Non-linear interpolation learning for example-based inverse problem regularization

Jerome Bobin, R Carloni Gertosio, C Bobin, C Thiam

Non-linear interpolation learning for example-based inverse problem regularization

J. Bobin, R. Carloni Gertosio, C. Bobin and C. Thiam .

***Abstract*—A large number of signal recovery problems are not well-posed -if not ill-posed- that require extra regularization to be tackled. In this context, the ability to inject physical knowledge is of utmost importance to design effective regularization schemes. However, most physically relevant models are generally non-linear: signals generally lie on an unknown low-dimensional manifolds structure, which needs to be learnt. This is however quite challenging when available training samples are scarce. To that end, we investigate a novel approach that builds upon learning a non-linear interpolatory scheme from examples. We show how the proposed approach resonates with transport-based methods, but with a learnt metric. This eventually allows to build efficient non-linear regularizations for linear inverse problems. Extensive numerical experiments have been carried out to evaluate the performances of the proposed approach. We further illustrate its application to a blind regression problem in the field of γ-ray spectroscopy.**

I. Introduction

A. Context

Inverse problems are commonplace in a very wide span of applications, from neuroscience to astrophysics or geophysics [1]–[4]. A major challenge, and a hot topic in the field of signal and image processing, is to find out how to design regularization scheme that allow to accurately inject physical prior knowledge about the signal to be retrieved. This is essential to provide robust and effective signal recovery algorithm, as well as physically meaningful and interpretable solutions. To that end, we hereafter investigate a new framework to build physics-driven regularizations for multilinear inverse problems (*i.e.* linear with respect to blocks of variables). More formally, we address the following type of linear inverse problems:

$$\mathbf{y} = \mathcal{F}(\Omega) + \mathbf{n}, \tag{1}$$

where $\mathbf{y}$ is a measurement vector of size t and $\mathbf{n}$ stands for some noise. The term $\mathcal{F}(\Omega)$ is a general linear or multilinear function with parameter Ω, which can actually represent a wide variety of observation models in signal and image processing, such as (but not limited to):

- **Linear model :** $\Omega = \mathbf{x}$ and $\mathcal{F}(\Omega) = \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ is a linear operator, as in standard linear inverse problems (*e.g.* denoising, deconvolution, inpainting, etc. to only name a few - [2]).
- **Multilinear model :** $\Omega = \{A, \mathbf{x}\}$ and $\mathcal{F}(\Omega) = \mathbf{A}\mathbf{x}$, as in blind deconvolution or calibration problems [5], [6].

Assuming the noise $\mathbf{n}$ has entries that are identically and independently drawn from a Gaussian distribution, it is customary to estimate the parameters Ω with a regularized least-squares estimator of the following form:

$$\min_{\Omega} \mathcal{R}(\Omega) + \|\mathbf{y} - \mathcal{F}(\Omega)\|_2^2, \tag{2}$$

where $\mathcal{R}(\Omega)$ stands for a regularization term that enforces some desired properties on Ω. In a very large number of applications (*e.g.* deconvolution or blind calibration problems to only cite two), the above inverse problem is ill-conditioned, if not ill-posed at all, admitting an infinite number of solutions. The role played by $\mathcal{R}$ is therefore central to precisely favor some solution Ω with meaningful structures. From a general viewpoint, high-dimensional data can be assumed to belong to a low-dimensional, nonlinear manifold as stated by the manifold hypothesis [7]. Consequently, in physical applications, physically relevant signals Ω *generally belong to some unknown low-dimensional manifold*. It is therefore of paramount importance to carefully account for this property to design high accuracy estimators. When the underlying manifold is unknown, this property somehow needs to be learnt.

B. The art of learning regularizations

During the last few years, various machine learning approaches have been proposed to tackle inverse problems [3]. As such, end-to-end machine learning solutions have been introduced that allow to directly estimate the sought-after parameters Ω from observations y (*e.g.* deep generative models [8]). However, one of the main disadvantage of such methods is that they generally ignore the underlying analytic description of the forward model (*e.g.* convolutive model, linear mixture model, precise statistical noise modelling, etc.), which largely limits their accuracy. Precisely accounting for the exact forward model and the noise statistics is central in scientific applications. As a consequence, we rather advocate the use of hybrid approaches, which combine standard variational approaches for inverse problems [9] and prior model or more generally regularization learning. In this spirit, various paths have been explored to put together traditional variational inference with regularization learning. For instance, it has been proposed to substitute standard numerical solvers, such as proximal algorithms, with recurrent neural networks that mimic them [10]–[13]. On top of allowing for faster solvers, such architectures

J. Bobin and R. Carloni Gertosio were with IRFU, CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France. C. Bobin and C. Thiam are with Université Paris-Saclay, CEA, List, Laboratoire National Henri Becquerel (LNE-LNHB), F-91120 Palaiseau, France. This work is supported by the European Community through the grant LENA (ERC StG - contract no. 678282). JB would like to thank Morgan Schmitz for having provided the codes that compute Wasserstein barycenters. The IAE code will be available at https://github.com/jbobin/IAE. E-mail: name.lastname@cea.fr.

virtually allow to learn the regularization. So far, it is unclear whether this type of approach is well-suited for general multilinear problems. From a different viewpoint, solving inverse problems can be performed by first providing a fast, but generally rough, inversion of the forward model, which is further cleaned from the inversion artifacts by making use of learnt denoisers. This category of methods includes learning approaches as different as traditional convolutional networks [14]–[18] as well as generative models with adversarial learning [19]. Deep representation learning [20] can also enter this class of technique when additional constraints, such as sparsity, are imposed. However, the efficiency of the learnt denoisers is strongly dependent on the forward model to be inverted. Moreover, the success of the state-of-the-art, learning-based solutions to inverse problems largely relies on the availability of large training sets. Unfortunately, whether based on physical simulations or measurements, available samples for training are generally of limited number: this further leaves unresolved the learning of physics-informed signal models when training samples are scarce.

C. Designing regularizations from examples, as a transportation problem

In this article, our goal is to build regularizations or constraints to provide efficient and yet interpretable solutions to inverse problems. For that purpose, capturing the properties of the manifold on which meaningful signals lie is central. In contrast to attempting to learn the manifold structure straight from the data, an interesting approach proposed in [21] consists in estimating how to travel on the unknown manifold. The gist of this method is to learn a transport operator from pairs of samples. This has recently been combined with a variational autoencoder architecture to build a generative model allowing to learn a latent space with a topology with increased interpretability. Inspired by this seminal work, we rather build upon the availablity of examples, whether derived from simulations or measurements in a large number of physical applications, which can be exploited to learn an efficient generative model from scarce training samples. In this context, the first ingredient of the proposed method is to build some regularization by *exploring the manifold $\mathcal{M}$ from these examples*, which we coin anchor points, rather than trying to capture the overall manifold structure. This will be conveniently carried out by building a new sample on $\mathcal{M}$ as an interpolation or barycenter of the anchor points, according to a given metric. Learning the manifold structure then boils down to learning a metric that is adapted so as to compute barycenters that actually belong to $\mathcal{M}$.

More formally, we assume that we have access to d anchor points $\{\varphi_a\}_{a=1,\dots,d}$. For now, each one is a vector of size t and we naturally assume that they belong to the manifold $\mathcal{M}$. Let $\mathbf{b}$ be some barycenter of these anchor points according to a metric ϕ as defined by:

$$\mathbf{b}(\lambda) = \text{argmin}_{\mathbf{z}} \sum_{a=1}^{d} \lambda_a \phi(\mathbf{z}, \varphi_a), \tag{3}$$

where $\{\lambda_a\}_a$ are the barycentric weights. The ability for such approach to actually provide physically meaningful solutions then relies on the capacity of this metric ϕ to preserve the desired physical properties. For that purpose, different metrics can be considered. In the rest of the paper, we will assume without loss of generality that the signals of interest are positive and sum to one. They can then be treated as a probability distribution. In such a case, different standard metrics can be considered:

- **Euclidean distance** Equipped with the Euclidean distance leads to defining some barycenter as follows:

$$\mathbf{b}(\lambda) = \text{argmin}_{\mathbf{z}\in\mathcal{S}_d} \sum_{a=1}^{d} \lambda_a \left\|\mathbf{z} - \varphi_a\right\|_2^2, \tag{4}$$

  where the weight parameters are enforced to belong to the d-dimensional simplex $\mathcal{S}_d$, thus guaranteeing them to sum to one. Without the simplex constraint, such barycenter would simply be the standard weighed combination of the examples $\{\varphi_a\}$.
- **Kullback-Leibler divergence** Since the data are assumed to belong to the m-dimensional simplex, it is natural to consider choosing the metric ϕ as a f-divergence, which measures a discrepancy between two probability masses. In case of the Kullback-Leibler divergence, barycenters are defined as:

$$\mathbf{b}(\lambda) = \text{argmin}_{\mathbf{z}\in\mathcal{S}_d} \sum_{a=1}^{d} \lambda_a \left(\varphi_a \odot \log\left(\mathbf{z} \oslash \varphi_a\right)\right), \tag{5}$$

  where the operators $\odot$ (*resp.* $\oslash$) stands for the element-wise multiplication (*resp.* division).
- **Optimal transport and the Wasserstein distance** The Wasserstein or Earth Mover's distance [22], [23] is an alternative choice, which has attracted a lot of interest during the last decade in the scope of image processing [24]. In the present context, manifold exploration can be performed by computing Wasserstein barycenters [25], [26]. This has been used with success for signal morphing, texture mixing, image registration [27] as well as non-linear dictionary learning [28] machine learning [27], [28].

It is importance to notice that none of the standard metric we considered above is fully adapted to build barycenters that belong, or at least are closed, general manifolds. In contrast, we propose to further learn how to build interpolants or barycenters, which are adapted to better capture the structure of the observed data.

Contributions

In the present paper, we investigate a learning-based approach that allows designing physically relevant regularization schemes from few training samples. In the spirit of optimal transportation, we introduce an autoencoder-inspired model that allows learning to transport points onto the underlying manifold from known anchor points. New points are then build though a non-linear interpolation scheme. In contrast to standard representation learning methods, it allows to build

efficient constraints that can be be conveniently plugged into traditional variational inference methods. In Section II, we introduce the proposed learning model, which we coin Interpolatory AutoEncoder (IAE). In the same section, we describe how the learnt IAE model can be used to regularize inverse problems. In Section IV, the IAE algorithm is evaluated with numerical simulations involving various learning scenarios. Section V, we apply the proposed method to realistic unmixing problems in γ-ray spectroscopy.

We will mainly focus on a particularly illustrative examples featured by X-ray spectroscopic measurements, where the underlying spectra are the instrumental response of line emission (*i.e.* dirac function) as displayed in Fig. 2. In this simple case, the observed spectrum for a single line emission is a Gaussian-shaped signal with width that decreases linearly with the position of its center. The (here known) underlying manifold a one-dimensional manifold that is embedded in a t-dimensional space. While the structure of this signal manifold is quite easy to handle, most physically relevant models are generally much more complicated, as we will see in Section V.

II. Learning to travel on an unknown manifold

A. Principle

The gist of our approach is to learn how to transport data points onto some unknown manifold $\mathcal{M}$. To that end, this is done by computing interpolants or barycenters of known anchor points, according to some metric to be learnt. This will therefore lead to building some interpolatory procedure, which we hereafter detail.

Ideally, a natural approach would amount to design some model for the metric to be learnt, leading to some parametric formulation ϕ_θ with θ the parameters to be learnt. This would eventually lead to the following expression of the barycenters:

$$\mathbf{b}(\lambda) = \text{argmin}_{\mathbf{z}} \sum_{a=1}^{d} \lambda_a \phi_\theta(\mathbf{z}, \varphi_a), \tag{6}$$

where ϕ_θ now stands for a parametric description of the metric, described with parameters θ to be learnt. Let's now define some training set $\mathcal{T} = \{s_i\}_{i=1,\cdots,T}$, each one being of size m. The learning procedure would then boil down to minimizing over the training set $\mathcal{T}$ the mean square error between the training samples and their barycenters. This also implies that the barycenter weights λ should also be optimized with respect to each training sample. This would lead to the following cost function:

$$\min_{\theta} \sum_{s_i \in \mathcal{T}} \min_{\lambda_i} \|\mathbf{b}(\{\lambda_i\}) - s_i\|_2^2, \tag{7}$$

where $\mathbf{b}(\lambda_i) = \text{argmin}_{\mathbf{z}} \sum_{a=1}^{d} \lambda_a^i \phi_\theta(\mathbf{z}, \varphi_a)$. However, this problem relies on an intricate formulation of the barycenters and the learning process with respect to the metric, which is very likely to be cumbersome to learn.

We rather propose to reformulate the interpolatory scheme by introducing some non-linear transformation or encoder Φ_θ, which maps the sample domain to some code domain, where the barycenters can be expressed in a much simpler form:

$$\mathbf{b}_{\Phi_\theta}(\lambda) = \text{argmin}_{\mathbf{z}} \sum_{a=1}^{d} \lambda_a \|\Phi_\theta(\mathbf{z}) - \Phi_\theta(\varphi_a)\|_2^2, \tag{8}$$

where $\mathbf{b}_{\Phi_\theta}$ stands for the resulting barycenter in the code domain, as defined by the transformation Φ_θ. It can then be mapped back to the sample domain through the application of the decoder Ψ_θ.

For fixed values of θ, the resulting weights for a single element $s_i \in \mathcal{T}$ can be formulated quite simply in the code domain as:

$$\lambda_i = \text{argmin}_{\lambda \in \mathcal{S}_d} \|\mathbf{b}_{\Phi_\theta}(\lambda) - s_i\|_2^2. \tag{9}$$

In this case, Eq. 8 simply leads to:

$$\begin{array}{rcl} \mathbf{b}_{\Phi_\theta}(\lambda) & = & \sum_a \lambda \Phi_\theta(\varphi_a) \\ & = & \varphi\lambda \end{array} \tag{10}$$

where φ stands for the $m \times d$ matrix, whose a-th row stores φ_a. This further allows for a very simple expression of the weights related to $s_i \in \mathcal{T}$:

$$\begin{array}{rcl} \lambda_i & = & \text{argmin}_{\lambda \in \mathcal{S}_d} \|\varphi\lambda - s_i\|_2^2 \\ & \simeq & \text{P}_{\mathcal{S}_d}\left(\left(\varphi^T\varphi\right)^{-1}\varphi^T s_i\right), \end{array} \tag{11}$$

where $\text{P}_{\mathcal{S}_d}$ stands for the projection onto the simplex $\mathcal{S}_d$. Due to the simplex constraint, the first problem is that such a problem does not admit a closed form solution. For the sake of simplicity, it is further approximated by a projected least-squares proxy, which is much less costly to implement in a learning procedure.

Rewritting the interpolatory scheme now allows for a simple three-step description:

- Applying the forward non-linear transform Φ_θ that maps the sample domain to the code or representation domain.
- A linear interpolator in the code domain, denoted by χ, which computes barycentric approximations as follows: $I(s) = \varphi \text{P}_{\mathcal{S}_d}\left(\left(\varphi^T\varphi\right)^{-1}\varphi^T s\right)$.
- Application of some backward Ψ_θ that maps back elements of the code domain to the sample domain.

Such a description is quite reminiscent of an autoencoder [29], where the mapping Φ_Θ would play the role of the encoder, and Ψ_Θ being the decoder. The main difference lies in the presence of a linear interpolator χ that computes barycenters from input samples in the code domain. In the next paragraph, we further detail the encoder and decoder mappings.

B. Network architecture and learning stage

Following the analogy with the autoencoders, the forward encoder Φ_Θ and backward decoder Ψ_Θ will be described by standard multi-layer forward neural networks. Both will be composed of $\mathcal{L}$ layers. In the present article, we restrict ourselves to a single type of architecture with a fixed number of neurons per layer, which will be equal to the dimension of the samples t. We leave to future work investigations on more general architectures. Consequently, the parameters to be learnt θ will be composed of the weight matrices and biases of both the encoder and decoder networks. The first discrepancy

with respect to a standard autoencoder architecture is that each layer is equipped with a skipped connection, which allows to provide a more robust learning stage for deeper networks. In the spirit of residual networks [30], [31], each layer of the encoder can be described as:

$$h^{(l+1)} = r_l\left(f_l\left(h^{(l)}\right), h^{(l)}\right) \quad (12)$$
$$= f_l\left(\mathbf{W}^{(l)}h^{(l)} + z^{(l)}\right) + \rho_l h^{(l)}, \quad (13)$$

where $\mathbf{W}^{(i)}$ is a standard weight matrix, $z^{(i)}$ the bias vector, and f_i an activation function. The skip connection acts by partially re-injecting $h^{(i)}$ up to a layer-dependent scalar factor $\rho_i < 1$. Generally speaking, these residual injection factors will be chosen small for low-level layers and larger for deeper layer, which will limit the vanishing gradient phenomenon commonly found when training deep networks [30].
A second discrepancy with respect to standard autoencoders is that the exact same encoder is applied simultaneously to both the training samples $\{x_i\}_{i=1,\cdots,T}$ and the anchor points $\{\varphi_a\}_{a=1,\cdots,d}$, as featured in Fig. 1.
Both encoded quantities $\Phi_\theta(s)$ and $\Phi_\theta(\varphi)$ are then processed by the *interpolator* I.

Algorithm 1: Sequential description of the *IAE*

Set $h^{(1)} = s$ and $h_\varphi^{(1)} = \varphi$
1-Apply the encoder to the training set and the anchor points
for $l \in 1\cdots,\mathcal{L}$ **do**
  $h^{(l+1)} = f_l\left(\mathbf{W}^{(l)}h^{(l)} + z^{(l)}\right) + \rho_l h^{(l)}$
  $h_\varphi^{(l+1)} = f_l\left(\mathbf{W}^{(l)}h_\varphi^{(l)} + z^{(l)}\right) + \rho_l h_\varphi^{(l)}$
end
2-Apply the interpolator χ in the code domain
$b_{\Phi_\theta} = I\left(h^{(\mathcal{L}+1)}, h_\varphi^{(l+1)}\right)$
3-Apply the decoder: $\tilde{v}_i = \Psi_\theta(b_{\Phi_\theta}(v_i))$
for $l \in \mathcal{L},\cdots,1$ **do**
  $\tilde{h}^{(l+1)} = g_l\left(\tilde{\mathbf{W}}^{(l)}\tilde{h}^{(l)} + \tilde{z}^{(l)}\right) + \rho_l\tilde{h}^{(l)}$
end

As illustrated in 1, the decoder is designed similarly to the encoder, with distinct parameters (no weight sharing).

Similarly to the encoder, the goal of the learning stage is to minimize the average reconstruction error defined by the Euclidean distance between the training samples and the estimated barycenters. On top of that, we propose adding an additional error term to measure the discrepancy between the data in the code domain. This extra term aims at enforcing the proximity of the encoded training samples with their barycenters in the code domain. We will see in Section IV that the resulting regularization allows for a more robust training stage.

The resulting cost function writes as follows:

$$\mathcal{J}(\theta) = \mu\sum_{s_i\in\mathcal{T}}\left\|s_i - \left(\Psi_\theta\circ I\circ\Phi_\theta\right)(s_i))\right\|_2^2\cdots$$
$$\cdots + \sum_{s_i\in\mathcal{T}}\left\|\Phi_\theta(s_i) - \left(I\circ\Phi_\theta\right)(s_i))\right\|_2^2, \quad (14)$$

where the interpolator I computes the barycenter of the transformed anchor points $\{\Phi_\theta(\varphi_a)\}_a$. The parameter μ controls the trade-off between the two terms. Implementation details as well as the choice and impact of the network parameters will be investigated and discussed in Section IV.

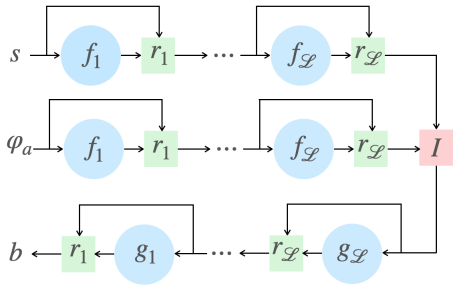

Fig. 1: Sketch of the IAE network.

III. Building model-based constraints from examples

A. Principle

In this paragraph, we discuss how one can make use of the proposed interpolatory scheme to build a dedicated function to regularize inverse problems of the form Eq. 2. In convex analysis, if the regularization term $\mathcal{R}$ is convex and admits some regularity properties [32], the solution of the problem in Eq. 2 is unique and can be solved with a proximal gradient descent algorithm, which takes the following form at iteration k:

$$\mathbf{x}^{(k+1)} = \text{prox}_{\mathcal{R}}\left(\mathbf{x}^k + \tau\mathbf{A}^T\left(\mathbf{y} - \mathbf{A}\mathbf{x}\right)\right), \quad (15)$$

where τ stands for the gradient step size. The operator $\text{prox}_{\mathcal{R}}(.)$ is the so-called proximal operator of $\mathcal{R}$: $\text{prox}_{\mathcal{R}}(\mathbf{x}) = \text{argmin}_{\mathbf{u}}\mathcal{R}(\mathbf{u}) + \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2$.
If one wants to enforce the solution $\mathbf{x}$ to belong to some manifold $\mathcal{M}$, then it would make sense to choose $\mathcal{R}$ as being the characteristic function of $\mathcal{M}$. In this case, $\text{prox}_{\mathcal{R}}$ would be defined as the orthogonal projection onto $\mathcal{M}$, in case the manifold maps to $\mathbb{R}^n$ via a surjective mapping.
Next, we substitute the property $\mathbf{x}\in\mathcal{M}$ with $\mathbf{x}\in\mathbb{B}_\phi(\{\varphi_a\})$, which denotes what we call, with a little abuse of language, the "barycentric span" of the anchor points $\{\varphi_a\}$:

$$\mathbb{B}_\phi(\{\varphi_a\}) = \left\{\mathbf{x}; \exists\{\lambda_a\}\in\mathcal{S}_d, \mathbf{x} = \Psi_\theta\left(\sum_a \lambda_a\Phi_\theta(\varphi_a)\right)\right\} \quad (16)$$

In plain english, this set is composed of all barycenters that can be built from the anchor point $\{\varphi_a\}$, which provides a good approximation of the manifold $\mathcal{M}$, at least at the vicinity of the anchor points, provided the learning stage has performed efficiently.

B. Projection onto the barycentric span

As detailed above, the equivalent proximal operator of the regularization term we introduce is the orthogonal projection onto the barycentric span of the anchor points. In practice, it amounts to seeking some weights $\hat{\lambda}$ so that the decoded interpolation minimizes the Euclidean distance with some data point $\mathbf{u}$:

$$\{\hat{\lambda}_a(\mathbf{u})\}_a = \text{Argmin}_{\lambda_a \in \mathcal{S}_d} \left\| \mathbf{u} - \Psi_\theta \left(\sum_a \lambda_a \Phi_\theta(\varphi_a) \right) \right\|_2^2 \tag{17}$$

Once the optimal solution $\{\hat{\lambda}_a(\mathbf{u})\}_a$ is computed, the final projection is defined as the decoded interpolation:

$$\text{prox}_{\mathcal{R}}(\mathbf{u}) = \Psi_\theta \left(\sum_a \hat{\lambda}_a(\mathbf{u}) \Phi_\theta(\varphi_a) \right) \tag{18}$$

This amounts to tackling a non-linear least-squares optimization problem.

IV. Numerical experiments

A. Illustration

In this section, we investigate how the proposed learnt interpolatory scheme performs in various scenarios. More precisely, we focus on testing how the training sample size, the depth of the underlying network, the sampling or the choice of the anchor points impact the learning procedure. For that purpose, we first make use of a simple but quite illustrative toy example.

The data are composed of Gaussian-shaped 1D signals with widths that depend on the position of their mean. Two different types of width/mean dependencies are considered, which lead to an increased complexity of the underlying manifold: i) linear dependency and ii) a periodic dependency. These second case is displayed in Figure 2. Examples of the such Gaussian-shaped signals are featured at the top of the same figure. The dimension of each sample is fixed to 50. While being a simple toy example, such signals, in the linear dependency case, are good approximations of the kind of emission line spectra that can be observed in Astrophysics [33]. In this context, the linear dependency originates from the instrumental response of the observation apparatus.

The set-up of the *IAE* algorithm is detailed in Table 1. The codes are implemented using JAX [34]. The learning stage is performed using the Adam optimizer [35]. The projection onto the barycentric span described in III makes use of the AdaGrad algorithm [36].

In the next, the quality of the recovered signals will be quantified using the Mean Squared Error (MSE). It is defined as the average of the sample-based MSE across the test set:

$$\text{MSE} = \sum_{i \in \mathcal{T}_{\text{test}}} \|x_i - \hat{x}_i\|_2^2, \tag{19}$$

where $\hat{x}_i$ stands for the projection onto the barycentric span of sample x_i: $\hat{x}_i = \text{prox}_{\mathcal{R}}(x_i)$. When noise is added, the MSE is computed with respect to the noiseless sample. Unless stated specifically, the anchor points are fixed so that they correspond to the extreme low and high positions of the Gaussian shapes.

Parameter	Set-up
Number of samples T	variable between 10 and 1000
Regularization parameter μ	variable between $1e-2$ and $1e3$
Number of layers $\mathcal{L}$	variable between 2 and 5
Residual parameter ρ	$\forall l \in [1, \mathcal{L}], \rho_l = \rho(2\frac{l}{\mathcal{L}} - 1)$ with $\rho = 0.1$
Number of epochs	15000
Learning rate	10^{-3}
Solver	Adam [35]
Batch size	100

TABLE I: IAE code setup parameters.

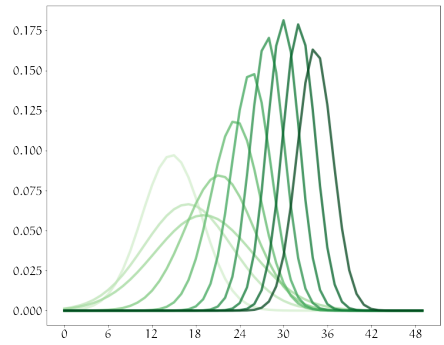

Fig. 2: Examples of Gaussian-shaped samples for a periodic relationship between width and position.

1) Impact of the regularization parameter μ: In this paragraph, we evaluate the role played by the regularization parameter μ. Let us first recall that μ allows to control the trade-off between the sample-based and code-based terms that compose the total cost function of Eq. 14. Hence, a large value of μ will favor a more reliable reconstruction of the retrieved barycenters in the sample domain but at the cost of relaxing the constraint in the code domain. In contrast, a small value will allow to better constrain the code, but with a potential deterioration of the reconstructed signals. Furthermore, the goal of the code-based constraint is somehow to ease the training process. Indeed, without this constraint, the cost function only depends on the input sample and its recovered estimate, which are separated by $2\mathcal{L}$ layers. When the number of layers increases, such models might be more difficult to learn.

To illustrate this point, Figure 3 shows the scatter plot of the reconstructed and input samples, both in the sample domain (left column) and code domain (right column). These results have been obtained with $\mathcal{L} = 2$ layers with a training set size of 500 samples in the case of a periodic dependency. To ease visualization, these scatter plots are obtained by projecting the samples onto the subspace generated by the two first principle components (*i.e.* projection onto the subspace generated by the two eigenvectors related to the two largest eigenvalues of the covariance matrix of the test set) of the test set in both domains. In these figures, the color bar provides the recovered MSE in dB, which allows quantifying the quality of the interpolant with respect to the signal across the overall test set.

μ	0.1	10	1000
Mean value in dB	68.14	66.96	68.32
Std of the mean value in dB	3.40	3.95	4.04

TABLE II: Mean and standard deviation across 50 model training trials of the mean MSE for various values of the regularization parameter μ.

In Figure 3, the plots in the first row display results obtained with a regularization parameter of $\mu = 1$. Since the number of anchor points is fixed to 2, constraining the barycentric weights to sum to one then enforces the barycenters to belong to the line defined by the anchor points in the code domain. This is exactly what can be observed in the top-right panel of Fig. 3. In contrast, opting for a larger $\mu = 10^3$, no more allows for this property to hold true anymore, giving much more weight to the sample-based term in the cost function. From the domain-based panels (first column if Fig. 3), it is interesting to point out that a larger value for μ does not seem to provide better reconstruction results in the sample domain.

More quantitatively, we propose computing the MSE of the recovered (interpolated) data for $\mu = 0.1$, 10 and 1000. To account for training variabilities, each result is obtained from 50 different trained models with independent random initializations. The test set size of 1000 samples. Table 2 reports the mean and standard deviation across the 50 independent model training trials of the mean MSE (taken across the test set). This allows to quantify the average MSE across the test set along with some stochastic variability that originates from the training procedure. As we emphasized previously, the code-based term in the total cost function allows to provide a more robust training by imposing an additional constrain in the code domain. Furthermore, this is at no cost for the reconstruction quality. Therefore, in practice, μ will be chosen as small as 0.1 or 1.

2) Going deeper, with few training samples: We now shed light on how the *IAE* algorithm performs when the number of layers to the network increases. To that purpose, the algorithm is run with a number of layers $\mathcal{L}$ that varies between 2 and 5, for various training sample sizes from 10 to 1000.

The quality of the recovered interpolated signals are evaluated with the mean MSE over 5 values $\mu \in [10^{-2}, 10^3]$, to take the average performance on a large range of regularization parameters. Figure 5 displays the evolution of the MSE in dB when the number of layers and the training sample size vary.

Let us first have a look at the linear dependency case. In this case, the underlying manifold has slower variations and therefore is expected to require less layers. Hence, it can be highlighted that for most training sample sizes, the quality of the reconstruction increases as the number of layers increase. A plateau is reached when for $\mathcal{L} = 3$. Except for the case $T = 50$, further increasing $\mathcal{L}$ does not improve the results if not leading to a slight deterioration. In this case, it is likely that the complexity of the learning procedure for

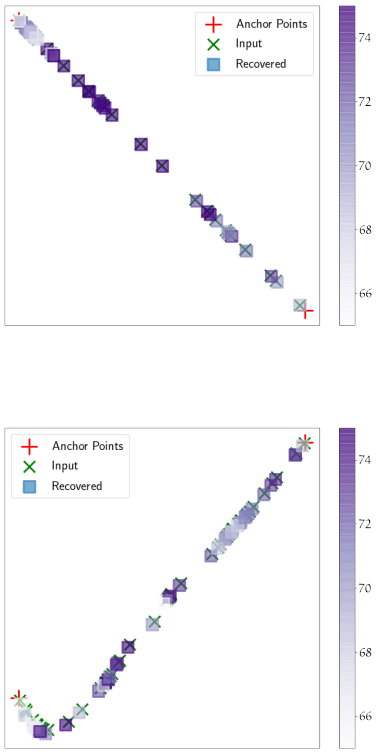

Fig. 3: Illustrating the role played by the regularization parameter μ: these plots display the scatter plot (after projection onto the two first principal components to ease visualization) of the test set samples in the code domain for $\mu = 1$ (top) and $\mu = 10^3$ (bottom).

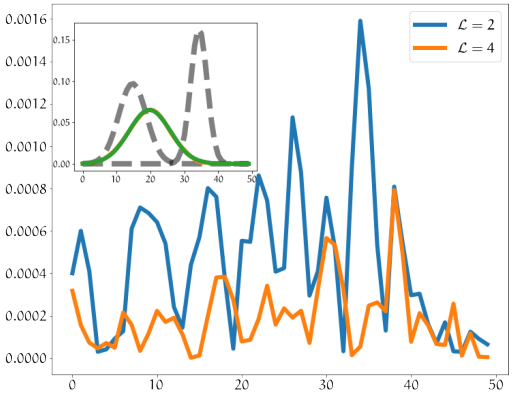

Fig. 4: Illustration of the recovery of a single signal for various number of layers. The inset figure shows displays the anchor points, the input and reconstructed signals for $\mathcal{L} = 2$ and 4. The main plot shows the reconstruction error (absolute value of the difference between the input and recovered signal). The number of training samples was fixed to 100, and the regularization parameter to $\mu = 0.1$.

deeper networks generate errors that dominate over potential benefits of better interpolatory schemes with larger depths. As expected, the quality of the reconstruction increases with the training sample size. If the case $T = 10$ leads to poor results, it is striking that with as small as 25 training samples, the performances quickly rocket close to 70 dB. This highlights the ability of the proposed scheme to provide a reliable and efficient process to interpolatory procedure in a sample-weaned off learning setting.

In the case of the more complex manifold generated with a periodic dependency (panel of the right in Fig. 5), similar conclusions can be drawn. In contrast, going deeper up to $\mathcal{L} = 5$ now constantly increase the reconstruction quality as long as the number of samples is large enough to learn the additional parameters that deeper networks involve. In this case, a number of samples as small as 50 already provides good construction results for $\mathcal{L} = 3$ or more.

We also performed comparisons with a variational autoencoder (VAE - [37]), which can also be used for manifold learning. In this case, we used a combination the denoising VAE (for increased robustness with respect to noise - [38]) and the β-VAE [39] which allows to better balance between reconstruction accuracy and stochastic modeling in the latent space. For fair comparisons, the network parameters (*i.e.* number of layers, number of hidden units per layer and dimensionality of the latent space and number of training samples) are the same as the *IAE* . More quantitatively, Table 3 reports on the evolution of the MSE when the number of layers increases and the training sample size varies from 10 to 500. It first highlights that the β-VAE provides good results when the number of training samples is large enough; as it could be expected, it fails at learning a good data representation when the number of samples is very low (*i.e.* 10). In contrast, the *IAE* provides significantly more accurate signal approximations, with a limited degradation when the number of training samples decreases.

These two simple scenarios already allow to highlight that the *IAE* algorithm can provide good interpolatory schemes with a quite small training sample size. It is clear that the number of layers to be used as well as the number samples to train the underlying network closely depend on the complexity of the manifold, whose properties somehow need to be captured, which is problem specific.

		10	50	100	500
2	VAE	28.6	28.9	33.8	52.8
	IAE	53.7	64.4	68.1	71.5
3	VAE	27.4	26.9	31.9	50.8
	IAE	54.2	69.9	70.6	74.5
4	VAE	27.7	29.0	35.0	51.2
	IAE	52.2	70.6	71.8	75.5

TABLE III: Comparison between the *IAE* and β-VAE - evolution of the MSE in dB when the number of layers varies in the case of periodic dependencies.

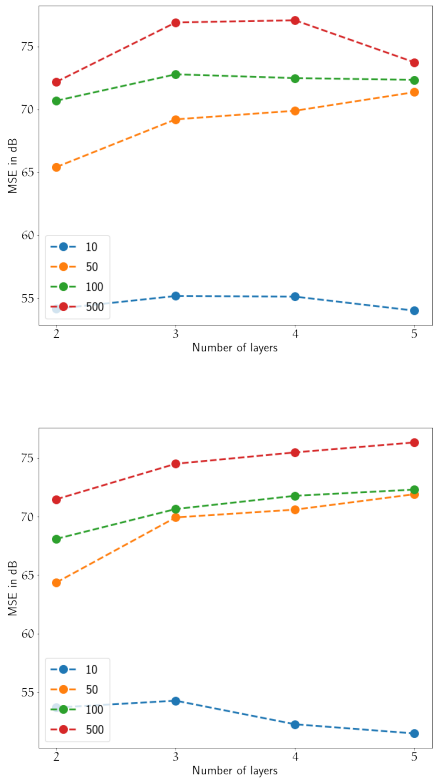

Fig. 5: Increasing the number of layers: evolution of the MSE (in dB) when the number of layers and the training sample size vary, for the linear dependency case (top) and the periodic dependency case (bottom).

3) Robustness to noise: In this article, the goal of the proposed interpolatory scheme is to eventually build regularizations for inverse problems. We highlighted in section III that this is performed by designing the projection onto the barycentric span of the anchor points. However, tackling a linear inverse problem with such a constraint requires plugging the resulting proximal operator in an iterative minimization processing, such as a gradient descent in the popular Forward-Backward Splitting algorithm [40]. In this minimization scheme, the current estimate to which the proximal operator is applied does not lie on the barycentric span nor on the manifold $\mathcal{M}$. At best, it can be described as the sought-after signal with an extra contamination or additive noise. Therefore, the ability to use the proposed regularization scheme to solve inverse problems highly relies on the robustness of the barycentric span projection to noise. In this section, we investigate the robustness of the projection onto the learnt barycentric span. For that purpose, the projection $\text{prox}_{\mathcal{R}}$ is applied to noisy versions of the training samples. In this experiment, white and centered Gaussian noise with standard deviance σ is added. This has been performed with a toy example involving the periodic dependency case. The left (*resp.* right) panel of Figure 6 shows the evolution of the MSE in dB as a function of the signal-to-noise ratio (SNR) in dB, when the number of layers is fixed to $\mathcal{L} = 2$

(*resp.* $\mathcal{L} = 5$) . As expected, the recovery MSE improves as the SNR increases. Furthermore, using a larger number of training samples leads to learning models with increased robustness with respect to the input noise. It is interesting to note that when a very small training sample size is used (10 samples), the recovery MSE quickly reaches a plateau for SNR values larger than 30dB, which reveals that the approximation or interpolation error then dominates over the noise-related error. This is not observed when the training sample size is larger than 25.
It is also important to point out that increasing the number of layers leads to the same conclusion. Increasing the complexity of the model allows improving the quality of the reconstruction without hampering its robustness to noise.

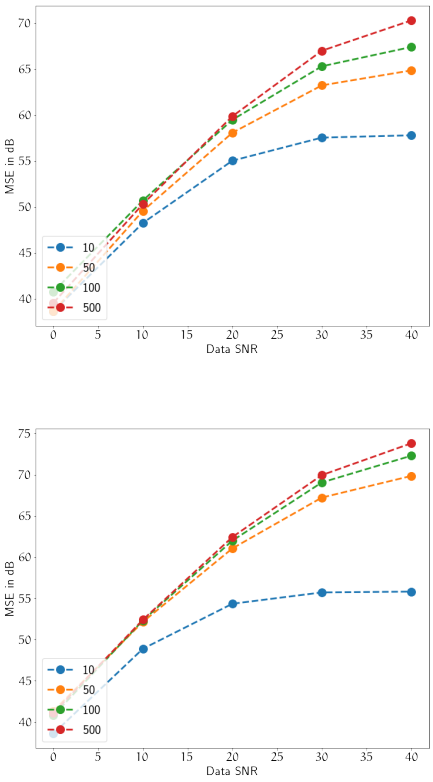

Fig. 6: Periodic dependency case: evolution of the recovered MSE as a function of the SNR for various training sample sizes. The number layers is set to $\mathcal{L} = 2$ (top) and $\mathcal{L} = 5$ (bottom).

4) Comparisons to other methods: The next step consists in comparing how the learning interpolatory scheme performs with respect to fixed metrics. For that purpose, we make use of the same test set and compute the projection of the test samples onto the barycentric span according to the Euclidean norm, Kullback-leibler divergence and Wasserstein distance. Comparisons have also been carried out with the β-VAE. Several values for the parameter β [39] have been tested in the range $[10^{-4}, 10]$; we found that $\beta = 10^{-3}$ provided the most accurate reconstructions.

Figure 7 displays an example of reconstruction results using these different metrics as well as the learnt interpolatory methods when $\mathcal{L} = 5$ layers are used and the number of training samples is fixed to 500. The noise level is fixed to 10dB. The anchor points are the same for all the methods and displayed in dashed lines. As expected the Euclidean metric does not lead to a realistic reconstruction since it basically spits out a linear combination of the anchor points. Both the Kullback-Leibler divergence and the Wasserstein distance provide results with more relevant morphologies. Recalling that the data are generated from the Gaussian-shaped samples with periodic dependency, the complexity of the variations as a function of the position prevents such metrics to yield good estimation results. Switching to the linear dependency would definitely lead to more reasonable results. β-VAE yields significantly better solutions than the interpolation methods based on fixed metrics; however, it does not seem to be as accurate as the *IAE* , with clearly less robustness with respect to noise. These results particular highlight that the *IAE* is well suited to capture the structured of the underlying manifold, which is out of reach for fixed metrics.
More quantitatively, Figure 8 shows the evolution of the MSE as a function of the SNR for the fixed metrics and the *IAE* algorithm. This plot highlights that, for this specific manifold, fixed metrics are not well adapted, whose reconstruction errors are dominated by the approximation error not the noise-related error.

5) Choosing the anchor points: The role played by the anchor points, and more precisely their particular choice, will highly depend on the complexity of the manifold $\mathcal{M}$. Let us recall that ideally the *IAE* tends to linearize the code domain. In such ideal case, the choice of the anchor points would be minor as long as their number is strictly larger than the intrinsic dimension of $\mathcal{M}$, as long as defining a barycenter on $\mathcal{M}$ boils down to some triangulation problem. In practice, and more specifically for complex manifolds, it is likely that the quality of the learnt model will depend on the choice of the parameters.
To evaluate the impact of the anchor points on the performances of the interpolatory procedure, we hereafter consider different strategies to choose the anchor points for two distinct datasets: i) the Gaussian-shaped signals with a periodic dependency and ii) simulations of astrophysical spectral in X-rays [33] displayed in Figure 9, which we know live on a 2-dimensional manifolds.
The strategies we propose to investigate are:

- **"hand-chosen" anchor points:** highly contrasted anchor points are chosen manually within the training set. In the case of the Gaussian-shaped signals, the two most extreme anchor points are chosen. In the case of the astrophysical signals, three anchor points have been chosen so as to maximize the contrast between them.
- **random anchor points:** the anchor points are chosen randomly within the training set. In this case, the forthcoming results will be averaged over 10 Monte-Carlo simulations obtained with 10 random draws of the anchor points.

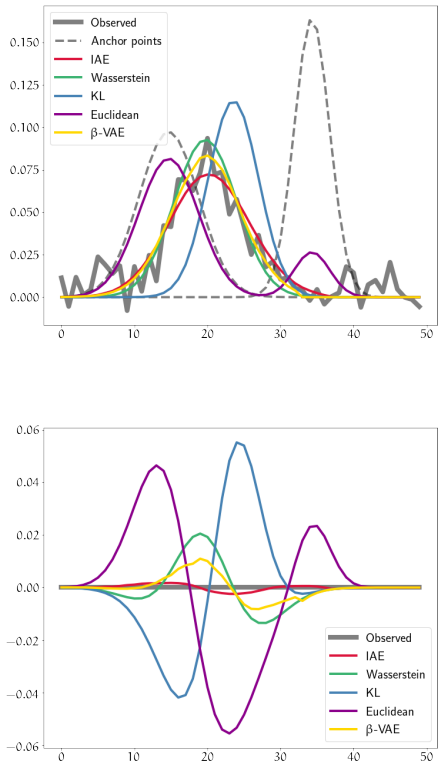

Fig. 7: Illustration of a signal recovery problem from noisy measurements. This top figure displays the input noise signal as well as the ones recovered with the IAE, β-VAE and interpolation with fixed metrics. The bottom panel shows the estimation residual defined by the difference between the recovered signals with the input signal.

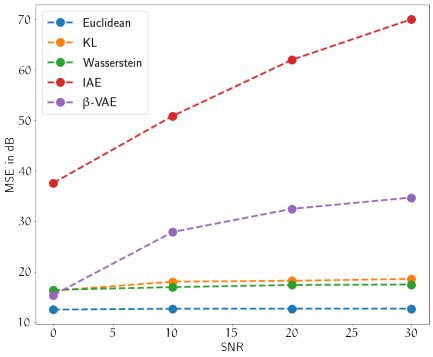

Fig. 8: Periodic dependency case: evolution of the recovered MSE as a function of the SNR for the *IAE* with $\mathcal{L} = 5$ layers and fixed metrics: Euclidean, KL and Wasserstein.

- **separable NMF:** the problem of choosing the anchor points amounts to picking certain elements of the training set. A simple approach to select significantly different anchor points can be done by tackling this problem as a separable non-negative matrix factorization problem. For that purpose, we use the Successive Nonnegative Projection Algorithm (SNPA - [41]) to automatically select anchor points.

The quality of the model is then performed by learning a *IAE* model for each strategy. In all cases, the *IAE* parameters are the same. The regularization parameter is fixed to 1 and the number of training samples is 100. The number of layers varies between 2 and 4. The results are reported in Table **??**. In the case of the Gaussian-shaped signals, which are associated with a rather smooth manifold, the results are not significantly sensitive to the choice of anchor points. Both a careful manual choice and the NMF-based approach lead to similar results with a slight advantage for the separable NMF. Random choices yield slightly degraded results, with about a 3 ot 4dB loss. For the more complex manifold spanned by the astrophysical components, the results are significantly more contrasted. the NMF-based approach always lead to the best results. With the exception of the case $\mathcal{L} = 2$, randomly choosing the anchor points always yield worse results with a difference reaching 12.7dB for $\mathcal{L} = 4$ with respect to NMF. Interestingly for these simple manifolds, the NMF-based approach tend to provide an automatic method to select reasonably good anchor points. It is very likely that much more complex manifolds will require more sophisticated means to pick more adapted anchor points and select the right number of anchor points. These investigations go beyond of the scope of the present article.

		hand-chosen	NMF	random
Gaussian-shaped	$\mathcal{L} = 2$	66.1	65.9	62.0
	$\mathcal{L} = 3$	70.0	70.2	66.7
	$\mathcal{L} = 4$	70.5	71.5	68.7
Astrophysical	$\mathcal{L} = 2$	60.1	64.7	60.7
	$\mathcal{L} = 3$	66.4	74.0	61.4
	$\mathcal{L} = 4$	72.3	76.7	64.0

TABLE IV: TBD.

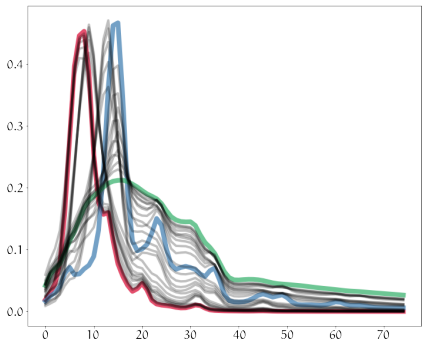

Fig. 9: Astrophysical signals representing the thermal emission spectra in X-ray as seen by the Chandra telescope.

V. Application to semi-blind linear regression

A. Principle

In this paragraph, we apply the proposed interpolatory learning scheme and its resulting barycentric span projection to a realistic linear inverse problem in nuclear physics: unmixing of γ-ray spectra. In various radioactivity-related applications, γ-spectrometry is a classical technique to analyze and to quantify γ-emitting radionuclides. Depending on γ-ray energies interacting in the detector used (*e.g.* semiconductor, scintillation), the main features of experimental γ-spectra consist in an overlap of Compton continua and full-absorption peaks of individual radionuclides. In the case of low-statistics measurements like for environmental applications, the current trend of automatic identification of radionuclides is to implement full-energy spectrum analysis to get the whole information of a γ-spectra [42], [43] for subsequent decision-making. In general, the overall shape of γ-spectra is sensitive to experimental conditions due to Compton scattering in the surrounding materials of a γ-emitting radioactive source. This effect generally yields significant variations of the peak-to-Compton ratios. In the present study, this problem was addressed with a 3"×3" NaI(Tl) scintillation detector by means of radiation-matter simulations using the Monte Carlo code Geant4 [44].
The modeling of a 3"×3" NaI(Tl) detector with the Geant4 code was performed by considering the geometry of the scintillation crystal and the housing which has a total thickness of 2 mm with 1.5 mm for the reflector (alumina) and 0.5 mm of aluminum. The γ-ray emission was simulated from a point source in a sphere made of steel and placed at 20 cm from the front of the detector. The response of the scintillation detector was simulated by increasing the thickness of the sphere from 2 mm to 30 mm according to successive steps of 2 mm. In the present work, the γ-ray emission was considered for two radionuclides having different energies taken from nuclear data : ^{57}Co and ^{133}Ba. Each simulated spectrum corresponds to 10^8 radioactive decays leading to γ-ray emission according to a 4π solid angle. The γ-spectra shown in the Fig.10 features the evolution of the overall shape when increasing the radius of the sphere. The energy resolution of the scintillation detector was also taken into account.

In this context, the measurement $\mathbf{y}$ (of size t) is composed of a noisy linear non-negative mixture of the spectral signatures of the sought-after radionuclides:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n},$$

where each column of $\mathbf{A}$ (of size $t \times p$) now contains the individual spectral signatures, and $\mathbf{x}$ (of size p) stands for some weight that quantify the contribution of each radionuclide in the measurement. In this article, we chose for simplicity to stick to an additive white Gaussian noise of mean value 0 and standard deviation σ.
In in-situ measurements (*e.g.* radioactivity monitoring at the borders), one of the key challenges that the estimation of $\mathbf{x}$ raises is that the signatures $\mathbf{A}$ may significantly vary, depending on the measurement condition, which results in some screening effect on the observed signature as described above. Therefore, both $\mathbf{A}$ and $\mathbf{x}$ need to be estimated simultaneously. The problem in Equation 2 needs to be rewritten as follows:

$$\min_{\mathbf{x},\mathbf{A}} \mathcal{R}(\mathbf{x}) + \frac{1}{2}\left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2^2. \tag{20}$$

Since only a single measurement $\mathbf{y}$ is available, this is an ill-posed inverse problem, which requires imposing strong constraints on the shapes of the spectral signatures to be retrieved.
As an illustration, Figure 10 display the evolution of the spectral signatures of the cobalt ^{57}Co and baryum ^{133}Ba for different screening effects due to steal shielding. Ideally, since the variations of these spectra only depend on the thickness of the steal shielding, these spectra should belong to an unknown one-dimensional manifold.
In this application of the *IAE* algorithm, the goal is to first learn a model for each of the two considered radionuclides, namely cobalt and baryum, and then constrain the corresponding spectra to belong to the barycentric span of theses models. Since, only non-negativity is imposed to $\mathbf{x}$, the cost function we propose to minimize is the following one:

$$\min_{\mathbf{x},\mathbf{A}} \chi_{\{\,.\,\geq 0\}}(\mathbf{x}) + \sum_{i=1}^{p} \chi_{\mathbb{B}_{\phi_i}}(\mathbf{A}^i) + \frac{1}{2}\left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2^2, \tag{21}$$

where $\chi_{\{\,.\,\geq 0\}}(\mathbf{x})$ stands from the characteristic function of the positive orthant (*i.e.* set of vectors with non-negative entries) and $\chi_{\mathbb{B}_{\phi_i}}(\mathbf{A}^i)$ stands for the characteristic function of the barycentric span for model ϕ_i applied to the i-th column of $\mathbf{A}$.
This is a non-convex problem but assuming that it is a multi-convex problem (*i.e.* convex once $\mathbf{x}$ or $\mathbf{A}$ is fixed), it can be minimized using the Block coordinate descent (BCD) algorithm [45]. In the present context, BCD performs iteratively by minimizing the overall cost function with respect to $\mathbf{x}$ and $\mathbf{A}$. The proposed semi-blind spectral unmixing algorithm is the following:

Algorithm 2: Semi-blind spectral unmixing

Initialization;
while *While stopping criterion is not valid* **do**
  1-Update $\mathbf{x}$:
  $\mathbf{x}^{(k+1)} = \text{Argmin}_{\mathbf{x}} \chi_{\{\,.\,\geq 0\}}(\mathbf{x}) + \frac{1}{2}\left\|\mathbf{y} - \mathbf{A}^{(k)}\mathbf{x}\right\|_2^2$
  ;
  2-Update $\mathbf{A}$: $\mathbf{A}^{(k+1)} =$
  $\text{Argmin}_{\mathbf{A}} \sum_{i=1}^{p} \chi_{\mathbb{B}_{\phi_i}}(\mathbf{A}^i) + \frac{1}{2}\left\|\mathbf{y} - \mathbf{A}\mathbf{x}^{(k+1)}\right\|_2^2$;
  Check convergence;
end

Both steps 1 and 2 are implemented using a Forward Backward Splitting algorithm [40]. According to [45], the BCD algorithm is guaranteed to converge to a local critical point provided that each update problem is convex. This is however not theoretically proved for step 2 since even the uniqueness nor the convexity of the projection onto the barycentric span of a given model is not proved theoretically. So far, in the numerical test we carried out on γ-ray spectra, the BCD algorithm tends to stabilize towards a fixed point. For

each sub-problem, the maximal number of iterations is fixed to 10^4, the gradient path length is fixed based on the Lipshitz gradient as detailed in [46]. The maximal number of iterations of the outer loop is fixed to 1000. Both in the inner loops and in the outer loop, the convergence criterion is defined as the relative variation of the variable to be minimized and the loop breaks when it is lower than 10^{-6}.

B. Numerical experiments

In this application, the measurement y is composed of the linear mixture of 2 radionuclides: baryum and cobalt, whose spectral signatures are displayed in Figure 11. The mixture weights are set as $\mathbf{x} = [0.25, 1]$ respectively for the cobalt and baryum. White Gaussian noise with standard deviation σ_n is added. An example of such measurement with a SNR of 10dB is displayed in Figure 12.

The *IAE* algorithm is applied to 15 training samples for both radionuclides, with a number of layers $\mathcal{L} = 4$ and a regularization parameter $\mu = 1$.

The semi-blind spectral unmixing algorithm is run with a maximum number of iterations of 1000 for both the main BCD loop and updates 1 and 2. Convergence is said to be reached for update 1 (*resp.* update 2) when the relative variation of $\mathbf{x}$ (*resp.* $\mathbf{A}$) between two iterations is lower than 10^{-6}. For the BCD loop, convergence is declared when the relative variation of $\mathbf{A}$ between two iterations is lower than 10^{-6}. The initial point for $\mathbf{x}$ is a vector with zero entries. The initial matrix of spectral signatures $\mathbf{A}$ is taken at random in the training set.

Figure 12 shows the initial and estimated spectral signatures for baryum for a SNR of 10dB. It has to be highlighted that although the initial point is quite far from the true solution, the BCD-based solution based on the learnt model allows to retrieve a very decent solution. The same figure also features the solution provided by the exact same BCD algorithm but using the barycentric span constraint involving the Kullback-Leibler divergence. In this experiment, the latter allows to retrieve a reasonable spectral signatures for both radionuclides, with no visible leakage between them. However, it is clearly outperformed by the more adaptive learning-based constraint.

More quantitatively, the left panel of Figure 13 shows the evolution of the maximum percentage error of the mixing weights $\mathbf{x}$ as a function of the signal-to-noise ratio. This is defined as: $\max_i 100 * |1 - \hat{\mathbf{x}}_i / \mathbf{x}_i^\star|$, where $\hat{\mathbf{x}}$ (*resp.* $\mathbf{x}^\star$) are the estimated (*resp.* true) mixing weights. The purple line, which is dubbed *oracle*, is defined as the error obtained by making use of the true matrix of spectral signatures; it is the error one would obtain if the spectral signatures were perfectly known. This defines a best error lower bound. In contrast, the red line corresponds to the non-blind unmixing, which corresponds to no update of $\mathbf{A}$ in the spectral unmixing code. The resulting error is therefore largely impacted by the error on the spectral signatures. This figure also displays the results we obtained using the KL-based as well as the *IAE* -based spectral unmixing algorithms with two different numbers of layers $\mathcal{L} = 3$ and 4. Firstly, while Figure 12 showed that the KL-based method allowed visually reasonable estimated $\mathbf{A}$, the retrieved mixing weights are very sensitive to even a slight mis-estimation of $\mathbf{A}$, as shown by the poor error observed.

The *IAE* -based unmixing method lead to significantly improved mixing weights error in the top panel of Figure 13. This is all the more accurate as the number of layers is larger. The bottom panel of Fig. 13 shows the evolution of the MSE in dB of the recovered baryum spectrum for both the KL-based and *IAE* -based spectral unmixing algorithms. This highlights that the KL-based algorithm quickly results in errors that are dominated by the model approximation error rather than the noise-related error (typically for SNR $>$ 25dB). In contrast, the *IAE* -based method leads to significantly better estimated spectral signatures, with a MSE already larger than 40dB for SNR $>$ 10dB when $\mathcal{L} = 4$ layers are used.

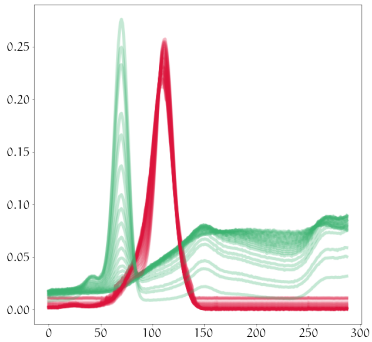

Fig. 10: Spectral variabilities of baryum (green) and cobalt (red). The units in the abscissa axis are in keV.

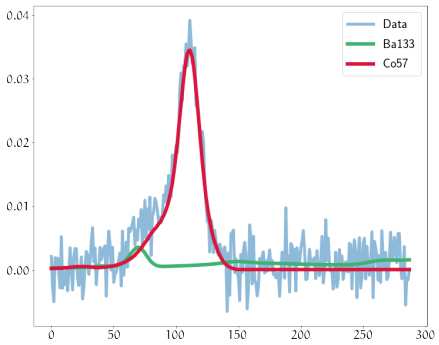

Fig. 11: Example of a γ-ray spectrum measurement: the input ^{57}Co spectrum in red, the input ^{133}Ba in green and the noisy mixture in red. The units in the abscissa axis are in keV.

VI. Conclusion

In this article, we introduce a novel approach to build learning-based regularization methods to tackle inverse prob-

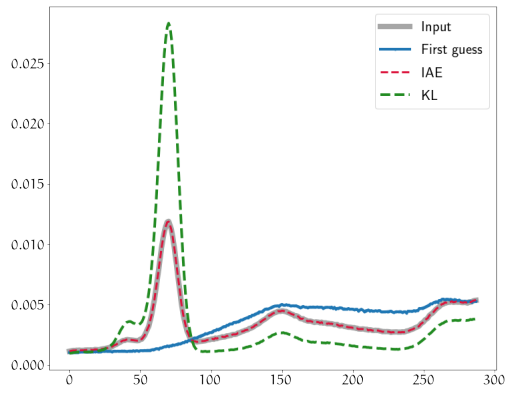

Fig. 12: The input, first guess (initialization of the unmixing algorithm) and reconstructed baryum spectra for SNR $= 10$dB. The units in the abscissa axis are in keV.

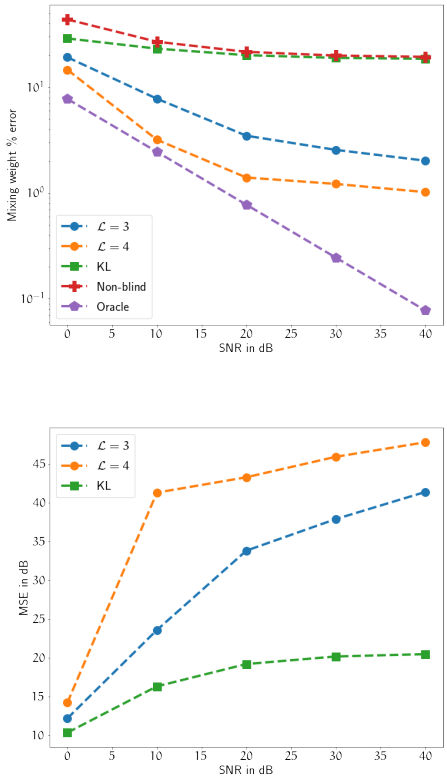

Fig. 13: Top panel: evolution of the maximum percentage error of the mixing weights as a function of the SNR. Bottom panel: evolution of the mean squared estimation error of the spectral signatures as a function of the SNR.

lems. In contrast to standard learning methods, it builds upon a procedure that learns to interpolate onto an unknown manifold from known anchor points. For that purpose, we introduce a interpolatory learning scheme that is reminiscent of the autoencoder architecture, with the exception that an extra linear interpolator is introduced in the code domain. We show that the proposed model can be fruitfully used to regularize inverse problem by imposing the sought-after solution to belong to the barycentric span of the anchor points, yielding physically meaningful solutions. The proposed approach, which we coined *IAE* , performs quite efficiently, even when the training samples are scarce. This makes it a very good candidate for a large number of applications few training samples are available. Extensive numerical experiments have been carried out to test the *IAE* algorithm, in a very large range of experimental scenario. We further demonstrate how the proposed approach can be beneficial to regularize a semi-blind linear regression problem in the field of γ-ray spectroscopy.

REFERENCES

[1] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Institute of Physics, 1998.

[2] J.-L. Starck, F. Murtagh, and M.-J. Fadili, *Sparse Image and Signal Processing - Wavelets, Curvelets, Morphological Diversity*. Cambridge University Press, 2010.

[3] M. T. McCann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," *IEEE Signal Processing Magazine*, vol. 34, pp. 85–95, 2017.

[4] M. Mardani, Q. Sun, S. Vasawanala, V. Papyan, H. Monajemi, J. Pauly, and D. Donoho, "Neural proximal gradient descent for compressive imaging," in *Neural Information Processing Systems (NIPS)*, 2018.

[5] J. Bobin, J. Rapin, A. Larue, and J.-L. Starck, "Sparsity and adaptivity for the blind separation of partially correlated sources." *IEEE Transanctions on Signal Processing*, vol. 63, no. 5, pp. 1199–1213, 2015.

[6] A. Repetti, M. Pham, L. Duval, E. Chouzenoux, and J.-C. Pesquet, "Euclid in a taxicab: Sparse blind deconvolution with smoothed l1/l2 regularization," *IEEE Signal Processing Letters*, vol. 22, no. 5, 2015.

[7] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," *Journalof the American Mathematical Society*, vol. 29, pp. 983–1049, 2013.

[8] A. Bora, A. Jalal, E. Price, and A. Dimakis, "Compressed sensing using generative models," *International Conference on Machine Learning (ICML)*, p. 537–546, 2017.

[9] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen, "Variational methods in imaging," *Springer*, 2009.

[10] J. Adler and O. Oktem, "Solving ill-posed inverse problems using iterative deep neural networks," *CoRR - abs/1704.04058*, 2017.

[11] ——, "Learned primal-dual reconstruction," *Transactions on Medical Imaging*, vol. 37, 2018.

[12] D. Gilton, G. Ongie, and R. Willett, "Neumann networks for inverse problems in imaging," *arXiv:1901.03707*, 2019.

[13] Y. Yan, S. Jian, L. Huibin, and X. Zongben, "Deep admm-net for compressive sensing mri," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[14] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012, pp. 341–349. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/6cdd60ea0045eb7a6ec44c54d29ed402-Paper.pdf

[15] H. Kerstin, K. Teresa, K. Erich, R. M. P., S. D. K., P. Thomas, and K. Florian, "Learning a variational network for reconstruction of accelerated mri data," *Magnetic Resonance in Medicine*, vol. 79, no. 6, pp. 3055–3071. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.26977

[16] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, 2017.

[17] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM Journal on Imaging Sciences*, vol. 10(4), 2017.

[18] F. Sureau, A. Lechat, and J.-L. Starck, "Deep learning for a space-variant deconvolution in galaxy surveys," *Astronomy and astrophysics*, vol. 11, no. A67, 2020.

[19] S. Lunz, O. Öktem, and C.-B. Schönlieb, "Adversarial regularizers in inverse problems," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018, pp. 8507–8516. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/d903e9608cfbf08910611e4346a0ba44-Paper.pdf

[20] S. Ravishankar, A. Lahiri, C. Blocker, and J. Fessler, "Deep dictionary-transform learning for image reconstruction," *IEEE International Symposium on Biomedical Imaging*, p. 1208–1212, 2018.

[21] B. Culpepper and B. Olshausen, "Learning transport operators for image manifolds," in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds., vol. 22. Curran Associates, Inc., 2009, pp. 423–431. [Online]. Available: https://proceedings.neurips.cc/paper/2009/file/a1d50185e7426cbb0acad1e6ca74b9aa-Paper.pdf

[22] C. Villani, *Optimal Transport: Old and New*. Grundlehren Math. Wiss. 338, Springer, 2008.

[23] Peyre,G. and Cuturi,M., "Computational optimal transport," *Foundations and Trends in Machine Learning*, vol. 51, no. 1, p. 1–44, 2019.

[24] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.,*, vol. 40, 2000.

[25] M. Agueh and G. Carlier, "Barycenters in the wasserstein space," *SIAM J. Math. Anal.*, vol. 43, 2011.

[26] M. Cuturi and A. Doucet, "Fast computation of wasserstein barycenters," in *Proceedings of the 31st International Conference on Machine Learning,*, 2014.

[27] S. Kolouri, S. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE Signal Processing Magazine*, vol. 34, pp. 43–59, 07 2017.

[28] Schmitz,M. and Heitz,M. and Bonneel,N., and Ngole,F. and Coeurjolly, D. and Cuturi,M. and Peyre,G. and Starck,J.-L., "Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning," *SIAM Journal on Imaging Sciences*, vol. 11, no. 1, pp. 643–678, 2018.

[29] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, cite arxiv:1512.03385Comment: Tech report. [Online]. Available: http://arxiv.org/abs/1512.03385

[31] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016, cite arxiv:1608.06993Comment: CVPR 2017. [Online]. Available: http://arxiv.org/abs/1608.06993

[32] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[33] A. Picquenot, F. Acero, J. Bobin, P. Maggi, J. Ballet, and G. Pratt, "Novel method for component separation of extended sources in x-ray astronomy," *A&A - accepted - ArXiv:1905.10175*, 2019.

[34] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, and S. Wanderman-Milne, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/google/jax

[35] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980v8

[36] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[37] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[38] D. J. Im, S. Ahn, R. Memisevic, and Y. Bengio, "Denoising criterion for variational auto-encoding framework," in *AAAI*, 2017, pp. 2059–2065. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14213

[39] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *ICLR*, 2017.

[40] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005. [Online]. Available: http://link.aip.org/link/MMSUBT/v4/i4/p1168/s1&Agg=doi

[41] N. Gillis, "Successive nonnegative projection algorithm for robust nonnegative blind source separation," *SIAM Journal on Imaging Sciences*, vol. 7, no. 2, pp. 1420–1450, 2014. [Online]. Available: https://doi.org/10.1137/130946782

[42] J. Xu, J. Bobin, A. de Vismes Ott, and C. Bobin, "Sparse spectral unmixing for activity estimation in γ-ray spectrometry applied to environmental measurements," pp. 405–409, 2020.

[43] R. André, C. Bobin, J. Bobin, J. Xu, and A. de Vismes Ott, "Metrological approach of γ-emitting radionuclides identification at low statistics: application of sparse spectral unmixing to scintillation detectors," 2021.

[44] S. Agostinelli and et al., "Geant4—a simulation toolkit," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250 – 303, 2003.

[45] Y. Xu and W. Yin, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *arXiv preprint arXiv:1410.1386*, 2014.

[46] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, p. 183, 2009. [Online]. Available: http://link.aip.org/link/SJISBI/v2/i1/p183/s1&Agg=doi