



# END-TO-END LEARNING OF VARIATIONAL MODELS AND SOLVERS FOR THE RESOLUTION OF INTERPOLATION PROBLEMS

Ronan Fablet, Lucas Drumetz, François Rousseau

## ► To cite this version:

Ronan Fablet, Lucas Drumetz, François Rousseau. END-TO-END LEARNING OF VARIATIONAL MODELS AND SOLVERS FOR THE RESOLUTION OF INTERPOLATION PROBLEMS. ICASSP 2021 : IEEE International Conference on Acoustics, Speech and Signal Processing, Jun 2021, Toronto, United States. 10.1109/ICASSP39728.2021.9414629 . hal-03139133

**HAL Id: hal-03139133**

**<https://imt-atlantique.hal.science/hal-03139133>**

Submitted on 26 Mar 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# END-TO-END LEARNING OF VARIATIONAL MODELS AND SOLVERS FOR THE RESOLUTION OF INTERPOLATION PROBLEMS

*R. Fablet<sup>1\*</sup>, L. Drumetz<sup>1</sup>, F. Rousseau<sup>2</sup>*

<sup>1</sup>IMT Atlantique, UMR CNRS Lab-STICC, Brest, FR

<sup>2</sup>IMT Atlantique, UMR INSERM LaTIM, Brest, FR

## ABSTRACT

Variational models are among the state-of-the-art formulations for the resolution of ill-posed inverse problems. Following recent advances in learning-based variational settings, we investigate the end-to-end learning of variational models, more precisely of the regularization term given some observation model, jointly to the associated solver, so that we can optimize the reconstruction performance. In the proposed end-to-end setting, both the variational cost and the gradient-based solver are stated as neural networks using automatic differentiation for the latter. We consider an application to inverse problems with incomplete datasets (image inpainting and multivariate time series interpolation). We experimentally illustrate that this framework can lead to a significant gain in terms of reconstruction performance, including w.r.t. the direct minimization of the variational formulation derived from the known generative model.

**Index Terms**— inverse problems, missing data, variational models, regularization, optimizer learning, end-to-end

## 1. INTRODUCTION

Solving an inverse problem consists in computing an acceptable solution of a model that can explain measured observations. Inverse problems involve so-called forward or generative models describing the generation process of the observed data. The inversion of the defined forward model is frequently an ill-posed problem. Variational approaches are very popular in this context. They state the resolution of the inverse problems as the minimization of an energy or cost. This cost can be broken down into two terms: a data fidelity term related to a specific observation model and a regularization term characterizing the space of acceptable solutions. From a Bayesian point of view, it refers to a trade-off between the data fidelity term and the regularization term specified independently from a particular observation configuration.

Overall, for a given inverse problem, it is necessary to specify accurately the observation model, to define an appropriate regularization term and to implement a suitable optimization method. These three steps are the key elements in solving an inverse problem using variational approaches. An important issue is that there is no guarantees in general that the solution of the resulting optimization problem is actually the true state from which the observed data have been generated (see Fig.1). This mismatch can have several causes: discrepancy between the forward model and the actual data generation process, unsuitable regularization term or optimization algorithm stuck in a local minimum.

Here, through a focus on interpolation (inpainting) problems, we show that deep learning frameworks provide new means for the design and implementation of such variational models. We introduce an end-to-end framework to jointly learn in a supervised setting the variational cost and the associated solver. This is done by formulating this joint learning problem as a meta-learning problem where the regularization term is stated as a neural network. From the defined variational representation, we design a learnable gradient-based algorithm (in an inner solver loop), whose iterations are implemented through a recurrent neural network. An outer optimization loop is performed simultaneously via automatic differentiation to minimize the reconstruction error on the true parameters, thus learning an adapted variational cost and solver at the same time. As illustrated in Fig.1, the learned regularization term is therefore no longer independent of the generative model and the available observation data, and the associated solver is adapted to the configuration at hand.

Our main contributions are as follows: (i) a versatile end-to-end framework for solving interpolation problems by jointly learning the variational cost and the corresponding solver; (ii) experiments showing that learned solvers can greatly speed up and improve the interpolation performance compared to a standard gradient descent for predefined generative models; (iii) experiments showing that the joint learning of the variational representation and the associated solver may further improve the reconstruction performance, including when the true forward model is known.

This paper is organized as follows. Section 2 introduces the problem statement. We detail the proposed end-to-end

---

\*This work was supported by LEFE program (LEFE MANU project IA-OAC), CNES (grant SWOT-DIEGO) and ANR Projects Melody and OceaniX. It benefited from HPC and GPU resources from Azure (Microsoft EU Ocean awards) and from GENCI-IDRIS (Grant 2020-101030).

framework in Section 3. We report results in Section 4 and concluding statements in Section 5.

## 2. PROBLEM STATEMENT AND RELATED WORK

This work addresses the resolution of ill-posed inverse problems stated as the minimization of a variational cost [1]

$$\hat{x} = \arg \min_{x \in \mathcal{X}} U_O(x, y) + U_R(x) \quad (1)$$

where  $x$  is the unknown state,  $y$  an observation,  $U_O$  the observation or data-fidelity term and  $U_R$  the regularization term. The role of the latter is to constrain the inversion of the observation model to a realistic set of solutions.

It may be noticed that, only for very specific parameterizations of the above observation and regularization terms, the minimization of the variational cost truly retrieves the true state  $x^{true}$  associated with observation  $y$ , or even minimizes the reconstruction error of the true state. For a given observation model  $U_O$ , one would theoretically expect to retrieve some optimal parameterization  $\Phi$  of regularization term  $U_R$  such that the reconstruction error for the true state is truly minimized. This typically leads to considering the following bi-level optimization problem [2]

$$\begin{aligned} & \arg \min_{\Phi} \sum_n \|x_n^{true} - \tilde{x}_n\|^2 \\ & \text{s.t. } \forall n, \tilde{x}_n = \arg \min_{x_n} U(x_n, y_n) \end{aligned} \quad (2)$$

where  $U(x, y) = U_O(x, y) + U_R(x)$  and  $\Phi$  refers to the parameterization of regularization term  $U_R$ . One may consider the direct minimization of this bi-level optimization criterion w.r.t. both  $\Phi$  and  $\{x_n\}_n$ . Here, we investigate a different approach where we define an end-to-end neural-network architecture to solve the inner optimization based on the considered variational formulation. As described in the next Section, we state the inner optimization as the output of a learnable iterative solver, which exploits as input the gradient of variational cost  $U$ . As such, we may distinguish the observable and learnable variational cost  $U$ , which can be computed from the available observation dataset for any state  $x$ , and the target loss to be optimized, which includes here the reconstruction error for the true state. To our knowledge, no related previous works, e.g. [3, 4], considers such learnable iterative solvers which exploits the automatic differentiation of the learnable variational cost.

As an illustration, we focus on interpolation issues such that the observation term  $U_O(x, y)$  writes

$$U_O(x, y) = \|x - y\|_{\Omega}^2 \quad (3)$$

where  $\Omega$  is the domain on which observation  $y$  is actually available. This applies to n-dimensional tensors, such as scalar and multivariate signals, images and image series. For instance, for a scalar image,  $\Omega$  refers to a spatial domain and  $\|x - y\|_{\Omega}^2$  is computed as  $\int_{\Omega} x(p)^2 dp$ . This formulation holds for a variety of applications, including among others inpainting problems [5], matrix completion [6] and data assimilation problems [7].

## 3. PROPOSED END-TO-END FRAMEWORK

This section presents the proposed end-to-end framework for the joint learning of a variational representation and of the associated solver with a view to solving an interpolation problem. We introduce in Section 3.1 the considered parameterizations  $\Phi$  for regularization term  $U_R$  in (1), the proposed end-to-end architecture in Section 3.2 and the learning strategy in Section 3.3.

### 3.1. NN parameterization for regularization term $U_R$

Here, the regularization term  $U_R$  is parameterized as

$$U_R(x) = \lambda \rho(\|x - \Phi(x)\|) \quad (4)$$

where  $\Phi$  is a neural network operator and  $\rho$  a scalar-valued function, typically the square function or the absolute value. This parameterization relates to a variety of classical regularization terms for the resolution terms. Among others, we may cite gradient norms [1, 8] where  $\Phi$  relates to the discretization of the gradient operator onto the image grid, dictionary-based terms [9, 10] or Markov Random Fields [11, 12]. Operator  $\Phi$  may be interpreted as a projection operator, which encodes the manifold in which state  $x$  lives. In our experiments, we set  $\rho(u) = u^2$ . Other parameterizations could be considered as well, including trainable ones.

In the neural network (NN) literature, auto-encoder architectures naturally arise as possible parameterizations for operator  $\Phi$ . Such architectures impose that state  $x$  may be represented by a lower-dimensional state according to some possibly non-linear mapping. This is similar to the dictionary-based setting. The constraint that  $x$  is fully described by a lower-dimensional state may appear restrictive. For instance, it can be expected to result in smoothing out fine-scale patterns when considering image inpainting issues. One may then rather consider any NN architecture such that input and output states have the same dimension. This includes any previously proposed architecture to solve inverse problems [13, 14, 15]. Here, we focus on U-Net architectures, initially introduced for image segmentation purposes [16]. U-Net architectures embeds multi-scale representations, which are particularly appealing to address interpolation issues. Here, we consider a two-scale U-Net-like architectures such that

$$\Phi(x) = Up(\Phi_1(Dw(x))) + \Phi_2(x) \quad (5)$$

where  $Up$  and  $Dw$  are upsampling (ConvTranspose layer) and downsampling (AveragePooling layer) operators.  $\Phi_{1,2}$  are CNNs combining convolutional layers and ReLU activations. We may point out that the resulting regularization term may be regarded as Gibbs random field prior where the cliques are implicitly specified through the kernels of the convolution layers [11].

When considering space-time interpolation issues for physical processes, we may also exploit physics-informed

parameterizations for operator  $\Phi$ . Assuming that  $x$  is a time-dependent process governed by an Ordinary or Partial Differential Equation (ODE/PDE) such that

$$\frac{dx(t)}{dt} = \phi(x(t)) \quad (6)$$

we may define  $\Phi$  as the associated flow operator

$$\Phi(x)(t) = x(t - \Delta) + \int_{t-\Delta}^t \phi(x(u)) du \quad (7)$$

with  $\Delta$  the considered integration time step. The resulting NN architecture involves the implementation of a numerical integration scheme [17, 18]. Here, as in [19], we consider a fourth-order Runge-Kutta integration scheme for a given NN parameterization of ODE operator  $\phi$ .

### 3.2. End-to-end architecture

We state the resolution of the considered inverse problem as the implementation of an iterative gradient-based solver from variational cost  $U$  (1). We assume that we are provided as inputs an observation  $y$ , the associated observation domain  $\Omega$  and some initialization  $x^{(0)}$ . Following meta-learning schemes [20], we investigate an LSTM-based solver. More precisely, the  $k^{th}$  update of the proposed iterative solver is given by

$$\begin{cases} g^{(k+1)} &= LSTM[\alpha \cdot \nabla_x U_\Phi(x^{(k)}, y, \Omega), h(k), c(k)] \\ x^{(k+1)} &= x^{(k)} - \mathcal{H}(g^{(k+1)}) \end{cases} \quad (8)$$

where  $g^{(k+1)}$  is the output of an LSTM cell,  $h(k), c(k)$  the internal states of the LSTM cell,  $\alpha$  a normalization scalar and  $\mathcal{H}$  a linear or convolutional mapping. At each iteration, the LSTM updates its internal states based on gradient  $\nabla_x U(x^{(k)}, y, \Omega)$ , which automatically derives from the automatic differentiation of variational cost  $U$  w.r.t. state  $x$ .

Overall, the proposed end-to-end architecture implements  $K$  steps of the proposed LSTM iterative updates. Let us denote by  $\Psi_{\Phi, \Gamma}(x^{(0)}, y, \Omega)$  the output of the resulting  $K$ -step gradient solver using as inputs initialization  $x^{(0)}$ , observation  $y$  and observed domain  $\Omega$ . We may point that one may consider other types of solvers, for instance a fixed-step gradient descent as in [4], a conjugate gradient descent [3] or fixed-point solvers [21]. Experimentally, the proposed LSTM-based solver has been proven more efficient and has been considered in this study. Regarding the complexity of this end-to-end architecture, it involves the parameters of operator  $\Phi$  for regularization term  $U_R$ , weight  $\lambda$  and the parameters of the gradient-based LSTM solvers, which comprise the parameters of a LSTM cell, scalar parameter  $\alpha$  and linear operator  $\mathcal{H}$ .

### 3.3. Learning scheme

We benefit from the considered end-to-end architecture to learn jointly all parameters, that is to say that we jointly learn

the variational cost  $U$  and the associated solver so that we minimize the reconstruction error as targeted in (2). As such, we may learn a regularization term adapted to the available observation setting. More precisely, the considered training loss decomposes as follows

$$\begin{aligned} \mathcal{L} &= \nu_1 \sum_n \|x_n - \tilde{x}_n\|^2 + \nu_2 \sum_n \|x_n - \Phi(x_n)\|^2 \\ &+ \nu_3 \sum_n \|\tilde{x}_n - \Phi(\tilde{x}_n)\|^2 \end{aligned} \quad (9)$$

with  $\nu_{1,2,3}$  weighing parameters. In this loss, we complement the minimization of the reconstruction error with the minimization of the projection error  $x - \Phi(x)$  for both the true states and the reconstructed ones. The latter terms are regarded as regularization terms to better constrain the training scheme so that the learnt regularization term leads to low energy values for both the true states and the reconstructed ones. Thus, to some extent, these additional constraints may be related to energy-based GAN formulations [22].

For a given end-to-end architecture  $\Psi_{\Phi, \Gamma}$ , we apply stochastic optimizers such as Adam to jointly learn all parameters from the minimization of the above learning loss. Experimentally, we notice that the training phase depicts better convergence features if we gradually increase the number of iterations of the LSTM solvers (typically, from 1 to 20 iterations). Weights  $\nu_{1,2,3}$  were empirically set to 1., 0.05 and 0.05.

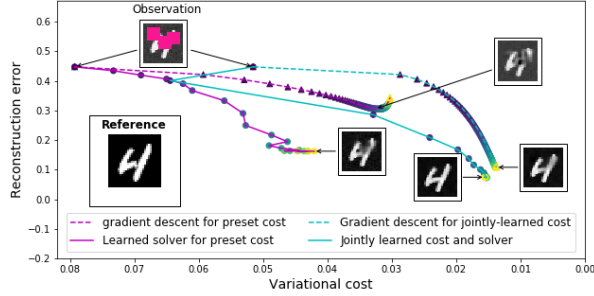
Model $\Phi$	Joint learning	Solver	R-Score	P-score
DICT	No	OMP	0.25	0.21
	No	Lasso	0.20	0.21
PCA	No	FSGD	0.39	<b>0.12</b>
	No	LSTM-S	0.15	<b>0.12</b>
U-Net	No	LSTM-S	0.21	< 1e-3
	Yes	LSTM-S	<b>0.09</b>	0.02

**Table 1. MNIST experiment:** We report the performance of different inpainting methods based on the considered variational formulation with a fixed-step gradient descent (FSGD) and a trainable LSTM solver (LSTM-S). Regarding the parameterization of the regularization term, we consider a linear PCA and a two-scale U-Net architecture. For benchmarking purposes, we also include two sparse coding schemes (DICT) adapted from [10]. We refer the reader to the main text for the description of the performance score and additional information on the considered schemes.

## 4. RESULTS

### 4.1. MNIST data

We first report image inpainting experiments for MNIST data, especially to provide a comparison with dictionary and auto-encoder priors [9, 10]. We simulate observed images with



**Fig. 1. Solvers’ energy pathways:** we depict the energy pathways vs. the reconstruction error for a mini-batch for different solvers (FSGD,– and LSTM-S,–) for PCA (cyan) and U-Net (magenta) parameterizations of variational cost  $U$ . The U-Net regularization is learnt jointly to the LSTM solver.

3 randomly sampled  $9 \times 9$  gaps and an additive Gaussian noise of variance 0.1 for the normalized MNIST dataset. In these experiments, we test a dense auto-encoder ( $\approx 500,000$  parameters) and a two-scale U-Net parameterization ( $\approx 550,000$  parameters) for operator  $\Phi$ . In both cases, the considered LSTM solver involves a convolutional LSTM cell with 5 channels. For benchmarking purposes, we include sparse coding approaches using OMP (Orthogonal Matching Pursuit) and Lasso solutions [10]. We adapt these schemes for interpolation issues through iterative coding-decoding steps. For each method, we evaluate the normalized mean square error (NMSE) of the reconstruction over the entire image domain (R-Score) and NMSE of the projection error  $x - \Phi(x)$  for true states  $x$  (P-Score). We refer to [23] for a more detailed description of our experimental setting.

From Tab.1 and additional experiments in [23], a first key result is the improvement of the reconstruction error by the learnt LSTM solver compared with a FSGD minimization of the variational cost with a pre-trained regularization term (e.g., R-Score of 0.54 vs. 1.09 for the PCA version of operator  $\Phi$ ). A second important result is the greater gain issued from the joint learning of operator  $\Phi$  and of the LSTM solver (0.15 vs. 0.09 for the best R-scores). As illustrated through numerous MNIST examples in [23], this leads to a noticeable visual improvement. The learnt solver (Fig. 1) identifies much better descent pathways for the reconstruction error from the differentiation of the learnt variational cost than the direct minimization of this cost, while only considering 15 iterations of a gradient-based update. This is regarded as a key feature of LSTM solver to go beyond a pure gradient descent algorithms as in [3, 4].

#### 4.2. Multivariate signals governed by ODEs

The second case-study addresses multivariate signals governed by an ODE so that we are provided with a theoretical ground-truth for the true representation of the data. Here, we consider

Model $\Phi$	Joint learning	Solver	R-Score	P-score
ODE-RK4	No	FSGD	5.65e-2	<b>6.70e-7</b>
	No	LSTM-S	4.46e-2	<b>6.70e-7</b>
U-Net	Yes	LSTM-S	<b>1.94e-2</b>	3.70e-3

**Table 2. Experiment with multivariate signals governed by ODEs:** we report for Lorenz-96 dynamics reconstruction experiments from noisy and under-sampled observations. We let the reader refer to the main text for the details on the experimental setting and the parameterizations considered for operator  $\Phi$  in (1). The evaluation procedure is similar to Tab.1.

Lorenz-96 dynamics, which are widely used for benchmarking experiments in geoscience and data assimilation. Lorenz-96 dynamics involve a bilinear ODE which lead to chaotic patterns under the considered parametrization. The generated dataset involves 2000 time series with 200 time steps as training dataset and 256 as test dataset. Observation data are generated with a sampling every 4 time steps, for only 20 randomly selected components over the 40 components of the state, and a Gaussian additive noise with a variance of 2. We test the proposed framework with the parameterization of operator  $\Phi$  as the flow operator of the true ODE with a fourth-order Runge-Kutta scheme (RK4) and with a two-scale U-Net parameterization for operator  $\Phi$ . We refer to [23] for additional information on this experimental setting.

A surprising result of these experiments is that the ODE-based prior with the true physical model does not lead to the best reconstruction performance. The joint learning of the U-Net parameterization of operator  $\Phi$  and of the LSTM solver leads a relative decrease of more than 50% of the reconstruction error. Similarly to MNIST case-study, the LSTM solver also clearly outperforms the FSGD minimization of the variational cost.

## 5. CONCLUSION

To the best of our knowledge, this work provides the first illustration of the relevance of a joint end-to-end learning of a regularization term of a variational formulation and of an iterative gradient-based LSTM solver with a view to optimizing a reconstruction performance. Compared with previous works, e.g. [3, 4], our gradient-based LSTM solver may apply to any NN-based formulation of the regularization term and fully benefits from optimizer learning strategies beyond simpler gradient descent algorithms. They provide new means to design and implement variational formulations and solvers optimized for a given reconstruction performance. This work may open new research avenues for numerous applications beyond interpolation issues, possibly including the extension to trainable observation models.

## 6. REFERENCES

- [1] G. Aubert and P. Kornprobst, *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Springer, 2006, vol. 147.
- [2] G. Holler, K. Kunisch, and R. C. Barnard, “A bilevel approach for parameter learning in inverse problems,” *Inverse Problems*, vol. 34, no. 11, p. 115012, 2018.
- [3] H. K. Aggarwal, M. P. Mani, and M. Jacob, “MoDL: Model-Based Deep Learning Architecture for Inverse Problems,” *IEEE TMI*, vol. 38, no. 2, pp. 394–405, 2019.
- [4] E. Kobler, A. Effland, K. Kunisch, and T. Pock, “Total Deep Variation for Linear Inverse Problems,” *arXiv:2001.05005*, 2020.
- [5] Q. Cheng, H. Shen, L. Zhang, and P. Li, “Inpainting for Remotely Sensed Images With a Multichannel Nonlocal Total Variation Model,” *IEEE TGRS*, vol. 52, no. 1, pp. 175–187, 2014.
- [6] E. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Communications of the ACM*, vol. 55, no. 6, pp. 111–119, Jun. 2012.
- [7] J. Blum, F.-X. Le Dimet, and I. M. Navon, “Data assimilation for geophysical fluids,” in *Handbook of numerical analysis*. Elsevier, 2009, vol. 14, pp. 385–441.
- [8] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel, “Axioms and fundamental equations of image processing,” *Archive for rational mechanics and analysis*, vol. 123, no. 3, pp. 199–257, 1993.
- [9] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE TIP*, vol. 15, no. 12, pp. 3736–3745, 2006, publisher: IEEE.
- [10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *ICML*, 2009, pp. 689–696.
- [11] P. Perez, “Markov random fields and images,” *CWI Quarterly*, pp. 413–437, 1998.
- [12] D. Geman, “Random fields and inverse problems in imaging,” *LNM*, vol. 1427, pp. 113–193, 1990.
- [13] T. Meinhardt, M. Moller, C. Hazirbas, and D. Cremers, “Learning proximal operators: Using denoising networks for regularizing inverse imaging problems,” in *IEEE CVPR*, 2017, pp. 1781–1790.
- [14] S. Lunz, O. Öktem, and C.-B. Schönlieb, “Adversarial regularizers in inverse problems,” in *NIPS*, 2018, pp. 8507–8516.
- [15] J. Xie, L. Xu, and E. Chen, “Image Denoising and Inpainting with Deep Neural Networks,” in *NIPS*, 2012, pp. 341–349.
- [16] Cicek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: learning dense volumetric segmentation from sparse annotation.” Springer, 2016, pp. 424–432.
- [17] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019.
- [18] T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural Ordinary Differential Equations,” in *NIPS*, 2018, pp. 6571–6583.
- [19] R. Fablet, S. Ouala, and C. Herzet, “Bilinear residual Neural Network for the identification and forecasting of dynamical systems,” in *EUSIPCO*, Rome, Italy, Sep. 2018, pp. 1–5. [Online].
- [20] M. Andrychowicz, M. Denil, S. Gomez, M. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. de Freitas, “Learning to learn by gradient descent by gradient descent,” in *NIPS*, 2016, pp. 3981–3989.
- [21] R. Fablet, L. Drumetz, and F. Rousseau, “End-to-end learning of energy-based representations for irregularly-sampled signals and images,” *arXiv preprint arXiv:1910.00556*, 2019.
- [22] J. Zhao, M. Mathieu, and Y. LeCun, “Energy-based Generative Adversarial Network,” *arXiv:1609.03126*, 2017.
- [23] R. Fablet, L. Drumetz, and F. Rousseau, “Joint learning of variational representations and solvers for inverse problems with partially-observed data,” *arXiv:2006.03653*, 2020.