalities, we can then generate unobserved modalities with the decoders $(p_\theta(x_j|z))_{1\leq j\leq M}$. In the rest of the article, we note $x_S \coloneqq (x_j)_{j\in S}$ to simplify notations.

2.1. Estimating the generative model's parameter θ

Given N multimodal observations $(X^{(i)})_{1\leq i\leq N}$, a natural objective to estimate θ is to optimize the log-likelihood of the data [1]:

$$\theta^* \in \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_\theta(X^{(i)}) = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \left(\log \int_z p_\theta(X^{(i)}, z)dz\right)\,.$$

Since this objective is intractable, one can resort to Variational Inference [11, 1] by introducing an auxiliary parametric distribution $q_\phi(z|X)$ allowing us to derive an unbiased estimate of the likelihood of the data:

$$\widehat{p}_\theta(X, z) = \frac{p_\theta(X, z)}{q_\phi(z|X)} \quad \text{such that} \quad p_\theta(X) = \mathbb{E}_{q_\phi(z|X)}\left[\widehat{p}_\theta\right]\,. \tag{2}$$

Then, using Jensen's inequality allows us to derive a lower bound on $\log p_\theta(X)$, referred to as the Evidence Lower Bound (ELBO).

$$\begin{aligned}\log p_\theta(X) &= \log \mathbb{E}_{q_\phi(z|X)}\left[\widehat{p}_\theta\right] \\ &\geq \mathbb{E}_{q_\phi(z|X)}\left[\log p_\theta(X|z)\right] - KL(q_\phi(z|X)||p_\theta(z)) = \mathcal{L}(X; \theta, \phi)\,.\end{aligned} \tag{3}$$

This bound is tractable and can be optimized through Stochastic Gradient Descent [1]. Noteworthy, the first term can be seen as a reconstruction error

and the second as a regularization term encouraging latent embeddings to follow the prior distribution [12]. The distribution $q_\phi(z|X)$ is generally called the *encoder* and one may prove that:

$$\mathcal{L}(X;\theta,\phi) = \log p_\theta(X) - KL(q_\phi(z|X)||p_\theta(z|X))\,. \tag{4}$$

This implies that maximizing $\mathcal{L}(X;\theta,\phi)$ with respect to ϕ leads to minimizing the Kullback-Leibler (KL) divergence between the true posterior $p_\theta(z|X)$ and its variational approximation $q_\phi(z|X)$ [1]. Some models also rely on variations of Eq. (3) to learn θ: [8, 13] adds a β factor to weigh the KL term in (3). That hyperparameter can be tuned to promote disentanglement in the latent space [14]: by increasing the KL term, it increases pressure on the latent variables to be independent, so that a single unit might encode a single generative factor. Other models [7, 13] use a k-sampled importance weighted estimate of the log-likelihood (IWAE bound) [15] or replace the KL with a Jensen-Shannon divergence [16].

2.2. Choice of the approximate inference distribution

A simple choice is to model the approximate posterior $q_\phi(z|X)$ as a Gaussian distribution $\mathcal{N}(\mu_\phi(X), \Sigma_\phi(X))$ where a dedicated joint encoder network takes all modalities as input and outputs the parameters $\mu_\phi(X), \Sigma_\phi(X)$. By maximizing $\mathcal{L}$, we obtain an estimation of θ and an approximation of the joint posterior $p_\theta(z|X)$ with $q_\phi(z|X)$. However, we do not have access to the remaining subset posteriors $(p_\theta(z|x_S))_{S\in\mathcal{P}(M)}$ which are *intractable*. To estimate these posterior distributions, two approaches have been proposed, which we detail in the following paragraphs.

2.3. Surrogate distributions and learning objectives

First, a few models such as JMVAE [5], or TELBO [17] introduce surrogate parametric distributions $(q_{\phi_S}(z|x_S))_{S\in\mathcal{P}(M)}$ and train them with an additional loss function to approximate the desired posterior distributions. However, those models use quite a large number of parameters since the joint posterior $q_\phi(z|X)$ and each approximate posterior $(q_{\phi_S}(z|x_S))_{S\in\mathcal{P}(M)}$ use a dedicated network encoder. The number of parameters then scale with the number of subsets $|\mathcal{P}(M)| = 2^M$.

2.4. Aggregated models

Aggregated models compute the joint posterior $q_\phi(z|X)$ as an aggregation of unimodal encoders $q_{\phi_j}(z|x_j)$ for $1 \leq j \leq M$. MVAE [6] uses a Product-of-Experts (PoE) operation $q_\phi(z|X) \propto p_\theta(z) \prod_j q_{\phi_j}(z|x_j)$ while MMVAE [7] uses a Mixture-of-Experts (MoE). Many variants were then introduced such as Mixture-of-Product of Experts [8] or Generalized Product of Experts [18]. Such a choice for $q_\phi(z|X)$ has several advantages. First it reduces the number of trainable parameters since $q_\phi(z|X)$ shares the same parameters as the unimodal encoders $(q_{\phi_j}(z|x_j))_{1\leq j\leq M}$. To model a subset posterior $q_\phi(z|x_S)$ for $S \in \mathcal{P}(M)$, no additional parameter is necessary; one can simply aggregate on the modalities in S. Therefore, these models are easily scalable to large number of modalities. Furthermore, optimizing Eq. 3 allows to optimize the generative parameter θ and all inference parameters $\phi = (\phi_j)_{1\leq j\leq M}$ without introducing additional objectives to the loss function. In particular, [8] rewrites the ELBO (3) to explicitly highlight how these aggregation methods encourage each estimated posteriors $q_{\phi_j}(z|x_j)$ to be close to the true joint posterior $p_\theta(z|X)$.

However, it has been shown [9] that all *mixture-based* models suffer from a fundamental limitation that caps their generative quality. More precisely, for these models, there is a generative discrepancy $\Delta(X)$ between the log-likelihood of the data and the ELBO:

$$\mathbb{E}_{p(X)}(\log(p_\theta(X))) \geq \mathbb{E}_{p(X)}(\mathcal{L}(X;\theta,\phi)) + \Delta(X)\,, \tag{5}$$

where $p(X)$ is the observed empirical distribution. $\Delta(X)$ is strictly positive and only depends on the law of X and the mixture components [9].

Using $\mathbb{E}_{p(X)}(\log(p_\theta(X)) - \mathcal{L}(X;\theta,\phi)) = \mathbb{E}_{p(X)}\left(KL\left(q_\phi(z|X)||p_\theta(z|X)\right)\right)$, one can rewrite (5) as:

$$\mathbb{E}_{p(X)}\left(KL(q_\phi(z|X)||p_\theta(z|X))\right) \geq \Delta(X)\,. \tag{6}$$

This lower bound implies that the approximate joint posterior $q_\phi(z|X)$ can only approach the true joint posterior $p_\theta(z|X)$ up to $\Delta(X) > 0$.

The authors in [9] detail in extensive experiments how these generative discrepancy results in a diminished quality of generated samples.

For aggregated models that are only based on a Product-of-Experts such as the MVAE, this issue is avoided but a trade-off is observed between the generative coherence and the generative diversity [8].

2.5. Recent developments

In order to compensate for this diversity/coherence trade-off, additional terms might be added to the ELBO to further ensure certain properties of the unimodal encoders. For instance, the MVTCAE model adds Conditional Variational Information Bottleneck (CVIB) terms to the ELBO [19] while the CRMVAE model adds unimodal reconstruction terms [20]. Another approach is to modify the training paradigm with a contrastive learning objective [10]. Recently, methods have been proposed with more complex generative models including multiple, separated [8, 21, 22] or hierarchical latent variables [23, 24]. An additional goal of these models is to separate into different latent spaces the information shared across modalities from modality-specific factors. Models using multiple latent variables are sometimes sensitive to the *shortcut* issue, referring to shared information leaking into the modality specific latent spaces. Recently, MMVAE+ [13] was proposed with an amended ELBO loss and modalities' specific priors to limit that phenomenon [13]. However the MMVAE+ is still based on a mixture aggregation and therefore suffers from the intrinsic limitation mentioned above in Eq. (6), which we observe in our experiments.

Finally, recent work complement multimodal VAEs with diffusion models to improve generative quality [25, 24].

3. Proposed method

To overcome the generative discrepancy gap observed in mixture-based models, we propose to disentangle the training of the joint generative model $p_\theta(X)$ and the approximation of the posteriors $p_\theta(z|x_j)$ for $1 \leq j \leq M$ in the same line of work as [5, 17].Therefore our method consists of two separate steps:

- Train a Variational Autoencoder to learn the generative model θ as well as an approximation of the joint posterior $q_\phi(z|X)$.
- For conditional generation, approximate the unimodal posteriors with Normalizing Flows [26] $q_{\phi_j}(z|x_j)$ for $1 \leq j \leq M$.

For the subset posteriors, we show that, for any $S \in \mathcal{P}(M)$, we can approximate $p_\theta(z|x_S)$ with a Product-of-Experts $\prod_{j\in S} q_{\phi_j}(z|x_j)/p_\theta(z)^{|S|-1}$. This way, no additional network needs to be trained and our framework scales for large numbers of modalities. Note that this Product-of-Experts

is only used during inference *after* the training and not in the optimization of the multimodal ELBO, which means that it doesn't suffer from the same limitations as PoE aggregated models. In the following subsections, we detail each step of our method, and then we introduce an improvement that leverages information shared across modalities.

3.1. Step 1: Training the joint generative model

For learning the generative parameter θ, we optimize the ELBO (3) with a β factor weighting the regularization term. We model the joint encoder $q_\phi(z|X)$ as a Gaussian distribution $\mathcal{N}(\mu_\phi(X), \Sigma_\phi(X))$, with $\mu_\phi(X)$ and $\Sigma_\phi(X)$ given by a neural network taking all modalities as inputs. This step is exactly similar to training a unimodal VAE, and every improvement that was proposed for the unimodal case could be easily adapted here.

3.2. Step 2: Learning the posterior distributions

Once the generative model is learned, we freeze the generative model $p_\theta(X|z)$ and the joint encoder $q_\phi(z|X)$. For $1 \leq j \leq M$ we introduce a surrogate distribution $q_{\phi_j}(z|x_j)$ to approximate the unimodal posterior $p_\theta(z|x_j)$ that is intractable. To fit these distributions, we minimize the following objective introduced in [5].

$$\mathcal{L}_{uni}(X; (\phi_j)_{1\leq j\leq M}) = \sum_{j=0}^{M} KL(q_\phi(z|X)|q_{\phi_j}(z|x_j))\,. \tag{7}$$

Intuitively, minimizing (7) encourages $q_{\phi_j}(z|x_j)$ to cover all the relevant modes or support of the trained posterior $q_\phi(z|X)$. Since $q_\phi(z|X)$ is frozen, minimizing Equation (7) amounts to minimizing:

$$\tilde{\mathcal{L}_{uni}}(X; (\phi_j)_{1\leq j\leq M}) = -\sum_{j=0}^{M} \mathbb{E}_{q_\phi(z|X)}\left(\log q_{\phi_j}(z|x_j)\right)\,. \tag{8}$$

For $1 \leq j \leq M$, the expectation inside the sum can be estimated by sampling $z \sim q_\phi(z|X)$. Equation (8) shows that during training, the unimodal encoders are *informed* by the joint encoder: a latent variable z is sampled from $q_\phi(z|X)$ and then for each $1 \leq j \leq M$, $\log q_{\phi_j}(z|x_j)$ is maximized. [17, 5] and [19] provide interesting interpretations of this objective that we detail in Appendix C. In particular, one can prove that for any $1 \leq j \leq M$, optimizing (7) brings $q_{\phi_j}(z|x_j)$ close to an average distribution $q_\phi^{(avg)}(z|x_j) \coloneqq \mathbb{E}_{p((x_i)_{i\neq j}|x_j)}(q_\phi(z|X))$.

This loss function is used in [5], but the JMVAE model suffers from poor coherence in certain use cases. One reason for this is the use of Gaussian distributions to model $q_{\phi_j}(z|x_j)$ for $1 \leq j \leq M$, which lacks flexibility for approximating the true posteriors. We transform these gaussian distributions using Normalizing Flows which allow us to better approximate complex distributions. Normalizing Flows are a powerful modeling tool that enables the modeling of complex, differentiable distributions [26]. A flow is an invertible smooth transformation f that can be applied to an initial distribution to create a new one, such that if Z is a random vector with density $q(z)$, then $Z' = f(Z)$ has a density given by:

$$q'(z') = q(z) \left| \det \frac{\partial f^{-1}}{\partial z'} \right| = q(z) \left| \det \frac{\partial f}{\partial z} \right|^{-1} . \tag{9}$$

Combining K transformations $z_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(z_0)$ allows us to gain in complexity of the final distribution.

In our case, for each modality $1 \leq j \leq M$, we model the approximate posterior $q_{\phi_j}(z|x_j)$ with the following log-density:

$$\log q_{\phi_j}(z|x_j) = \log q^{(0)}_{\phi_j}(z_0|x_j) - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k^{(j)}}{\partial z_{k-1}} \right| , \tag{10}$$

where $q^{(0)}_{\phi_j}(z_0|x_j)$ is a simple parametrized Gaussian distribution, the parameters of which are given by neural networks, and $(f_k^{(j)})_{1 \leq k \leq K}$ are Masked Autoregressive Flows [27]. In section 4.1, we illustrate that this expression allow us to approximate much more precisely the true unimodal posteriors. Because of the joint training of Normalizing Flows during this step, we refer to our model as JNF.

3.3. Sampling from the subset posteriors

Recall that one of our goals is to be able to infer the latent variable z from *any subset of modalities* $S \in \mathcal{P}(M)$. Until now, we have estimated the joint posterior with $q_\phi(z|X)$ and the unimodal posteriors with $q_{\phi_j}(z|x_j)$ for any $j \in [|1, M|]$. Using the same derivation as [6], we prove that we can approximate any subset posterior using the trained unimodal encoders.

Let $S \in \mathcal{P}(M)$ and $x_S = (x_j)_{j \in S}$:

$$p_\theta(z|x_S) = \frac{p_\theta(x_S|z)p_\theta(z)}{p_\theta(x_S)} = \frac{p_\theta(z)\prod_{j\in S} p_\theta(x_j|z)}{p_\theta(x_S)} = \frac{p_\theta(z)\prod_{j\in S} \frac{p_\theta(x_j,z)}{p_\theta(z)}}{p_\theta(x_S)} \quad (11)$$

$$= \frac{\prod_{j\in S} p_\theta(z|x_j)p_\theta(x_j)}{p_\theta(z)^{|S|-1}p_\theta(x_S)} = \frac{1}{Z}\frac{\prod_{j\in S} p_\theta(z|x_j)}{p_\theta(z)^{|S|-1}} \approx \frac{1}{Z}\frac{\prod_{j\in S} q_{\phi_j}(z|x_j)}{p_\theta(z)^{|S|-1}} \quad (12)$$

where $\frac{1}{Z} = \frac{\prod_{j\in S} p_\theta(x_j)}{p_\theta(x_S)}$ is a normalizing constant. We use Equation (1) in the second equality. To sample from this distribution at inference time, we use Hamiltonian Monte Carlo (HMC) sampling [28, 29] that enables sampling from any distribution with a differentiable density function known up to a multiplicative constant. We recall the algorithm for HMC in Appendix F.

3.4. An improvement of our method leveraging shared information

Up until now, we have not made any assumption regarding the interactions between modalities $(x_1, x_2, \ldots, x_M)$. However, in many multimodal datasets, there is an amount of *shared* semantic information between modalities. For instance, in the MNIST-SVHN dataset [30, 31], the shared semantic content is the digit present in both images. The background information is *modality-specific* in the sense that it doesn't affect other modalities. To generate unobserved modalities, one would only need the shared semantic content and not the modality specific information. Therefore, it seems relevant to try to extract and use this *shared* information. Formally, let us assume that for any $1 \leq j \leq M$ we have a projector g_j such that:

$$\forall 1 \leq i \leq M, p_\theta(x_i|x_j) = p_\theta(x_i|g_j(x_j))\,. \quad (13)$$

Morally speaking, g_j extracts the information shared across modalities while tuning out the modality specific information. Then we can write:

$$p_\theta(x_i|g_j(x_j)) = \int_z p_\theta(x_i|z)p_\theta(z|g_j(x_j))dz \quad (14)$$

That is, to generate modality x_i from modality x_j, we can learn to approximate $p_\theta(z|g_j(x_j))$ which might be simpler than approximating $p_\theta(z|x_j)$ if we use relevant functions $(g_j)_{1\leq j\leq M}$.

We propose an improvement of our method, in which we use *pretrained* functions $(g_j)_{1\leq j\leq M}$ to extract shared information across modalities and model

the distributions $q_{\phi_j}(z|g_j(x_j))$ instead of $q_{\phi_j}(z|x_j)$. In that case, we model $q_{\phi_j}(z|g_j(x_j))$ with Normalizing Flows and use the adapted loss function for step 2 (Section 3.2):

$$\mathcal{L}_{uni}^{(shared)}(X;(\phi_j)_{1\leq j\leq M}) = \sum_{j=0}^{M} KL(q_\phi(z|X)|q_{\phi_j}(z|g_j(x_j)))\,. \tag{15}$$

Extracting information shared across modalities. How can we learn relevant functions $(g_j)_{1\leq j\leq M}$ that would verify Equation (13)? Many methods have been proposed to extract information shared across modalities, and the best method might depend on the dataset, which is why this is a flexible component of our method. In our experiments, we tried two general methods: Deep Canonical Correlation Analysis (DCCA) [32] and Contrastive Learning (CL) [33, 34, 35, 36]. In both cases, the projectors $(g_j)_{1\leq j\leq M}$ are trained *jointly* to learn *similar* representations across modalities. For the projections $(g_j(x_j))_{1\leq j\leq M}$ to be *similar* across modalities, the projectors have to extract shared information while discarding unrelated information. The notion of similarity is defined differently in both methods: DCCA maximizes correlation between projections while CL optimizes cosine similarity. We detail each method in Appendix B. We *conjecture* that using these methods, we can extract summary statistics $(g_j(x_j))_{1\leq j\leq M}$ verifying Equation (13) and check this assumption in our experiments. Note that the projectors (g_j) are trained *before* training the VAE and that existing pretrained networks could also be used.

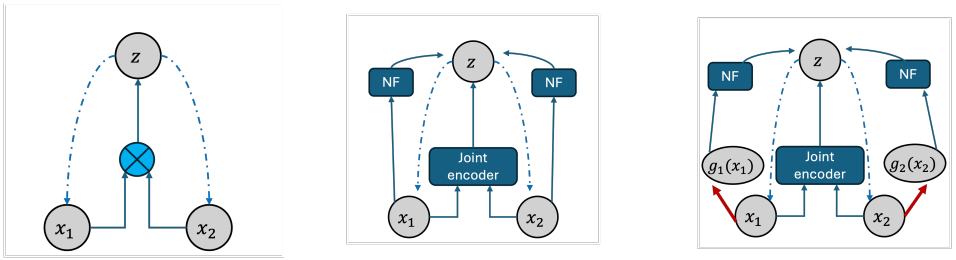
We refer to this improvement of our method as *JNF-Shared*. In Figure 1, we summarize both models and aggregated models in the case $M = 2$.

4. Experiments

In this section, we first illustrate our method on a toy dataset, and then compare results against state-of-the-art methods.

4.1. Toy dataset

We design a toy dataset with two black and white image modalities: x_1 is a square and x_2 is a circle. The sizes of each shape are independent. There are two classes of shapes: the *full* shapes and the *empty* ones. This class is shared across modalities: if the circle is full, the square is full regardless of their size. Figure 2 presents samples of this toy dataset. We perform the first



(a) Aggregated Models (b) JNF model (c) JNF-Shared model

Figure 1: Graphical models in the case $M = 2$. Dashed lines represent decoders, solid lines represent encoders, and red arrows represent the projectors extracting shared information. "NF" refers to Normalizing Flows.

step of our method on this dataset (see 3.1), which is training a simple joint VAE with a two-dimensional latent space that we can visualize. In Figure 2, we can see how this joint latent space is structured, with the full shapes on one side (blue dots) and the empty shapes on the other side (red dots). In Figure 2 ,the intensities of the colors indicate the size distribution with larger squares encoded away from the center.

Using this well-structured latent space we then train $q_{\phi_1}(z|x_1)$ to approximate $p_\theta(z|x_1)$ using our objective $\mathcal{L}_{uni}$ (7). We display an example distribution $q_{\phi_1}(z|x_1)$ that we obtain for x_1 being a large full square. We compare results when modeling $q_{\phi_1}(z|x_1)$ with either a Gaussian or Normalizing Flows (NF). The latter provides a more realistic approximation and generate coherent and diverse samples in the circles modality (shown on the right side of each plot).

In the last plot of Figure 2, we use the variant of our method where we first extract the information shared across modalities (here the emptiness or fullness of the shape) with a projector $g_1(x_1)$ and then approximate $q_{\phi_1}(z|g_1(x_1))$. Here, g_1 and g_2 are neural networks trained with the DCCA objective. We see in Figure 2 right panel, that $q_{\phi_1}(z|g_1(x_1))$ covers well the part of the latent space corresponding to full samples and generates coherent and diverse samples. This shows that we have been able to capture both the conditional distribution and the shared information with $g_1(x_1)$ driving the conditional generation. On this toy dataset, both $p_\theta(z|x_1)$ and $p_\theta(z|g_1(x_1))$ are well approximated but on benchmark datasets, it appears that the latter is often easier to approximate than the former because it has a larger support.

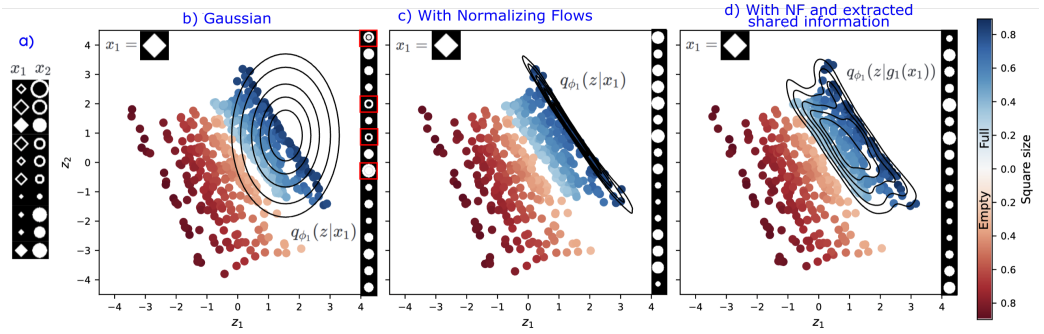


Figure 2: a) Samples from the toy dataset. b) The joint generative model $p_\theta(x_1, x_2)$ has been learned and we visualize the 2-dimensional latent space. Each point encodes a pair of images (x_1, x_2). Here the color of each point, indicates the size and class of the encoded *square*. We try to approximate the posterior $p_\theta(z|x_1)$ of a large square image x_1 (shown in the top left), that corresponds to *dark blue dots* in the latent space. In b), we use a diagonal Gaussian distribution and in c) we use Normalizing Flows. We see that Normalizing Flows capture a realistic posterior where the Gaussian distribution has a support that is too large, leading to unrealistic generation framed in red. d) Using DCCA, we extract the information shared across modalities, which is the shape class: full or empty. We learn $q_{\phi_1}(z|g_1(x_1))$ and see that it approximates well the part of the latent space which encodes full shapes. For b), c), and d) we present samples generated in the circle modality using the learned posterior on the right side of each plot. Both c) and d) produce relevant and diverse samples.

4.2. Benchmark datasets and evaluation metrics

We evaluate JNF and JNF-Shared on four benchmark datasets:

- MNIST-SVHN introduced in [7] that contains paired images from MNIST [30] and the Street View House Numbers (SVHN) dataset [31]. The latter contains natural images of digits with diverse backgrounds and sometimes cropped distracting digits on the sides of the digit of interest.
- PolyMNIST introduced in [8] with five image modalities built from MNIST images with varied and complex backgrounds. This dataset allows to test the scalability of our method.
- Translated PolyMNIST introduced in [9] to demonstrate the limitations of mixture-based models. It is made of downscaled and translated digits with the same backgrounds as PolyMNIST. In [9] the authors point out that the generative performance is very degraded for mixture-based models on this dataset.

- Finally, we test our method on a dataset with heterogeneous modalities: the Multimodal Handwritten Digits dataset (MHD) [23] which contains three modality types: image, sound and trajectory.

We provide additional details and samples for each dataset in Appendix A. We focus on conditional and unconditional generation and we evaluate:

- the *coherence* of multimodal samples. With pretrained classifiers, we assess whether the generated samples are consistent (i.e, share the same label) across modalities.

- the *diversity* of generated samples. To assess this diversity, we follow the procedures used in [13] and [23]. For the MNIST-SVHN and PolyMNIST datasets, we compute Fréchet Inception Distance [37] (FID) between the distributions of true and generated samples. For the MHD dataset, the Inception network is not relevant to extract meaningful features since the modalities that we use are not natural images. Therefore, we use pretrained, class-based and modality specific autoencoders to extract features for each sample and then compute the Mean Fréchet Distance (MFD) between true and generated samples.

4.3. Comparison to previous work

We compare our method to several strong models: JMVAE [5], MMVAE [7], MoPoE [8], MVTCAE model [19], MMVAE+ [13] and Nexus [23]. We use implementations that were first validated by reproducing previous results. For a fair comparison, we use the same architectures and the same latent capacity across models except for the MMVAE and MMVAE+ for which we use smaller latent spaces due to memory limitations from the K-sampled objective. We detail all hyperparameters in Appendix E. We train all models with a β-weighted ELBO and keep the $\beta \in \{0.5, 1, 2.5\}$ that maximizes average coherence for each model. Each experiment is repeated with four different seeds. We try training the projectors (g_j) for our model JNF-Shared with Contrastive Learning (CL) or DCCA and report results for both.

4.4. Experimental results

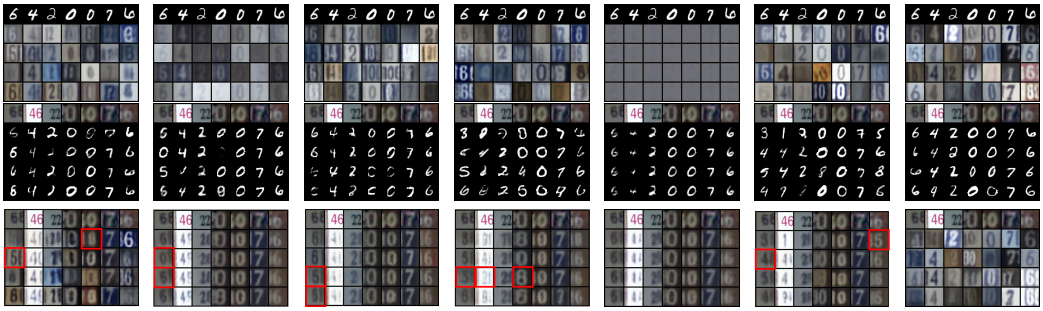
In Figure 3, we present generated samples and in Table 1, we present quantitative results for the MNIST-SVHN dataset.

Most models (except MoPoE and JNF-Shared) struggle to generate coherent MNIST images from SVHN images. We interpret this phenomenon

Model	Joint	M $\longrightarrow$ S	S $\longrightarrow$ M	FID ($\downarrow$)
JMVAE	$\underline{0.43} \pm 0.10$	0.73 ± 0.07	0.53 ± 0.05	57 ± 3
MMVAE ($k = 10, \beta = 0.5$)	0.35 ± 0.02	$\underline{0.80} \pm 0.01$	0.70 ± 0.01	130 ± 5
MVTCAE	$\underline{0.44} \pm 0.02$	$\mathbf{0.81} \pm 0.01$	0.52 ± 0.02	$\mathbf{48} \pm 2$
MoPoE	0.36 ± 0.01	0.12 ± 0.01	$\underline{0.72} \pm 0.01$	359 ± 12
MMVAE+ ($k = 10$)	$\underline{0.43} \pm 0.05$	0.60 ± 0.09	0.58 ± 0.04	63 ± 5
JNF (Ours)	$\mathbf{0.51} \pm 0.01$	$\mathbf{0.82} \pm 0.01$	0.52 ± 0.01	$\underline{54} \pm 2$
JNF-Shared (DCCA) (Ours)	$\mathbf{0.51} \pm 0.01$	0.75 ± 0.03	0.69 ± 0.05	$\underline{53} \pm 2$
JNF-Shared (CL) (Ours)	$\mathbf{0.51} \pm 0.02$	$\mathbf{0.81} \pm 0.01$	$\mathbf{0.75} \pm 0.02$	$\mathbf{49} \pm 1$

Table 1: Results on MNIST-SVHN. We present coherence for joint generation, conditional generation from MNIST (noted as M) to SVHN (noted as S) and vice-versa. FID values are computed on 50,000 SVHN images generated from MNIST. Best values are in bold and second-best are underlined.

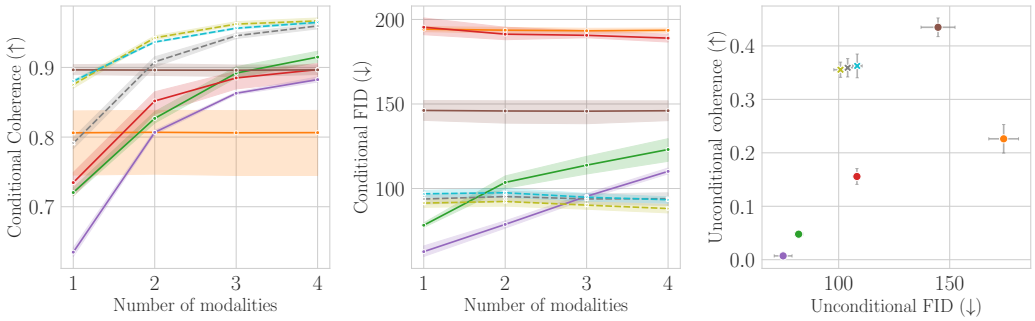
by looking at reconstructed SVHN images in Figure 3. For many models, the background is well reconstructed but not the digit which is not well inferred using $q_{\phi_2}(z|x_2)$ (where x_2 is the SVHN modality). With JNF-Shared, the background is tuned out by the projector g_2 and the digit information is therefore better preserved when sampling $z \sim q_{\phi_2}(z|g_2(x_2))$. Our model JNF-Shared (CL) is the only one to reach competitive values *for all metrics* on this dataset with coherent and diverse generations.



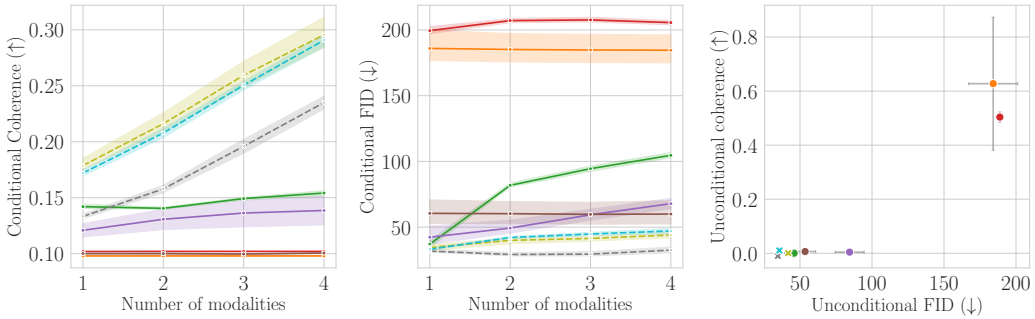
(a) JMVAE (b) MMVAE (c) MMVAE+ (d) MVTCAE (e) MoPoE (f) JNF (g) JNF-CL

Figure 3: On the first row: generation from MNIST to SVHN. On the second row: generation from SVHN to MNIST. On the third row: generation from SVHN to SVHN (unimodal reconstruction). In red, we frame samples where the background is well reconstructed but not the digit. JNF-CL refers to our model JNF-Shared with CL. Note that for this model, when reconstructing SVHN, we sample $z \sim q_{\phi_2}(z|g_2(x_2))$ and therefore the background information is filtered by the projector $g_2(x_2)$ and cannot be reconstructed. However, the digit is well preserved which is what is required for cross-modal generation.

JNF-Shared with CL projectors achieve higher coherence than DCCA projectors, which means that CL better extracts the shared information on this dataset. The MMVAE and MoPoE both produce SVHN samples that look 'averaged' resulting from the quality gap analyzed in [9]. In Figure 4,



(a) PolyMNIST

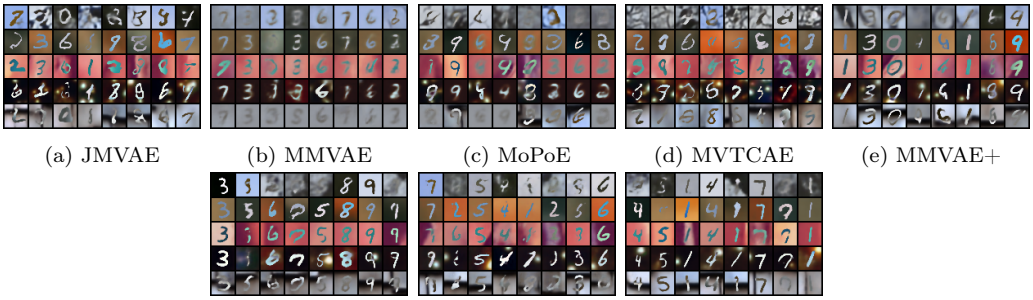


(b) Translated PolyMNIST

Figure 4: In the two left columns, we present results for conditional generation when varying the number of conditioning modalities. In the right column, we display coherence and FID for unconditional generation. Each point correspond to a different training seed. For these plots, best models having high coherence and low FID are in the top left corner. The FID is computed on 10,000 samples of the first modality.

we present coherence and diversity results for all models on PolyMNIST and Translated PolyMNIST. We observe that our models reach the best coherence while maintaining low FID values. We present samples of unconditional generation in Figure 5: our models produce coherent and diverse samples. Our method JNF-Shared works well with both CL and DCCA projectors

on this dataset. In [9], the authors observed that MMVAE and MoPoE show very degraded coherence on TranslatedPolyMNIST. We extend their observations to the MMVAE+ model that also has a conditional coherence close to 0.10, corresponding to random digit association. This is a direct consequence of the mixture aggregation, which limits generative quality on complex datasets [9]. In that setting, these models fail to extract any shared information across modalities. All models fail on the unconditional generation task from the prior: our models have good FID values but very low coherence. To improve this, a possible direction would be to fit a distribution on the latent embeddings *after* training rather than sampling from the prior [38]. On the contrary, MMVAE and MoPoE have high joint coherence but looking at the generated samples in Appendix D, we see that they only produce averaged images of the first digit.



(a) JMVAE (b) MMVAE (c) MoPoE (d) MVTCAE (e) MMVAE+

(f) JNF (g) JNF-CL (h) JNF-DCCA

Figure 5: Joint generation in all five modalities when sampling a latent code from the prior. In each image, each row corresponds to a modality. JNF-CL (resp. DCCA) correspond to our method JNF-Shared with CL (resp. DCCA).

In Table 2, we present results on the MHD dataset, where our models reach the best results for coherence and second best for diversity. For all datasets, additional results and generated samples can be found in Appendix D.

5. Discussion and Perspectives

In this article, we presented two novel VAE-based multimodal approaches for modeling and generating multimodal data. Several components of our methods are flexible and can be adapted to the use-case. For instance, the first step of our method consists of training a basic Joint Variational Autoencoder. However, many enhancements of the VAE have been proposed

	Coherence (↑)		**MFD** (↓)	
	Joint	Conditional	Joint	Conditional
JMVAE	0.57 ± 0.02	0.86 ± 0.01	$\mathbf{1.32} \pm 0.01$	0.29 ± 0.02
MMVAE	$\underline{0.63} \pm 0.01$	0.86 ± 0.01	1.63 ± 0.05	0.76 ± 0.01
MMVAE+	0.57 ± 0.01	$\underline{0.89} \pm 0.01$	1.58 ± 0.07	0.55 ± 0.08
MVTCAE	0.38 ± 0.01	0.87 ± 0.01	$\mathbf{1.31 \pm 0.02}$	$\mathbf{0.13 \pm 0.01}$
MoPoE	0.44 ± 0.02	0.74 ± 0.01	1.56 ± 0.03	2.17 ± 0.03
Nexus	0.13 ± 0.01	0.34 ± 0.01	2.98 ± 0.04	3.36 ± 0.03
JNF(Ours)	$\mathbf{0.67} \pm 0.01$	$\underline{0.89} \pm 0.01$	$\mathbf{1.32} \pm 0.02$	$\underline{0.23} \pm 0.02$
JNF-Shared(CL)(Ours)	$\mathbf{0.65} \pm 0.02$	$\mathbf{0.93} \pm 0.01$	$\underline{1.35} \pm 0.04$	$\underline{0.21} \pm 0.03$
JNF-Shared(DCCA)(Ours)	$\mathbf{0.66} \pm 0.01$	$\mathbf{0.92} \pm 0.01$	$\underline{1.37} \pm 0.04$	$\underline{0.23} \pm 0.03$

Table 2: Experimental results on the MHD dataset. We present average coherence and MFD results for each model, for conditional and unconditional generation. Best values are in bold and second-best values are underlined.

to better learn the generative parameter θ with more expressive modeling of the posterior or prior distributions([39, 40, 41]) or increased tightness of the objective bound function [15, 42]. These improvements can be used in our framework to enhance the estimation of the joint generative model on which the rest of the model depends. We also introduced the idea of learning unimodal posteriors conditioned on *a summary statistic* containing the information shared across modalities. For extracting the shared information, one can rely on Contrastive Learning or DCCA but also on other methods suited to the dataset. For instance Kernel Canonical Correlation Analysis [43] was used on functional imaging datasets [44] or genetics [45]. Finally, diffusion decoders [46] could also be used to improve the quality of generated samples as was done in [24].

Appendix A. Details on the datasets used in the experiments

Appendix A.1. The MNIST-SVHN dataset

To create this dataset, we paired images from the MNIST dataset [30] and the SVHN dataset [31]. Previous work [7] paired each image in MNIST with 30 different images in SVHN to create a train set of 1 682 040 samples. To create a more challenging and realistic dataset, we only paired each image 5 times to have a smaller (yet still large) training dataset of 280 340 samples.

Appendix A.2. PolyMNIST and Translated PolyMNIST Dataset

In Figure A.6, we plot example images of the PolyMNIST and Translated PolyMNIST dataset used in the experiments in section 4. For the Translated PolyMNIST dataset, we downscale the digit by a factor 0.75 and add a random translation. Each dataset contains 60 000 training samples and 10 000 test samples.

(a) PolyMNIST. (b) Translated PolyMNIST.

Figure A.6: Eight multimodal samples for the PolyMNIST and TranslatedPolyMNIST dataset: each row correspond to a modality.

Appendix A.3. Multimodal Handwritten Dataset

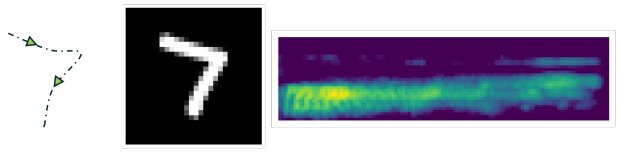


Figure A.7: The MHD dataset that we use contains three modalities.

The 'Multimodal Handwritten Digits' (MHD) introduced in [23] contains 4 modalities (including label):

- Image: gray digit images of size (1,28,28)

- Audio: spectrograms images with shape (1,32,128)
- Trajectory: flat arrays with 200 values
- Label : 10 categorical values

In our experiments, we don't use the label as a modality to make the task more challenging. This dataset contains 50 000 samples for training and 10 000 for testing.

Appendix A.4. Toy dataset with circles and squares

The images of circles and squares used in the toy experiment are of size (32,32) with black and white pixels. All circles and squares are centered in the middle of the image with a minimum width of 10 pixels and a maximum width of 28 pixels. This dataset contains 200,000 pairs of circles and squares. Half are empty and half are full.

Appendix B. Methods to learn shared information across multiple modalities

Here we detail two methods we have used to train the projectors $(g_j)_{1\leq j\leq M}$ to extract information *shared* across modalities. The projectors $(g_j)_{1\leq j\leq M}$ are trained *before* training our multimodal VAE JNF-Shared that uses them.

Appendix B.1. Deep Canonical Correlation Analysis

Deep Canonical Correlation Analysis [32] (DCCA) aims at finding correlated neural representations for two complex modalities such as images. It is based upon the classical Canonical Correlation Analysis (CCA) [43] which we briefly recall here. Let $(X_1, X_2) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ two random vectors, Σ_1, Σ_2 their covariances matrices and $\Sigma_{1,2} = \mathrm{Cov}(X_1, X_2)$. CCA's objective is to find linear projections $a^T X_1$, $b^T X_2$ that are maximally correlated :

$$(a^*, b^*) = \underset{a^T\Sigma_1 a = b^T\Sigma_2 b = 1}{\arg\max} \; a^T \Sigma_{1,2} b \,.$$

Once these optimal projections are found, we can set $(a_1, b_1) = (a^*, b^*)$ and search for subsequent projections $(a_i, b_i)_{2\leq i\leq k}$ with the additional constraint that they must be uncorrelated with the previous ones. We can rewrite the

problem of finding the first k optimal pairs of projection as finding matrices $A \in \mathbb{R}^{(n_1,k)}$, $B \in \mathbb{R}^{(n_2,k)}$ that solve:

$$(A^*, B^*) = \underset{A^T\Sigma_1 A = B^T\Sigma_2 B = I}{\arg\max} Tr(A^T \Sigma_{1,2} B) \tag{B.1}$$

If we further have $k = n_1 = n_2$ then the maximum value for $Tr(A^T\Sigma_{1,2}B)$ is $F(X_1, X_2) = Tr(T^\top T)^{\frac{1}{2}}$ with $T = \Sigma_1^{\frac{1}{2}}\Sigma_{1,2}\Sigma_2^{\frac{1}{2}}$. This value is the total CCA correlation of the random vectors X_1, X_2. It can also be seen as the sum of the singular values of T, each singular value representing the correlation of the embeddings along a direction. Note that this optimal value $F(X_1, X_2)$ only depends on the covariance matrices $(\Sigma_1, \Sigma_2, \Sigma_{1,2})$.

In the DCCA method, we consider two neural networks g_1, g_2 so as to optimize the total CCA correlation $F(g_1(X_1), g_2(X_2))$. The gradient of this objective with respect to the parameters of g_1, g_2 can be derived in order to use gradient descent.

In practice, to compute F we can use the singular value decomposition of T and sum the first k singular values of T. Furthermore the singular values are interesting since they give an information of how much correlation is contained in each projection. That information can be used to analyse the data and choose an optimal dimension k for the projection.

When considering more than two modalities, a proposed extension to the CCA is to optimize the sum of the pairwise CCA objectives [47]. We adapt this idea to the DCCA framework and train DCCA encoders for m modalities by maximizing $\sum_{i<j\in[|1,m|]} F(g_i(X_i), g_j(X_j))$.

Our implementation is based upon `https://github.com/Michaelvll/DeepCCA`.

Appendix B.2. Multimodal Contrastive Learning

Contrastive learning methods have emerged as a powerful tool to learn descriptive, transferable representations of high dimensional data such as images or text [36, 34].

In the two-modalities case, we aim at learning two embbeding functions $g_1(x_1)$, $g_2(x_2)$ that brings together "positive pairs" observed from the joint distribution $x_1, x_2 \sim p(x_1, x_2)$ and separates "negatives pairs" observed from the product of the marginal distributions $x_1, x_2 \sim p(x_1)p(x_2)$.

Formally, considering a batch of multimodal samples $(x_1^i, x_2^i)_{1\leq i\leq K}$, the loss function writes:

$$L = \sum_{i=1}^{K} L_{1,2}(i) + L_{2,1}(i) \tag{B.2}$$

$$L_{1,2}(i) = -\log\left(\frac{sim_\gamma(x_1^i, x_2^i)}{\sum_{j=1}^{K} sim_\gamma(x_1^i, x_2^j)}\right) \forall 1 \leq i \leq K \tag{B.3}$$

$$L_{2,1}(i) = -\log\left(\frac{sim_\gamma(x_2^i, x_1^i))}{\sum_{j=1}^{K} sim_\gamma(x_2^i, x_1^j)}\right) \forall 1 \leq i \leq K\,, \tag{B.4}$$

where $sim_\gamma(x_1, x_2) = \exp(\frac{1}{\tau}\frac{g_1(x_1)}{||g_1(x_1)||} \cdot \frac{g_2(x_2)}{||g_2(x_2)||})$ is the exponential cosine similarity between the embeddings,τ is a hyperparameter and γ parameterize the embedding functions g_1, g_2 that we aim to optimize. $\tau = 0.1$ in our experiments.

For any $1 \leq i \leq K$, the pair $(x_1^{(i)}, x_2^{(i)})$ is a *positive* pair which should have high similarity and the pairs $(x_1^{(i)}, x_2^{(j)})_{1 \leq j \neq i \leq K}$, $(x_1^{(j)}, x_2^{(i)})_{1 \leq j \neq i \leq K}$ are *negative* pairs that should have low similarity.

In order to bring together positive pairs in the embedding space and separate negative pairs, the projectors $(g_j)_{1 \leq j \leq M}$ have to extract the information between modalities.

For a larger number of modalities: $M \geq 2$, we can compute the sum of all pairwise losses and minimize them jointly [36].

Appendix C. Interpretations of the $\mathcal{L}_{uni}$ Objective

In this appendix, we provide several interpretations of the $\mathcal{L}_{uni}$ loss function Equation (7)that explains why minimizing it is a sensible objective to fit the unimodal posteriors. First, we reinterpret Equation (7) to show that it brings the unimodal encoder $q_{\phi_i}(z|x_i)$ (for $i \in [1, m]$) close to an average distribution $q_{\text{avg}}(z|x_i) = \mathbb{E}_{\hat{p}((x_j)_{j \neq i}|x_i)}(q_\phi(z|X))$ that is close to $p_\theta(z|x_i)$ provided that the joint encoder is well fit. Secondly, we recall an analysis from [5] that links Equation (7) to the notion of Variation of Information.

Appendix C.1. Interpretation in Relation to an Average Distribution

We recall an interpretation by [17] and extend it to a more general case.

First, let's suppose that we have only two modalities x_1, x_2 and that x_2 takes only discrete values in a set V_2. We isolate the term with $q_{\phi_2}(z|x_2)$ in Equation (7) and sum over the whole dataset D:

$$\sum_{(x_1,x_2)\in D}^{N} KL(q_\phi(z|x_1,x_2)||q_{\phi_2}(z|x_2)) = \sum_{y\in V_2}\sum_{(x_1,y)\in D} KL(q_\phi(z|x_1,y)||q_{\phi_2}(z|y)) \tag{C.1}$$

$$= \sum_{y\in V_2} KL(\sum_{(x_1,y)\in D} q_\phi(z|x_1,y)||q_{\phi_2}(z|y)) \tag{C.2}$$

Each distribution $q_\phi(z|x_1,y)$ is a gaussian with a small variance, and $q_{\phi_2}(z|y)$ is encouraged to cover this mixture of all distributions $\sum_{(x_1,y)\in D} q_\phi(z|x_1,y)$ which correspond to all parts of the latent space where a pair (x_1,y) was embedded with the joint encoder $q_\phi(z|X)$.

We now study the general case with $M \in \mathbb{N}$ and x_j not taking discrete values.

For $1 \leq j \leq M$, we isolate the term with $q_{\phi_j}(z|x_j)$ in Equation (7):

$$\mathcal{L}_{uni}(j) = KL(q_\phi(z|X)||q_{\phi_j}(z|x_j)) \tag{C.3}$$

$$= \mathbb{E}_{q_\phi(z|X)}\left(-\log(q_{\phi_j}(z|x_j))\right) - H(q_\phi(z|X)) \tag{C.4}$$

where $H(q_\phi(z|X))$ is the Shannon entropy of $q_\phi(z|X)$. Since, $q_\phi(z|X)$ is fixed while optimizing Equation x(7), this term is a constant.

When optimizing this loss over the entire dataset, we actually optimize the expectation of this term over the empirical distribution $p(X)$.

$$\mathbb{E}_{p(X)}(\mathcal{L}_{uni}(j)) = \mathbb{E}_{p(X)}\left(\mathbb{E}_{q_\phi(z|X)}\left(-\log(q_{\phi_j}(z|x_j))\right)\right) + cte \tag{C.5}$$

where cte is an additive constant term.

Furthermore, we can decompose $p(X) = p(x_j)p(X_{C_j}|x_j)$ where we note $X_{C_j} = (x_i)_{1\leq i\neq j\leq M}$ the set of modalities from which we exclude x_j.

$$\mathbb{E}_{p(X)}(\mathcal{L}_{uni}(j)) = \mathbb{E}_{p(x_j)}\left(\mathbb{E}_{p(X_{C_j}|x_j)}\left(\mathbb{E}_{q_\phi(z|X)}\left(-\log(q_{\phi_j}(z|x_j))\right)\right)\right) + cte \tag{C.6}$$

We suppose the density $q_{\phi_j}(z|x_j)$ bounded by a constant C, which allows us to use Fubini's theorem and exchange the expectations.

$$\mathbb{E}_{p(X)}(\mathcal{L}_{uni}(j)) = \mathbb{E}_{p(x_j)}\left(\mathbb{E}_{p(X_{C_j}|x_j)}\left(\mathbb{E}_{q_\phi(z|X)}\left(-\log(\frac{q_{\phi_j}(z|x_j)}{C})\right)\right)\right) - \log(C) + cte \tag{C.7}$$

$$= \mathbb{E}_{p(x_j)}\left(\int_{X_{C_j}}\int_z -\log\left(\frac{q_{\phi_j}(z|x_j)}{C}\right) q_\phi(z|X)p(X_{C_j}|x_j)dzdX_{C_j}\right) + cte \tag{C.8}$$

$$= \mathbb{E}_{p(x_j)}\left(\int_z -\log\left(\frac{q_{\phi_j}(z|x_j)}{C}\right)\int_{X_{C_j}} q_\phi(z|X)p(X_{C_j}|x_j)dX_{C_j}dz\right) + cte \tag{C.9}$$

$$= \mathbb{E}_{p(x_j)}\left(\mathbb{E}_{q_\phi^{(avg)}(z|x_j)}\left(-\log\left(\frac{q_{\phi_j}(z|x_j)}{C}\right)\right)\right) + cte \tag{C.10}$$

$$= \mathbb{E}_{p(x_j)}\left(KL\left(q_\phi^{(avg)}(z|x_j)||q_{\phi_j}(z|x_j)\right)\right) + H(q_\phi^{(avg)}(z|x_j)) + cte \tag{C.11}$$

$$\tag{C.12}$$

where $q_\phi^{(avg)}(z|x_j) \coloneqq \int_{X_{C_j}} q_\phi(z|X)p(X_{C_j}|x_j)dX_{C_j}$ and cte regroups all additive constant terms at each line. We use Fubini's theorem at line (C.10) since all terms in the integral are positive.

Since $H(q_\phi^{(avg)}(z|x_j))$ is also a constant term, we see that minimizing $\mathcal{L}_{uni}(j)$ reduces to miniming the Kullback-Leibler divergence between $q_{\phi_j}(z|x_j)$ and this average distribution $q_\phi^{(avg)}(z|x_j)$.

Appendix C.2. Interpretation in Relation to the Variation of Information

First, in the bimodal case where $M = 2$, we recall an interpretation provided by [5] that links (7) to the Variation of Information (VI) of x_1 and x_2 where x_1 (resp. x_2) represent the variable of the first modality (resp second).

Recall the definition of the VI :

$$VI(x_1, x_2) = -\mathbb{E}_{\mathbb{P}(x_1,x_2)}\big(\log\mathbb{P}(x_1|x_2) + \log\mathbb{P}(x_2|x_1)\big). \tag{C.13}$$

If we analyse Eq. (C.13), we see that the more the modalities are predictive of one another, the smaller is the Variation of Information. We do not know

the true joint and conditional distributions but we can use the following approximation summing on N training samples:

$$\widetilde{VI} = -\sum_{n=1}^{N} \log p_{\theta,\phi_1}(x_1^{(n)}|x_2^{(n)}) + \log p_{\theta,\phi_2}(x_2^{(n)}|x_1^{(n)}) \,,$$

where for $i, j \in \{1, 2\}$ with $i \neq j$, $p_{\theta,\phi_i}(x_j|x_i) := \int p_\theta(x_j|z) q_{\phi_i}(z|x_j) dz$ is our conditional generative models to sample x_j from x_i. We can show that with $\mathcal{L}$ being the ELBO defined in Eq. (3) and $\mathcal{L}_{uni}$ defined in Eq. (7):

$$-\mathcal{L}(x_1, x_2; \theta, \phi) + \mathcal{L}_{uni}(x_1, x_2; \phi) \geq \widetilde{VI} \,. \tag{C.14}$$

We recall that in our method, we first maximise $\mathcal{L}(x_1, x_2; \theta, \phi)$ and then we minimize $\mathcal{L}_{uni}(x_1, x_2; \phi)$, therefore we minimize an upper bound on $\tilde{VI}$ that is the empirical Variation of Information between modality 1 and 2. Minimizing $\tilde{VI}$ is a sensible goal as it encapsulates the predictive power of a modality given the other.

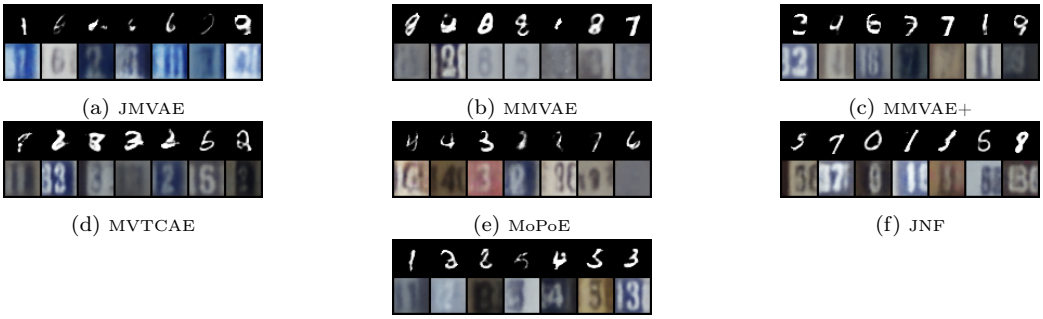
Let us now prove Equation (C.14) :

$$\begin{aligned}
\log p_{\theta,\phi_1}(x_2|x_1) + \log p_{\theta,\phi_2}(x_1|x_2) &\geq \mathbb{E}_{q_\phi(z|x_1,x_2)} \left(\log \frac{p_\theta(x_1|z) q_{\phi_2}(z|x_2)}{q_\phi(z|x_1, x_2)} \right) \\
&\quad + \mathbb{E}_{q_\phi(z|x_1,x_2)} \big(\log \frac{p_\theta(x_2|z) q_{\phi_1}(z|x_1)}{q_\phi(z|x_1, x_2)} \big) \\
&= \mathbb{E}_{q_\phi(z|x_1,x_2)} \big(\log p_\theta(x_1|z) \big) + \mathbb{E}_{q_\phi(z|x_1,x_2)} \big(\log p_\theta(x_2|z) \big) \\
&\quad - KL(q_\phi(z|x_1, x_2)||q_{\phi_2}(z|x_2)) - KL(q_\phi(|x_1, x_2)||q_{\phi_1}(z|x_1)) \\
&= \mathcal{L}(x_1, x_2) + KL(q_\phi(z|x_1, x_2)||p(z)) - \mathcal{L}_{uni}(x_1, x_2; \theta, \phi) \\
&\geq \mathcal{L}(x_1, x_2) - \mathcal{L}_{uni}(x_1, x_2; \theta, \phi) \,.
\end{aligned}$$

Appendix D. Additional experimental results

Appendix D.1. Additional results on MNIST-SVHN

In Figure D.8, we present samples generated from the prior. JNF-CL refers to our model JNF-Shared using Constrastive Learning (CL) to extract the shared information. This method performed best on this dataset, to extract the shared information.



(a) JMVAE (b) MMVAE (c) MMVAE+

(d) MVTCAE (e) MoPoE (f) JNF

(g) JNF-CL

Figure D.8: Unconditional generation: for each model, latent codes are sampled from the prior and decoded jointly.

In Table D.3, we report all coherences results for different values of the parameter β. For each model, we kept the value of β that maximises the mean coherence for the results presented in Table 1.

		Joint		$M \longrightarrow S$		$S \longrightarrow M$	
		mean	std	mean	std	mean	std
Model	β						
JMVAE	0.5	0.27	0.02	0.67	0.03	0.57	0.03
	1	0.34	0.07	0.69	0.05	0.54	0.03
	2.5	0.43	0.10	0.73	0.07	0.53	0.05
MMVAE	**0.5**	0.35	0.02	0.80	0.01	0.70	0.02
	1	0.35	0.02	0.80	0.02	0.68	0.02
	2.5	0.33	0.01	0.80	0.02	0.68	0.03
MMVAE+	0.5	0.24	0.04	0.55	0.04	0.62	0.02
	1	0.27	0.03	0.50	0.03	0.59	0.06
	2.5	0.43	0.05	0.60	0.09	0.58	0.05
MVTCAE	0.5	0.29	0.01	0.74	0.02	0.36	0.02
	1	0.35	0.02	0.75	0.05	0.44	0.02
	2.5	0.44	0.02	0.81	0.01	0.52	0.02
MoPoE	0.5	0.27	0.02	0.13	0.01	0.77	0.00
	1	0.32	0.01	0.12	0.00	0.75	0.01
	2.5	0.36	0.01	0.12	0.00	0.72	0.01
JNF	0.5	0.37	0.01	0.80	0.01	0.47	0.01
	1	0.43	0.01	0.81	0.01	0.48	0.02
	2.5	0.51	0.01	0.82	0.01	0.52	0.01
JNF-Dcca	0.5	0.36	0.02	0.76	0.01	0.71	0.02
	1	0.42	0.02	0.76	0.01	0.71	0.02
	2.5	0.51	0.01	0.75	0.03	0.69	0.05
JNF-CL	0.5	0.36	0.03	0.78	0.02	0.79	0.01
	1	0.42	0.01	0.81	0.01	0.78	0.02
	2.5	0.51	0.02	0.81	0.01	0.75	0.02

Table D.3: All coherences results for different values of β for each model. We indicate in bold, the value of β that maximises average (conditional and joint) coherence for each model and that we kept for table 1. In 1, we presented results for our model JNF-Shared using Constrastive Learning (CL). Here we present additional results with the DCCA used instead of Constrastive Learning.

Appendix D.2. Additional results on PolyMNIST

In Figure D.9 we present samples generated by conditioning on a subset of two modalities and in Figure 5 we present samples generated from the prior (unconditional generation). Our models produce diverse and coherent images, while the MoPoE and MMVAE models produce images that look "averaged" from using a mixture based aggregation [9].

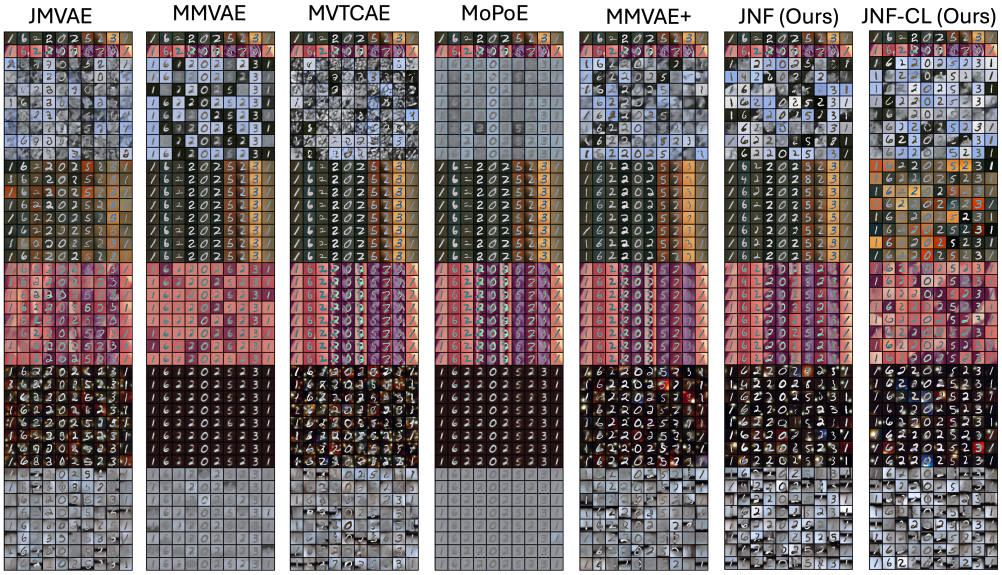


Figure D.9: We present generated samples when conditioning on the first two modalities. The first two rows are the samples we condition on and the rest of the rows are generated samples in each modality.

Appendix D.3. Additional results on Translated PolyMNIST

Figure D.10 shows examples of generated images on TranslatedPolyMNIST and in Figure D.11 we present samples generated from the prior. MMVAE and MoPoE reach a high joint coherence on this dataset but if we look at the generated images, we realize the generated images all look averaged, resembling a small "1" digit. The FID is very high since the generation is not diverse.

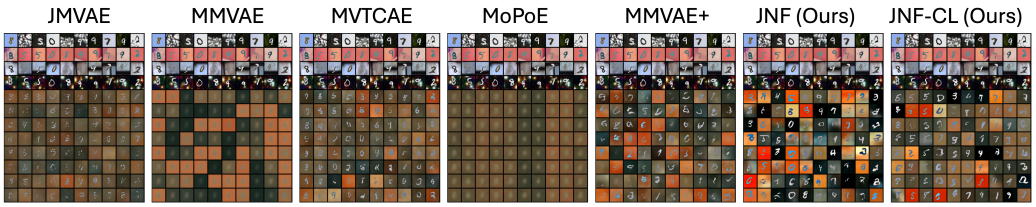
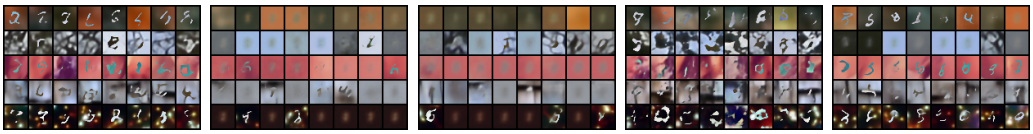
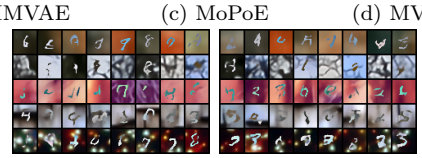


Figure D.10: Conditional generation on Translated PolyMNIST. The first four rows are the images we condition on and the newt rows are generated samples in the first modality. JNF-CL refers to our model JNF-Shared with CL.



(a) JMVAE (b) MMVAE (c) MoPoE (d) MVTCAE (e) MMVAE+



(f) JNF(Ours) (g) JNF-CL (Ours)

Figure D.11: Unconditional generation on Translated PolyMNIST when sampling a latent code from the prior.

In Table D.4, we present coherences and FID results for different values of the parameter β for each model. We used this table for selecting the value of β. For all models we observe inverse tendencies between joint and conditional coherence with the value of β. We chose to favor conditional coherence to select the best value of β for each model for the results presented in Table 4. In this table, we also test two values for the number of flows $n_{flows} \in \{2, 3\}$ for our models. *After* selecting β, we varied and selected the optimal parameter n_{flows}.

			Coherence (↑)		FID (↓)
			Joint	Conditional	1 modality to m_0
Model	β	n_{flows}			
JMVAE	**0.5**		0.00	0.15	37.06
	1.0		0.00	0.14	43.93
	2.5		0.00	0.12	55.09
MMVAE+	**0.5**		0.006	0.10	60.48
	1		0.005	0.10	69.80
	2.5		0.15	0.10	206.13
MVTCAE	**0.5**		0.004	0.13	42.35
	1		0.08	0.11	121.86
	2.5		0.23	0.11	178.49
MMVAE	**0.5**		0.63	0.10	185.97
	1.0		0.49	0.10	172.44
	2.5		0.58	0.10	181.08
MoPoE	0.5		0.26	0.10	195.53
	1.0		0.50	0.10	199.48
	2.5		0.50	0.10	199.94
JNF (Ours)	**0.5**	3	0.0004	0.17	30.91
	0.5	**2**	0.0002	0.18	31.76
	1.0	3	0.0007	0.17	33.82
	2.5	3	0.06	0.12	218.75
JNF-Shared (CL) (Ours)	**0.5**	3	0.0002	0.21	32.09
	0.5	**2**	0.0005	0.23	33.17
	1	3	0.0008	0.20	35.30
	2.5	3	0.06	0.13	217.02

Table D.4: Coherences and FID results for different values of β. Here, we average over all possible subsets for the conditional coherence. For almost all models, we observe inverse tendencies for joint and conditional coherence with the value of β. For the results presented in the main text, we chose to favor conditional generation to select the value of β for each model. The chosen β is set in bold. For the JNF-Shared (DCCA) we used the same $\beta = 0.5$ and 2 flows as for JNF-Shared (CL) because of the similarity between models.

Appendix D.4. Additional results on the MHD dataset

In Figure D.12, we display images and spectrograms obtained when conditioning on a given trajectory (that is not displayed here) drawing a zero digit. Our models generate diverse and constrasted images.

In Table D.5, we present all coherence results for different values of the parameter β for each model. We used this table to chose β for each model. For all models we observe inverse tendencies between joint and conditional coherence with the value of β. We chose to favor conditional coherence to select β for each model for the results presented in Table 2.

		Joint		Conditional	
		mean	std	mean	std
Model	β				
JMVAE	**0.5**	0.57	0.02	0.86	0.01
	1.0	0.15	0.02	0.79	0.01
	2.5	0.15	0.04	0.75	0.02
MMVAE	**0.5**	0.63	0.01	0.86	0.01
	1.0	0.60	0.07	0.84	0.04
	2.5	0.65	0.02	0.86	0.01
MMVAEPlus	**0.5**	0.58	0.03	0.89	0.02
	1.0	0.64	0.05	0.82	0.03
	2.5	0.47	0.15	0.50	0.08
MVTCAE	**0.5**	0.38	0.01	0.87	0.01
	1.0	0.48	0.01	0.85	0.00
	2.5	0.54	0.02	0.79	0.01
MoPoE	**0.5**	0.44	0.02	0.74	0.01
	1.0	0.50	0.01	0.72	0.02
	2.5	0.45	0.02	0.62	0.01
JNF	**0.5**	0.67	0.01	0.89	0.01
	1.0	0.71	0.02	0.86	0.01
	2.5	0.72	0.02	0.81	0.01
JNF-Shared (DCCA)	**0.5**	0.66	0.01	0.92	0.01
	1.0	0.71	0.02	0.90	0.01
	2.5	0.72	0.02	0.80	0.01
JNF-Shared (CL)	**0.5**	0.65	0.02	0.93	0.01

Table D.5: All coherence results for different values of β for each model. We indicate in bold, the value of β that maximises conditional coherence for each model and that we kept for Table 2. For the JNF-Shared (CL) we use the same β as for JNF-Shared (DCCA) since the model are very similar.

.

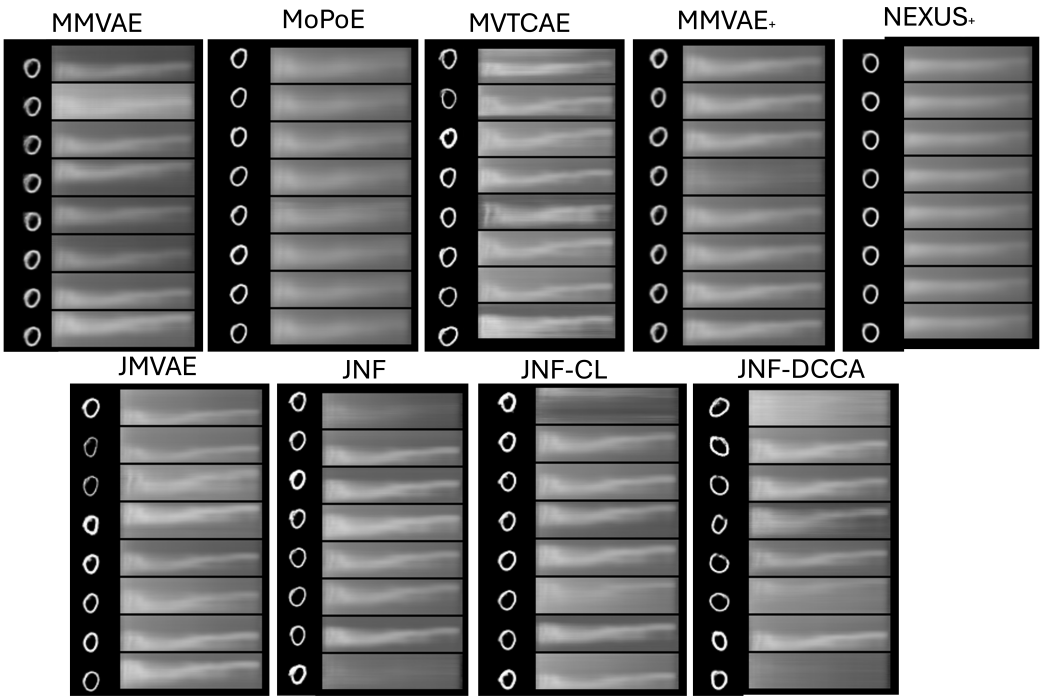


Figure D.12: Samples generated when conditioning on a given trajectory.

Appendix E. Architectures and hyperparameters used in the experiments

In Figure E.17 we summarize the general architectures of most models used in our experiments. For the Nexus model, we refer the reader to [23].

We describe in the following sections the encoders/decoders architectures for all experiments. Note that for the JNF-Shared, the projectors $(g_j)_{1\leq j\leq M}$ have the same architectures as the encoders of other models, and the encoders that parameterize $q_{\phi_j}(z|g_j(x_j))$ are simple two-layers MLPs taking the projections $(g_j)_{1\leq j\leq M}(x_j)$ as inputs.

Our implementations of Normalizing Flows rely on the opensource library Pythae [48].

Code and data needed for reproducing the experiments are available at `https://anonymous.4open.science/r/JNF_VAE/README.md`.

Appendix E.1. On MNIST-SVHN

In Table E.6, we indicate all architectures and training parameters used in the MNIST-SVHN experiments. All models are trained until convergence.

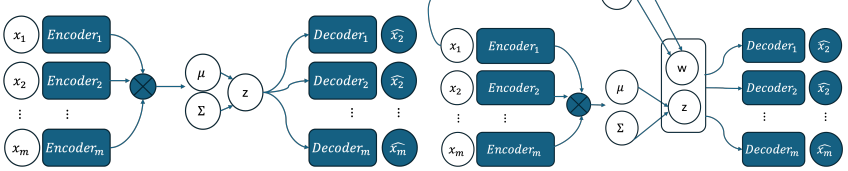


Figure E.13: Architectures of MMVAE, MoPoE and MVTCAE

Figure E.14: Architecture of MMVAE+

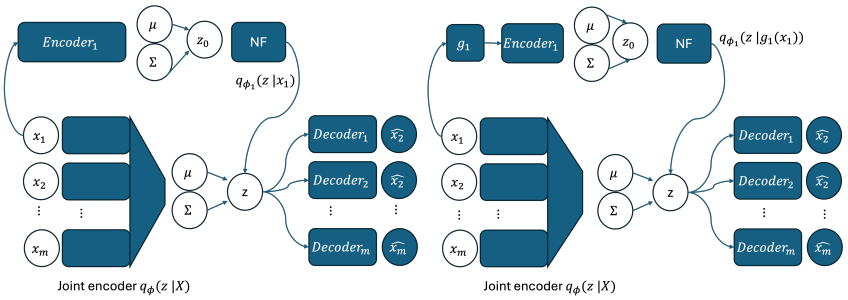


Figure E.15: Architecture of JNF and JMVAE model.

Figure E.16: Architecture of JNF-Shared

Figure E.17: Architectures for most models used. For the JMVAE model, it is the same architecture as the JNF model except without the Normalizing Flows.

For all models, we test three values for $\beta \in \{0.5, 1.0, 2.5\}$ and for each model we kept the value that maximized average coherence (joint and conditional). Extensive results for all values of β are presented in Table D.3. For the MMVAE and MMVAE+ model, we use Laplace distributions for modeling prior and posterior distribution following [7]. For all others models, we use Gaussian distributions for prior and posteriors. For the decoders distributions $p_\theta(X|z)$ we use Laplace distributions. Following previous work [7, 8] we rescale the likelihoods of each modality with factors $\lambda_{MNIST}, \lambda_{SVHN}$ in order to compensate for the different sizes of the modalities and mitigate conflictual gradients [49].The values for $\lambda_{MNIST}, \lambda_{SVHN}$ are indicated in Table E.6. Intuitively, we need to put more weight on the smaller modalities so that they are also well reconstructed.

We give specific details for each model:

- MVTCAE: we set $\alpha = 0.9$ following their recommandations in the supplemental material in [19].
- MMVAE: we set K=10 for the number of samples in the ELBO.
- MMVAE+: we set K=10 for the number of samples in the ELBO. The shared latent space as well as the modality-specific latent spaces have a dimension of 10.
- JMVAE model, we set $\alpha = 0.1$ as it appears as a good compromise value in [5]. We use annealing as in the original paper with a 100 epochs for warmup. The joint encoder is made up of separate heads and a common merging part where the separate heads have the same architecture as the unimodal encoders in Table E.6. The merging part is a simple two-layer MLP with 512 neurons in each layer.
- JNF: we used Masked Autoregressive Flows with two MADE blocks[27]. We use the same joint encoder as for the JMVAE model.
- JNF-Shared: We use the same flows and joint encoder as JNF. The projectors used for CL or DCCA have the same architectures as the encoders in Table E.6. The encoders $q_{\phi_j}(z|g_j(x_j))$ are simple networks with two linear hidden layers.

Appendix E.2. On PolyMNIST

For the PolyMNIST experiments, we used the same Resnet [50] architectures as used in [13]. These architectures are summarized in Figure E.18. Following [13], we train all models as β-VAE and set $\beta = 2.5$. Each model is trained until convergence with a batchsize of 128 and learning rate of 1e-3. The latent dimension is set to 190 to match the total capacity of the MMVAE+ model in [13].

Mnist Encoder	Mnist Decoder
Linear($1 \times 28 \times 28$,400) RELU Linear(400,20), Linear(400,20)	Linear(20, 400) RELU Linear(400, $1 \times 28 \times 28$), Sigmoid
SVHN Encoder	**SVHN Decoder**
Conv2d(3,32,4,2,1,bias=True), RELU Conv2d(32,64,4,2,1,bias=True), RELU Conv2d(64,128,4,2,1,bias=True), RELU Conv2d(128,20,4,1,0,bias=True) $\times 2$	Conv2dTranspose(d,128,4,4,1,0,bias=True), RELU Conv2dTranspose(128,64,4,4,1,0,bias=True), RELU Conv2dTranspose(64,32,4,4,1,0,bias=True), RELU Conv2dTranspose(32,3,4,4,1,0,bias=True), Sigmoid
Training parameters	**Joint Encoder for JMVAE, JNF, JNF-Shared**
Batchsize = 128 Learning rate = 1e-3 Optimizer = Adam Latent dimension = 20 $\lambda_{MNIST}, \lambda_{SVHN} = \frac{3\times32\times32}{28\times28}, 1$ Dimension for the projection (g_j) = 10	Separates head for each modality Linear(512), RELU Linear(512), RELU Linear(20), Linear(20)
Normaling Flows	**JNF-Shared encoders for $q_{\phi_j}(z\|g_j(x_j))$**
Masked Autoregressive with two MADE blocks	Linear(10,512) RELU Linear(512,512) RELU Linear(512,20), Linear(512,20)

Table E.6: Architectures and training parameters used for the Mnist-SVHN Experiments.

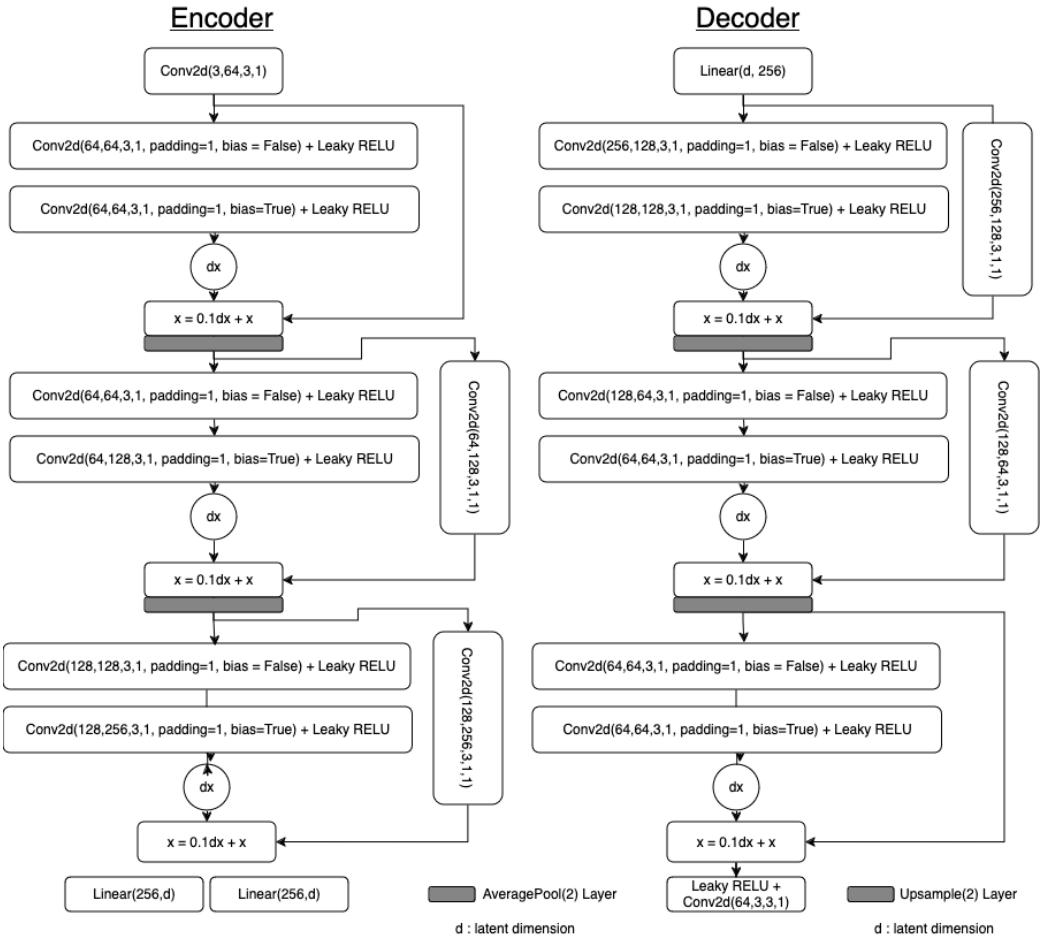


Figure E.18: Encoder and decoder architectures used for the experiments on the PolyMNIST dataset.

We give specific details for each model:

- MMVAE: Due to memory limitations, we set the latent dim to 64 and used K=10 for the number of samples in the ELBO.
- MVTCAE: we set $\alpha = \frac{5}{6}$.
- JMVAE: we set $\alpha = 0.1$ and annealing with a warmup of 100 epochs. In the original JMVAE model, a new encoder network needs to be introduced for each subset of modalities. In our experiments, we didn't choose that solution since it represents a very large number of parameters. Instead, we use for the JMVAE model, the PoE sampling solution

that we also use for our models (Equation (12)). The joint encoder is made-up of separate heads with the same architectures as in Figure E.18 and a merging neural networks with two hidden linear layers of 512 neurons.

- MMVAE+: We use 32 dimensions for the shared latent space and 32 dimension for each modality specific space as in [13].
- JNF: Same joint encoder as JMVAE. We use Masked Autoregressive flows with 2 MADE blocks.
- JNF-Shared: Same joint encoder and Normalizing flows as JNF. The projectors (g_j) are simple convolutional networks similar to the SVHN encoders in E.6 and the encoders $q_{\phi_j}(z|g_j(x_j))$ are simple linear encoders as for the MNIST-SVHN experiments: see Table E.6.

Appendix E.3. On Translated PolyMNIST

For the TranslatedPolyMNIST experiments, we used similar architectures as in the PolyMNIST experiments with a latent dimension of 200 (except for MMVAE and MMVAE+ whose parameters are specified below). We performed experiments with $\beta \in \{0.5, 1., 2.5\}$. For all models, we kept the value of β that maximized average conditional coherence. In Table D.4, we present results for different values of β and the selected values for each model. We use a latent dimension of 200 for all models but the MMVAE+ that has multiple latent spaces (see below). All models are trained until convergence with learning rate 1e-3 and batchsize 128.

We give specific details for each model:

- MMVAE: Due to memory limitations, we used a latent dimension of 100 for the MMVAE model and used K=10 for the number of samples in the ELBO.
- MVTCAE: we set $\alpha = \frac{5}{6}$ as in PolyMNIST.
- JMVAE: we set $\alpha = 0.1$ and a warmup of 100 epochs. PoE sampling is applied for JMVAE as in the other experiments. The joint encoder is made of separate heads with the same architectures as in Figure E.18 and we concatenate the outputs of each head to form the joint representation. This concatenation instead of a merging network allows to avoid conflictual gradient issues and modality collapse [49].

- MMVAE+: We use 32 dimensions for the shared latent space and 32 dimension for each modality specific space as in [13]. We use K=10 for the number of samples in the ELBO.

- JNF: Same joint encoder as JMVAE. We use Masked Autoregressive flows with 3 MADE blocks.

- JNF-Shared (CL): Same joint encoder and Normalizing flows as JNF. The projectors (g_j) have the encoder architectures in Figure E.18 and the encoders $q_{\phi_j}(z|g_j(x_j))$ are simple linear networks as for the MNIST-SVHN experiments: see Table E.6. We use Masked Autoregressive flows with 2 MADE blocks.

- JNF-Shared (DCCA): when using DCCA to extract the shared information, we used more simple architectures for the projectors (g_j) for instability reasons. We used simple convolutional networks similar to the SVHN encoders in Table E.6. Precise architectures are given in the code. We use Masked Autoregressive flows with 2 MADE blocks.

Appendix E.4. On MHD

Table E.7 contains all relevant architectures and general training parameters.

We use the same architectures than the ones used in [23] except that we don't pretrain the sound encoder and decoder. All models with a β term weighing the Kullback-Leibler divergence in (3) and for all models, the $\beta = 0.5$ gives the best average conditional coherence. We present additional results for all values of β in Table D.5. We used Gaussian distributions to model all posterior, prior and decoding distributions. We use a latent dimension of 64 for all models but the MMVAE+ that has multiple latent spaces (see below).

We use rescaling for the likelihoods of each modality following [7]. It has been shown that this limits the phenomenons of conflictual gradients and modality collapse [49]. The rescaling factors $\lambda_{image}, \lambda_{audio}, \lambda_{trajectory}$ are given in Table E.7

We train all models until convergence. We give specific details for each model:

- MMVAE: We used K=10 for the number of samples in the ELBO.

- MVTCAE: we tried $\alpha \in \{0.75, 0.9\}$ and kept best results obtained for $\alpha = 0.9$.
- JMVAE: we set $\alpha = 0.1$ and a warmup of 100 epochs. In the original JMVAE model, a new encoder network needs to be introduced for each subset of modalities. In our experiments, we didn't choose that solution since it represents a very large number of parameters. Instead, we use for the JMVAE model, the PoE sampling solution that we also use for our models (Equation (12)). The joint encoder is made-up of separate heads with the same architectures as in Table E.7 and a merging neural networks with two hidden linear layers of 512 neurons.
- MMVAE+: We use 32 dimensions for the shared latent space and 32 dimension for each modality specific space. We used K=10 for the number of samples in the ELBO.
- JNF: Same joint encoder as JMVAE. We use Masked Autoregressive flows with 2 MADE blocks.
- JNF-Shared: Same joint encoder and Normalizing flows as JNF. The projectors (g_j) have the encoder architectures in Figure E.18 and the encoders $q_{\phi_j}(z|g_j(x_j))$ have the same architectures as for the MNIST-SVHN experiments: see Table E.6.
- NEXUS : we use the same hyperparameters as used in [23].

Image Encoder	Image Decoder
Conv2d(1,128,4,2), Batchnorm, RELU Conv2d(128,256,4,2), Batchnorm, RELU Conv2d(256,512,4,2), Batchnorm, RELU Conv2d(512,1024,4,2), Batchnorm, RELU Linear(d), Linear(1024,d)	Linear(1024*4*4) ConvTranspose2d(1024,512,3,2,padding = 1), Batchnorm,RELU ConvTranspose2d(512,256,3,2,padding = 1),Batchnorm, RELU ConvTranspose2d(256,1,3,2,padding = 1),Sigmoid
Trajectory Encoder	**Trajectory Decoder**
Linear(512), Batchnorm, LeakyRELU Linear(512), Batchnorm, LeakyRELU Linear(512), Batchnorm, LeakyRELU Linear(d), Linear(d)	Linear(512), Batchnorm, LeakyRELU Linear(512), Batchnorm, LeakyRELU Linear(512), Batchnorm, LeakyRELU Linear(128), Sigmoid
Sound Encoder	**Sound Decoder**
Conv2d(1,128, kernel = (1,128), stride = (1,1)), Batchnorm, RELU Conv2d(128,128, kernel=(4,1), stride = (2,1)), Batchnorm, RELU Conv2d(128,256, kernel=(4,1),stride = (2,1)), Batchnorm, RELU Linear(d), Linear(d)	Linear(2048), Batchnorm, RELU ConvTranspose2d(256, 128, kernel=(4,1), stride=(2,1), padding=(1,0)) ConvTranspose2d(128, 128, kernel=(4,1), stride=(2,1), padding=(1,0)) ConvTranspose2d(128, 1, kernel=(1,128), stride=(1,1), padding=0), Sigmoid
Training parameters	**Joint Encoder**
Batchsize = 64 Learning rate = 1e-3 Optimizer = Adam $\lambda_{image} = \frac{32\times128}{3\times28\times28} \approx 1.7$ $\lambda_{audio} = 1.0$ $\lambda_{trajectory} = \frac{32\times128}{200} = 20.48$	Separates head for each modality with same architectures as encoders Linear(512), RELU Linear(512), RELU Linear(d), Linear(d)
Normalizing Flows	**JNF-Shared encoders for $q_{\phi_j}(z\|g_j(x_j))$**
Masked Autoregressive with two MADE blocks	Linear(10,512) RELU Linear(512,512) RELU Linear(512,20), Linear(512,20)

Table E.7: Architectures and training parameters used for the MHD Experiments.

Appendix F. Hamiltonian Monte Carlo Sampling

In this appendix, we recall the principles of Hamiltonian Monte Carlo Sampling and detail how we apply it in our model. The Hamiltonian Monte Carlo (HMC) sampling belongs to the larger class of Markov Chain Monte Carlo methods (MCMC) that allow to sample from any distribution $f(z)$ known up to a constant [28]. The general principle is to build a Markov Chain that will have our target $f(z)$ as stationary distribution. More specifically, the HMC is an instance of the Metropolis-Hasting Algorithm (see 1) that uses a physics-oriented proposal distribution.

Algorithm 1 Metropolis-Hasting Algorithm

1: **Initialization :** $z \leftarrow z_0$
2: **for** $i \coloneqq 0 \rightarrow \mathrm{N}$ **do**
3: Sample z' from the proposal $g(z'|z)$
4: With probability $\alpha(z', z)$ accept the proposal $z \leftarrow z'$
5: **end for**

Sampling from the proposal distribution $g(z'|z_0)$ is done by integrating the Hamiltonian equations :

$$\begin{cases} \dfrac{\partial z}{\partial t} = \dfrac{\partial H}{\partial v}\,, \\ \dfrac{\partial v}{\partial t} = -\dfrac{\partial H}{\partial z}\,, \\ z(0) = z_0 \\ v(0) = v_0 \sim \mathcal{N}(0, I)\,, \end{cases} \tag{F.1}$$

where the Hamiltonian is defined by $H(z, v) = -\log f(z) + \frac{1}{2} v^t v$. In physics, Eq. (F.1) describes the evolution in time of a physical particle with initial position z and a random initial momentum v. The leap-frog numerical scheme is used to integrate Eq. (F.1) and is repeated l times with a small integrator step size ϵ :

$$\begin{aligned} v(t + \frac{\epsilon}{2}) &= v(t) + \frac{\epsilon}{2} \cdot \nabla_z (\log f(z)(t))\,, \\ z(t + \epsilon) &= z(t) + \epsilon \cdot v(t + \frac{\epsilon}{2})\,, \\ v(t + \epsilon) &= v(t + \frac{\epsilon}{2}) + \frac{\epsilon}{2} \nabla_z \log f(z(t + \epsilon))\,. \end{aligned} \tag{F.2}$$

After l integration steps, we obtain the proposal position $z' = z(t + l \cdot \epsilon)$ that corresponds to step 3 in Algorithm 1. The acceptance ratio is then defined as $\alpha(z', z_0) = \min\left(1, \frac{\exp(-H(z_0, v_0))}{\exp(-H(z', v(t+l\cdot\epsilon)))}\right)$. This procedure is repeated to produce an ergodic Markov chain (z^n) converging to the target distribution f [51, 52, 53, 54]. In this work, we use HMC sampling to sample from the PoE of unimodal posteriors in Eq. (12). To do so we need to compute and derivate the (log) of the target distribution given by the PoE of the unimodal distributions:

$$\log q(z|(x_i)_{i\in S}) = -\log p(z) + \sum_{i\in S} \log q_{\phi_i}(z|x_i)\,. \tag{F.3}$$

We can use autograd to automatically compute the gradient $\nabla_z \log q(z|(x_i)_{i\in S})$ that is needed in the leapfrog steps.

In our experiments, we use 100 steps per sampling.

Appendix G. Information on the classifiers used for evaluation

Appendix G.1. MNIST-SVHN

In Table G.8 we provide the architectures and the accuracies for the classifiers that we use to evaluate coherence on the MNIST-SVHN dataset.

SVHN	**MNIST**
Conv2d(3,10,5)	Conv2d(1,10,5)
MaxPool2d,RELU	MaxPool2d,RELU
Conv2d(10,20,5), Dropout(0.5)	Conv2d(10,20,5), Dropout(0.5)
MaxPool2d,RELU	MaxPool2d,RELU
Linear(500,50), RELU, Dropout	Linear(350,50), RELU, Dropout
Linear(50,10), Softmax	Linear(50,10), Softmax
Accuracies on test	
0.87	0.99

Table G.8: Classifiers used for the MNIST-SVHN experiments.

Appendix G.2. Classifiers on PolyMNIST

We use the architectures and the pretrained models available at `https://github.com/thomassutter/MoPoE` [8].

The accuracies of the classifiers for the five modalities of the test set are respectively: 0.95, 0.99, 0.99, 0.97, 0.95.

Appendix G.3. Classifiers on TranslatedPolyMNIST

We pretrain classifiers on this dataset having similar architectures as in Figure E.18 with a output size of 10.

The accuracies of the trained classifiers for the five modalities of the test set are respectively: 0.98, 0.97, 0.98, 0.97, 0.98.

Appendix G.4. Classifiers on MHD

We use the pretrained classifiers available at `https://github.com/miguelsvasco/nexus_pytorch/`.

The accuracies of the trained classifiers on the test set are: 0.95 for the audio modality, 0.99 for the image modality and 0.99 for the trajectory modality.

References

[1] D. P. Kingma, M. Welling, Auto-Encoding Variational Bayes, arXiv:1312.6114 [cs, stat] (May 2014).
URL `http://arxiv.org/abs/1312.6114`

[2] I. Higgins, N. Sonnerat, L. Matthey, A. Pal, C. P. Burgess, M. Bosnjak, M. Shanahan, M. Botvinick, D. Hassabis, A. Lerchner, SCAN: Learning Hierarchical Compositional Visual Concepts (Jun. 2018). `doi:10.48550/arXiv.1707.03389`.
URL `http://arxiv.org/abs/1707.03389`

[3] Y. Tian, D. Krishnan, P. Isola, Contrastive Multiview Coding, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2020, pp. 776–794. `doi:10.1007/978-3-030-58621-8_45`.

[4] H. Yin, F. Melo, A. Billard, A. Paiva, Associate Latent Encodings in Learning from Demonstrations, Proceedings of the AAAI Conference on Artificial Intelligence 31 (1), number: 1 (Feb. 2017). `doi:10.1609/aaai.v31i1.11040`.
URL `https://ojs.aaai.org/index.php/AAAI/article/view/11040`

[5] M. Suzuki, K. Nakayama, Y. Matsuo, Joint Multimodal Learning with Deep Generative Models, arXiv:1611.01891 [cs, stat]ArXiv: 1611.01891

(Nov. 2016).
URL http://arxiv.org/abs/1611.01891

[6] M. Wu, N. Goodman, Multimodal Generative Models for Scalable Weakly-Supervised Learning, in: Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc., 2018.
URL https://proceedings.neurips.cc/paper/2018/hash/1102a326d5f7c9e04fc3c89d0ede88c9-Abstract.html

[7] Y. Shi, N. Siddharth, B. Paige, P. H. S. Torr, Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models, arXiv:1911.03393 [cs, stat]ArXiv: 1911.03393 (Nov. 2019).
URL http://arxiv.org/abs/1911.03393

[8] T. M. Sutter, I. Daunhawer, J. E. Vogt, Generalized Multimodal ELBO, ICLR (2021).

[9] I. Daunhawer, T. M. Sutter, K. Chin-Cheong, E. Palumbo, J. E. Vogt, On the limitations of multimodal VAEs, in: International Conference on Learning Representations, 2022.
URL https://openreview.net/forum?id=w-CPUXXrAj

[10] Y. Shi, B. Paige, P. Torr, S. N, Relating by contrasting: A data-efficient framework for multimodal generative models, in: International Conference on Learning Representations, 2021.
URL https://openreview.net/forum?id=vhKe9UFbrJo

[11] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, L. K. Saul, An Introduction to Variational Methods for Graphical Models, in: M. I. Jordan (Ed.), Learning in Graphical Models, Springer Netherlands, Dordrecht, 1998, pp. 105–161. doi:10.1007/978-94-011-5014-9_5.
URL http://link.springer.com/10.1007/978-94-011-5014-9_5

[12] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, B. Schölkopf, From Variational to Deterministic Autoencoders, arXiv:1903.12436 [cs, stat] (May 2020).
URL http://arxiv.org/abs/1903.12436

[13] E. Palumbo, I. Daunhawer, J. E. Vogt, MMVAE+: Enhancing the generative quality of multimodal VAEs without compromises, in: The

Eleventh International Conference on Learning Representations, 2023.
URL https://openreview.net/forum?id=sdQGxouELX

[14] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, 2017.

[15] Y. Burda, R. Grosse, R. Salakhutdinov, Importance Weighted Autoencoders, arXiv:1509.00519 [cs, stat] (Nov. 2016). doi:10.48550/arXiv.1509.00519.
URL http://arxiv.org/abs/1509.00519

[16] T. Sutter, I. Daunhawer, J. Vogt, Multimodal Generative Learning Utilizing Jensen-Shannon-Divergence, in: Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 6100–6110.
URL https://proceedings.neurips.cc/paper/2020/hash/43bb733c1b62a5e374c63cb22fa457b4-Abstract.html

[17] R. Vedantam, I. Fischer, J. Huang, K. Murphy, Generative Models of Visually Grounded Imagination, arXiv:1705.10762 [cs, stat]ArXiv: 1705.10762 (Nov. 2018).
URL http://arxiv.org/abs/1705.10762

[18] A. Lawry Aguila, J. Chapman, A. Altmann, Multi-modal variational autoencoders for normative modelling across multiple imaging modalities, in: H. Greenspan, A. Madabhushi, P. Mousavi, S. Salcudean, J. Duncan, T. Syeda-Mahmood, R. Taylor (Eds.), Medical Image Computing and Computer Assisted Intervention – MICCAI 2023, Springer Nature Switzerland, Cham, 2023, pp. 425–434.

[19] H. Hwang, G.-H. Kim, S. Hong, K.-E. Kim, Multi-View Representation Learning via Total Correlation Objective, in: Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., 2021, pp. 12194–12207.
URL https://proceedings.neurips.cc/paper/2021/hash/65a99bb7a3115fdede20da98b08a370f-Abstract.html

[20] M. Suzuki, Y. Matsuo, Mitigating the limitations of multimodal VAEs with coordination-based approach (2023).
URL https://openreview.net/forum?id=Rn8u4MYgeNJ

[21] M. Lee, V. Pavlovic, Private-shared disentangled multimodal vae for learning of latent representations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2021, pp. 1692–1700.

[22] I. Daunhawer, T. M. Sutter, R. Marcinkevičs, J. E. Vogt, Self-supervised Disentanglement of Modality-Specific and Shared Factors Improves Multimodal Generative Models, in: Z. Akata, A. Geiger, T. Sattler (Eds.), Pattern Recognition, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2021, pp. 459–473. doi:10.1007/978-3-030-71278-5_33.

[23] M. Vasco, H. Yin, F. S. Melo, A. Paiva, Leveraging hierarchy in multimodal generative models for effective cross-modality inference, Neural Networks 146 (2022) 238–255. doi:10.1016/j.neunet.2021.11.019.
URL https://linkinghub.elsevier.com/retrieve/pii/S0893608021004470

[24] E. Palumbo, L. Manduchi, S. Laguna, D. Chopard, J. E. Vogt, Deep generative clustering with multimodal diffusion variational autoencoders, in: The Twelfth International Conference on Learning Representations, 2024.
URL https://openreview.net/forum?id=k5THrhXDV3

[25] M. Bounoua, G. Franzese, P. Michiardi, Multi-modal latent diffusion, Entropy 26 (4) (2024). doi:10.3390/e26040320.
URL https://www.mdpi.com/1099-4300/26/4/320

[26] D. Rezende, S. Mohamed, Variational inference with normalizing flows, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, Vol. 37 of Proceedings of Machine Learning Research, PMLR, Lille, France, 2015, pp. 1530–1538.

[27] G. Papamakarios, T. Pavlakou, I. Murray, Masked Autoregressive Flow for Density Estimation, in: Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017.
URL https://proceedings.neurips.cc/paper/2017/hash/6c1da886822c67822bcf3679d04369fa-Abstract.html

[28] MCMC Using Hamiltonian Dynamics, Chapman and Hall/CRC, 2011, pages: 139-188 Publication Title: Handbook of Markov Chain Monte Carlo. doi:10.1201/b10905-10.
URL https://www.taylorfrancis.com/chapters/edit/10.1201/b10905-10/mcmc-using-hamiltonian-dynamics-radford-neal

[29] M. Betancourt, A conceptual introduction to hamiltonian monte carlo (2018). arXiv:1701.02434.
URL https://arxiv.org/abs/1701.02434

[30] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324, conference Name: Proceedings of the IEEE. doi:10.1109/5.726791.

[31] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Ng, Reading Digits in Natural Images with Unsupervised Feature Learning, NIPS (Jan. 2011).

[32] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep Canonical Correlation Analysis, in: Proceedings of the 30th International Conference on Machine Learning, PMLR, 2013, pp. 1247–1255, iSSN: 1938-7228.
URL https://proceedings.mlr.press/v28/andrew13.html

[33] P. Poklukar, M. Vasco, H. Yin, F. S. Melo, A. Paiva, D. Kragic, Geometric Multimodal Contrastive Representation Learning, in: Proceedings of the 39th International Conference on Machine Learning, PMLR, 2022, pp. 17782–17800, iSSN: 2640-3498.
URL https://proceedings.mlr.press/v162/poklukar22a.html

[34] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, CoRR abs/2103.00020 (2021). arXiv:2103.00020.
URL https://arxiv.org/abs/2103.00020

[35] A. Abid, J. Zou, Contrastive Variational Autoencoder Enhances Salient Features, arXiv:1902.04601 [cs, stat] (Feb. 2019).
URL http://arxiv.org/abs/1902.04601

[36] Y. Tian, D. Krishnan, P. Isola, Contrastive multiview coding, CoRR abs/1906.05849 (2019). arXiv:1906.05849.
URL http://arxiv.org/abs/1906.05849

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, in: Advances in Neural Information Processing Systems, 2017.

[38] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, B. Schölkopf, From variational to deterministic autoencoders (2020). arXiv:1903.12436.
URL https://arxiv.org/abs/1903.12436

[39] D. J. Rezende, S. Mohamed, Variational inference with normalizing flows (2016). arXiv:1505.05770.
URL https://arxiv.org/abs/1505.05770

[40] J. Tomczak, M. Welling, Vae with a vampprior, in: A. Storkey, F. Perez-Cruz (Eds.), Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, Vol. 84 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 1214–1223.
URL https://proceedings.mlr.press/v84/tomczak18a.html

[41] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, M. Welling, Improving Variational Inference with Inverse Autoregressive Flow, number: arXiv:1606.04934 arXiv:1606.04934 [cs, stat] (Jan. 2017).
URL http://arxiv.org/abs/1606.04934

[42] G. Tucker, D. Lawson, S. Gu, C. J. Maddison, Doubly reparameterized gradient estimators for monte carlo objectives (2018). arXiv:1810.04152.
URL https://arxiv.org/abs/1810.04152

[43] D. R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: an overview with application to learning methods, Neural Computation 16 (12) (2004) 2639–2664. doi:10.1162/0899766042321814.

[44] N. Y. Bilenko, J. L. Gallant, Pyrcca: Regularized Kernel Canonical Correlation Analysis in Python and Its Applications to Neuroimaging, Frontiers in Neuroinformatics 10 (2016). doi:10.3389/fninf.2016.00049.

URL https://www.frontiersin.org/journals/neuroinformatics/articles/10.3389/fninf.2016.00049

[45] M. A. Alam, V. D. Calhoun, Y.-P. Wang, Identifying outliers using multiple kernel canonical correlation analysis with application to imaging genetics, Computational Statistics & Data Analysis 125 (2018) 70–85. doi:https://doi.org/10.1016/j.csda.2018.03.013.
URL https://www.sciencedirect.com/science/article/pii/S0167947318300732

[46] K. Preechakul, N. Chatthee, S. Wizadwongsa, S. Suwajanakorn, Diffusion autoencoders: Toward a meaningful and decodable representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10619–10629.

[47] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, M. Hong, Structured sumcor multiview canonical correlation analysis for large-scale data, IEEE Transactions on Signal Processing 67 (2) (2019) 306–319. doi:10.1109/TSP.2018.2878544.

[48] C. Chadebec, L. J. Vincent, S. Allassonniere, Pythae: Unifying Generative Autoencoders in Python - A Benchmarking Use Case, 2022.
URL https://openreview.net/forum?id=w7VPQWgnn3s

[49] A. Javaloy, M. Meghdadi, I. Valera, Mitigating Modality Collapse in Multimodal VAEs via Impartial Optimization, arXiv:2206.04496 [cs] (Jun. 2022).
URL http://arxiv.org/abs/2206.04496

[50] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 770–778.
URL https://api.semanticscholar.org/CorpusID:206594692

[51] S. Duane, A. D. Kennedy, B. J. Pendleton, D. Roweth, Hybrid Monte Carlo, Physics Letters B 195 (2) (1987) 216–222. doi:https://doi.org/10.1016/0370-2693(87)91197-X.
URL https://www.sciencedirect.com/science/article/pii/037026938791197X

[52] J. Liu, Monte Carlo Strategies in Scientic Computing, 2009. doi:10.1007/978-0-387-76371-2.

[53] S. Brooks, A. Gelman, G. Jones, X.-L. Meng, Handbook of Markov Chain Monte Carlo, Chapman and Hall/CRC, 2011. doi:10.1201/b10905.
URL http://dx.doi.org/10.1201/b10905

[54] M. Girolami, B. Calderhead, Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods, Journal of the Royal Statistical Society Series B: Statistical Methodology 73 (2) (2011) 123–214, _eprint: https://academic.oup.com/jrsssb/article-pdf/73/2/123/49162769/jrsssb_73_2_123.pdf. doi:10.1111/j.1467-9868.2010.00765.x.
URL https://doi.org/10.1111/j.1467-9868.2010.00765.x