



**HAL**  
open science

## Solving Inverse Problems by Joint Posterior Maximization with a VAE Prior

Mario Gonzalez, Andrés Almansa, Mauricio Delbracio, Pablo Musé, Pauline  
Tan

► **To cite this version:**

Mario Gonzalez, Andrés Almansa, Mauricio Delbracio, Pablo Musé, Pauline Tan. Solving Inverse Problems by Joint Posterior Maximization with a VAE Prior. (KHIPU) Latin American Meeting in Artificial Intelligence, Nov 2019, Montevideo, Uruguay. hal-02364354

**HAL Id: hal-02364354**

**<https://hal.science/hal-02364354>**

Submitted on 14 Nov 2019

**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.



Distributed under a Creative Commons Attribution 4.0 International License

---

# SOLVING INVERSE PROBLEMS BY JOINT POSTERIOR MAXIMIZATION WITH A VAE PRIOR

**Mario GONZÁLEZ**  
DMEL, CenUR LN  
Universidad de la República  
Salto, Uruguay  
mgonzalez@unorte.edu.uy

**Andrés ALMANSA**  
MAP5 - CNRS & Université de Paris  
Paris, France  
andres.almansa@parisdescartes.fr

**Mauricio DELBRACIO & Pablo MUSÉ**  
IIE - Facultad de Ingeniería  
Universidad de la República  
Montevideo, Uruguay  
{mdelbra, pmuse}@fing.edu.uy

**Pauline TAN**  
LJLL - Sorbonne Université  
Paris, France  
pauline.tan@sorbonne-universite.fr

## ABSTRACT

In this paper we address the problem of solving ill-posed inverse problems in imaging where the prior is a neural generative model. Specifically we consider the decoupled case where the prior is trained once and can be reused for many different log-concave degradation models without retraining. Whereas previous MAP-based approaches to this problem lead to highly non-convex optimization algorithms, our approach computes the joint (space-latent) MAP that naturally leads to alternate optimization algorithms and to the use of a stochastic encoder to accelerate computations. The resulting technique is called JPMAP because it performs Joint Posterior Maximization using an Autoencoding Prior. We show theoretical and experimental evidence that the proposed objective function is quite close to bi-convex. Indeed it satisfies a weak bi-convexity property which is sufficient to guarantee that our optimization scheme converges to a stationary point. Experimental results also show the higher quality of the solutions obtained by our JPMAP approach with respect to other non-convex MAP approaches which more often get stuck in spurious local optima.

---

CONTENTS

<b>1</b>	<b>Introduction and Related Work</b>	<b>3</b>
<b>2</b>	<b>Joint Posterior Maximization with Autoencoding Prior (JPMAP)</b>	<b>4</b>
2.1	Variational Autoencoders as Image Priors . . . . .	4
2.2	Alternate Joint Posterior Maximization . . . . .	5
<b>3</b>	<b>Experimental results</b>	<b>7</b>
3.1	AutoEncoder and dataset . . . . .	7
3.2	Empirical Validation of Assumption 2 . . . . .	8
3.3	Image restoration experiments . . . . .	10
<b>4</b>	<b>Conclusions and Future work</b>	<b>10</b>
<b>5</b>	<b>Acknowledgements</b>	<b>11</b>
<b>A</b>	<b>Convergence Proofs</b>	<b>13</b>

---

## 1 INTRODUCTION AND RELATED WORK

General inverse problems in imaging consist in estimating a clean image  $\mathbf{x} \in \mathbb{R}^n$  from noisy, degraded measurements  $\mathbf{y} \in \mathbb{R}^m$ . In many cases the degradation model is known and its conditional density

$$p_{Y|X}(\mathbf{y} | \mathbf{x}) \propto e^{-F(\mathbf{x}, \mathbf{y})}$$

is log-concave with respect to  $\mathbf{x}$ . To illustrate this, let us consider the case where the negative log-conditional is quadratic with respect to  $\mathbf{x}$

$$F(\mathbf{x}, \mathbf{y}) = \frac{1}{2\sigma^2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2. \quad (1)$$

This boils down to a linear degradation model that takes into account degradations such as, white Gaussian noise, blur, and missing pixels. When the degradation operator  $\mathbf{A}$  is non-invertible or ill-conditioned, or when the noise level  $\sigma$  is high, obtaining a good estimate of  $\mathbf{x}$  requires prior knowledge on the image, given by  $p_X(\mathbf{x}) \propto e^{-G(\mathbf{x})}$ . Variational and Bayesian methods in imaging are extensively used to derive MMSE or MAP estimators,

$$\hat{\mathbf{x}}_{\text{MAP}} = \arg \max_{\mathbf{x}} p_{X|Y}(\mathbf{x} | \mathbf{y}) = \arg \min_{\mathbf{x}} \{F(\mathbf{x}, \mathbf{y}) + G(\mathbf{x})\} \quad (2)$$

based on explicit priors like total variation (Chambolle, 2004; Louchet and Moisan, 2013; Pereyra, 2016; Rudin et al., 1992), or learning-based priors like patch-based Gaussian mixture models (Teodoro et al., 2018; Zoran and Weiss, 2011).

**Neural network regression.** Since neural networks (NN) showed their superiority in image classification tasks (Krizhevsky et al., 2012) researchers started to look for ways to use this tool to solve inverse problems too. The most straightforward attempts employed neural networks as regressors to learn a risk minimizing mapping  $\mathbf{y} \mapsto \mathbf{x}$  from many examples  $(\mathbf{x}_i, \mathbf{y}_i)$  either agnostically (Dong et al., 2014; Gao et al., 2019; Gharbi et al., 2016; Schwartz et al., 2018; Zhang et al., 2017a, 2018) or including the degradation model in the network architecture via unrolled optimization techniques (Chen and Pock, 2017; Diamond et al., 2017; Gilton et al., 2019; Gregor and LeCun, 2010).

**Implicitly decoupled priors.** The main drawback of neural networks regression is that they require to retrain the neural network each time a single parameter of the degradation model changes. To avoid the need for retraining, another family of approaches seek to *decouple* the NN-based learned image prior from the degradation model. A popular approach within this methodology are *plug & play* methods. Instead of directly learning the log-prior  $-\log p_X(\mathbf{x}) = G(\mathbf{x}) + C$ , these methods seek to learn an approximation of its gradient  $\nabla G$  (Bigdeli and Zwicker, 2017; Bigdeli et al., 2017) or proximal operator  $\text{prox}_G$  (Chan et al., 2017; Meinhardt et al., 2017; Ryu et al., 2019; Zhang et al., 2017b), by replacing it by a denoising NN. Then, these approximations are used in an iterative optimization algorithm to find the corresponding MAP estimator in equation (2).

**Explicitly decoupled priors.** Plug & play approaches became very popular because of their convenience but obtaining convergence guarantees under realistic conditions is quite challenging. Indeed, the actual prior is unknown, and the existence of a density whose proximal operator is well approximated by a neural denoiser is most often not guaranteed (Reehorst and Schniter, 2018), unless the denoiser is retrained with specific constraints (Gupta et al., 2018; Ryu et al., 2019; Shah and Hegde, 2018). In our experience these inconsistencies may result in sub-optimal solutions that introduce undesirable artifacts. It is tempting to use neural networks to learn an explicit prior for images. For instance one could use a generative adversarial network (GAN) to learn a generative model for  $X = G(Z)$  with  $Z \sim N(0, I)$  a latent variable. Nevertheless, current attempts (Bora et al., 2017) to use such a generative model as a prior to estimate  $\hat{\mathbf{x}}_{\text{MAP}}$  in (2) lead to a highly non-convex optimization problem. Indeed, the posterior writes

$$\begin{aligned} p_{X|Y}(\mathbf{x} | \mathbf{y}) &= \int p_{Y|X}(\mathbf{y} | \mathbf{x}) p_{X|Z}(\mathbf{x} | \mathbf{z}) p_Z(\mathbf{z}) d\mathbf{z} \\ &= p_{Y|X}(\mathbf{y} | \mathbf{x}) \int p_{X|Z}(\mathbf{x} | \mathbf{z}) p_Z(\mathbf{z}) d\mathbf{z} \\ &= p_{Y|X}(\mathbf{y} | \mathbf{x}) p_Z(G^{-1}(\mathbf{x})), \end{aligned}$$

where the last equality follows from  $p_{X|Z}(\mathbf{x} | \mathbf{z}) = \delta(\mathbf{x} - G(\mathbf{z}))$ . Therefore,

$$\begin{aligned} \hat{\mathbf{x}}_{\text{MAP}} &= \arg \max_{\mathbf{x}} p_{X|Y}(\mathbf{x} | \mathbf{y}) \\ &= G \left( \arg \max_{\mathbf{z}} \{p_{Y|X}(\mathbf{y} | G(\mathbf{z})) p_Z(\mathbf{z})\} \right) \\ &= G \left( \arg \min_{\mathbf{z}} \left\{ F(G(\mathbf{z}), \mathbf{y}) + \frac{1}{2} \|\mathbf{z}\|^2 \right\} \right). \end{aligned} \quad (3)$$

Convergence guarantees for this problem are of course extremely difficult to establish, and our experimental results in Section 3 on the CSGM approach by Bora et al. (2017) confirm this.

A possible workaround to avoid minimization over  $\mathbf{z}$  could consist in learning an encoder network  $E$  (inverse of  $G$ ) to directly minimize over  $\mathbf{x}$ . This does not help either because an intractable term appears when we develop

$$p_X(\mathbf{x}) = p_Z(E(\mathbf{x})) \det \left( \left( \frac{\partial E}{\partial \mathbf{x}} \right)^T \left( \frac{\partial E}{\partial \mathbf{x}} \right) \right)^{1/2} \delta_{G(E(\mathbf{x}))=\mathbf{x}}(\mathbf{x})$$

via the push-forward measure.

**Proposed method: Joint MAP<sub>x,z</sub>.** In order to overcome the limitations of the previous approach, in this work we show that the numerical solution of the explicitly decoupled approach is greatly simplified when we introduce two modifications:

- Given the noisy, degraded observation  $y$ , we maximize the joint posterior density  $p_{X,Z|Y}(\mathbf{x}, \mathbf{z} | \mathbf{y})$  instead of the usual posterior  $p_{X|Y}(\mathbf{x} | \mathbf{y})$ ;
- We use both a (deterministic or stochastic) generator and a stochastic encoder.

In addition, we show that for a particular choice of the stochastic decoder the maximization of the joint log-posterior becomes a bi-concave optimization problem or approximately so. And in that case, an extension of standard bi-convex optimization results (Gorski et al., 2007) show that the algorithm converges to a stationary point that is a partially global optimum.

The remainder of this paper is organized as follows. In Section 2 we derive a model for the joint conditional posterior distribution of space and latent variables  $\mathbf{x}$  and  $\mathbf{z}$ , given the observation  $\mathbf{y}$ . This model makes use of a generative model, more precisely a VAE with Gaussian decoder. We then propose an alternate optimization scheme to maximize for the joint posterior model, and state convergence guarantees. Section 3 presents first a set of experiments that illustrates the convergence properties of the optimization scheme. We then test our approach on classical image inverse problems, and compare its performance with state-of-the-art methods. Concluding remarks are presented in Section 4.

## 2 JOINT POSTERIOR MAXIMIZATION WITH AUTOENCODING PRIOR (JPMAP)

### 2.1 VARIATIONAL AUTOENCODERS AS IMAGE PRIORS

In this work we construct an image prior based on a variant of the variational autoencoder (Kingma and Welling, 2013) (VAE). Like GANs and other generative models, VAEs allow to obtain samples from an unknown distribution  $p_X$  by taking samples of a latent variable  $Z$  with known distribution  $\mathcal{N}(0, I)$ , and feeding these samples through a learned generator network. For VAEs the generator (or decoder) network with parameters  $\theta$  can be deterministic or stochastic and it learns

$$p_{X|Z}(\mathbf{x} | \mathbf{z}) = p_\theta(\mathbf{x} | \mathbf{z}),$$

whereas the stochastic encoder network with parameters  $\phi$ , approximates

$$p_{Z|X}(\mathbf{z} | \mathbf{x}) \approx q_\phi(\mathbf{z} | \mathbf{x}).$$

Given a VAE we could plug in the approximate prior

$$p_X(\mathbf{x}) = \frac{p_\theta(\mathbf{x} | \mathbf{z}) p_Z(\mathbf{z})}{p_{Z|X}(\mathbf{z} | \mathbf{x})} \approx \frac{p_\theta(\mathbf{x} | \mathbf{z}) p_Z(\mathbf{z})}{q_\phi(\mathbf{z} | \mathbf{x})} \quad (4)$$

in (2) to obtain the corresponding MAP estimator, but this leads to a numerically difficult problem to solve. Instead, we propose to maximize the joint posterior  $p_{X,Z|Y}(\mathbf{x}, \mathbf{z} | \mathbf{y})$  over  $(\mathbf{x}, \mathbf{z})$  which is equivalent to minimize

$$J_1(\mathbf{x}, \mathbf{z}) := -\log p_{X,Z|Y}(\mathbf{x}, \mathbf{z} | \mathbf{y}) = -\log p_{Y|X,Z}(\mathbf{y} | \mathbf{x}, \mathbf{z}) p_\theta(\mathbf{x} | \mathbf{z}) p_Z(\mathbf{z}) \quad (5)$$

$$= F(\mathbf{x}, \mathbf{y}) + H_\theta(\mathbf{x}, \mathbf{z}) + \frac{1}{2} \|\mathbf{z}\|^2. \quad (6)$$

Note that the first term is quadratic in  $\mathbf{x}$  (assuming (1)), the third term is quadratic in  $\mathbf{z}$  and all the difficulty lies in the coupling term  $H_\theta(\mathbf{x}, \mathbf{z}) = -\log p_\theta(\mathbf{x} | \mathbf{z})$ . For Gaussian decoders (Kingma and Welling, 2013), the latter can be written as

$$H_\theta(\mathbf{x}, \mathbf{z}) = \frac{1}{2} \left( n \log(2\pi) + \log \det \Sigma_\theta(\mathbf{z}) + \|\Sigma_\theta^{-1/2}(\mathbf{z})(\mathbf{x} - \boldsymbol{\mu}_\theta(\mathbf{z}))\|^2 \right). \quad (7)$$

which is also convex in  $\mathbf{x}$ . Hence, minimization with respect to  $\mathbf{x}$  takes the convenient closed form:

$$\arg \min_{\mathbf{x}} J_1(\mathbf{x}, \mathbf{z}) = (\mathbf{A}^T \mathbf{A} + \sigma^2 \Sigma_\theta^{-1}(\mathbf{z}))^{-1} (\mathbf{A}^T \mathbf{y} + \sigma^2 \Sigma_\theta^{-1}(\mathbf{z}) \boldsymbol{\mu}_\theta(\mathbf{z})). \quad (8)$$

Unfortunately the coupling term  $H$  and hence  $J_1$  is a priori non-convex in  $\mathbf{z}$ . As a consequence the  $\mathbf{z}$ -minimization problem

$$\arg \min_{\mathbf{z}} J_1(\mathbf{x}, \mathbf{z}) \quad (9)$$

is a priori more difficult. However, for Gaussian encoders, VAEs provide an approximate expression for this coupling term which is quadratic in  $\mathbf{z}$ . Indeed, given the equivalence

$$p_\theta(\mathbf{x} | \mathbf{z}) p_Z(\mathbf{z}) = p_{X,Z}(\mathbf{x}, \mathbf{z}) = p_{Z|X}(\mathbf{z} | \mathbf{x}) p_X(\mathbf{x}) \approx q_\phi(\mathbf{z} | \mathbf{x}) p_X(\mathbf{x})$$

we have that

$$H_\theta(\mathbf{x}, \mathbf{z}) + \frac{1}{2} \|\mathbf{z}\|^2 \approx K_\phi(\mathbf{x}, \mathbf{z}) - \log p_X(\mathbf{x}). \quad (10)$$

where  $K_\phi(\mathbf{x}, \mathbf{z}) = -\log q_\phi(\mathbf{z} | \mathbf{x})$ . Therefore, this new coupling term becomes

$$K_\phi(\mathbf{x}, \mathbf{z}) = -\log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{x}), \Sigma_\phi(\mathbf{x})) \quad (11)$$

$$= \frac{1}{2} \left( k \log(2\pi) + \log \det \Sigma_\phi(\mathbf{x}) + \|\Sigma_\phi^{-1/2}(\mathbf{x})(\mathbf{z} - \boldsymbol{\mu}_\phi(\mathbf{x}))\|^2 \right), \quad (12)$$

which is quadratic in  $\mathbf{z}$ . This provides an approximate expression for the energy (5) that we want to minimize, namely

$$J_2(\mathbf{x}, \mathbf{z}) := F(\mathbf{x}, \mathbf{y}) + K_\phi(\mathbf{x}, \mathbf{z}) - \log p_X(\mathbf{x}) \approx J_1(\mathbf{x}, \mathbf{z}). \quad (13)$$

This approximate functional is quadratic in  $\mathbf{z}$ , and minimization with respect to this variable yields

$$\arg \min_{\mathbf{z}} J_2(\mathbf{x}, \mathbf{z}) = \boldsymbol{\mu}_\phi(\mathbf{x}). \quad (14)$$

## 2.2 ALTERNATE JOINT POSTERIOR MAXIMIZATION

The previous observations suggest to adopt alternate scheme to minimize  $-\log p_{X,Z|Y}(\mathbf{x}, \mathbf{z} | \mathbf{y})$  in order to solve the inverse problem. We begin our presentation by assuming that the approximation of  $J_1$  by  $J_2$  is exact; then we propose an adaptation for the non-exact case and we explore its convergence properties.

To begin with we shall consider the following (strong) assumption:

**Assumption 1.** *The approximation in (13) is exact, i.e.  $J_1 = J_2$ .*

Under this assumption, the objective function is biconvex and alternate minimization takes the simple and fast form depicted in Algorithm 1, which can be shown to converge to a partial optimum, as stated in Proposition 1 below. Note that the minimization in step 2 of Algorithm 1 does not require the knowledge of the unknown term  $-\log p_X(\mathbf{x})$  in Equation (13) since it does not depend on  $\mathbf{z}$ .

**Proposition 1** (Convergence of Algorithm 1). *Under Assumption 1 we have that :*

---

**Algorithm 1** Joint posterior maximization - exact case

---

**Require:** Measurements  $\mathbf{y}$ , Autoencoder parameters  $\theta, \phi$ , Initial condition  $\mathbf{x}_0$ **Ensure:**  $\hat{\mathbf{x}}, \hat{\mathbf{z}} = \arg \max_{\mathbf{x}, \mathbf{z}} p_{X,Z|Y}(\mathbf{x}, \mathbf{z} | \mathbf{y})$ 

- 1: **for**  $n := 0$  **to** maxiter **do**
  - 2:    $\mathbf{z}_{n+1} := \arg \min_{\mathbf{z}} J_2(\mathbf{x}_n, \mathbf{z})$  // Quadratic problem in (8)
  - 3:    $\mathbf{x}_{n+1} := \arg \min_{\mathbf{x}} J_1(\mathbf{x}, \mathbf{z}_{n+1})$  // Quadratic problem in (14)
  - 4: **end for**
  - 5: **return**  $\mathbf{x}_{n+1}, \mathbf{z}_{n+1}$
- 

1. The sequence  $\{J_1(\mathbf{x}_n, \mathbf{z}_n)\}$  generated by Algorithm 1 converges monotonically when  $n \rightarrow \infty$ . The sequence  $\{(\mathbf{x}_n, \mathbf{z}_n)\}$  has at least one accumulation point.
2. All accumulation points are partial optima of  $J_1$  and they all have the same function value.

If in addition  $J_1$  is differentiable then:

3. The set of all accumulation points are stationary points of  $J_1$  and they form a connected, compact set.

The proof of this proposition is given in Appendix A, and follows the same arguments as in (Aguerebere et al., 2017; Gorski et al., 2007). Note that the third part requires that  $J_1$  be differentiable, which is the case if we use a differentiable activation function like the Exponential Linear Unit (ELU) (Clevert et al., 2016) with  $\alpha = 1$ , instead of the more common ReLU activation function.

When the autoencoder approximation in (13) is not exact (Assumption 1), the algorithm needs some additional steps to ensure that the energy we want to minimize, namely  $J_1$ , actually decreases. Nevertheless, the approximation provided by  $J_2$  is still very useful since it provides a fast and accurate heuristic to initialize the minimization of  $J_1$ . This method is presented in Algorithm 2.

---

**Algorithm 2** Joint posterior maximization - approximate case

---

**Require:** Measurements  $\mathbf{y}$ , Autoencoder parameters  $\theta, \phi$ , Initial conditions  $\mathbf{x}_0, \mathbf{z}_0$ **Ensure:**  $\hat{\mathbf{x}}, \hat{\mathbf{z}} = \arg \max_{\mathbf{x}, \mathbf{z}} p_{X,Z|Y}(\mathbf{x}, \mathbf{z} | \mathbf{y})$ 

- 1: **for**  $n := 0$  **to** maxiter **do**
  - 2:    $\mathbf{z}^0 := \text{grad descent}_{\mathbf{z}} J_1(\mathbf{x}_n, \mathbf{z})$ , starting from  $\mathbf{z} = \mathbf{z}_n$
  - 3:    $\mathbf{z}^1 := \arg \min_{\mathbf{z}} J_2(\mathbf{x}_n, \mathbf{z})$  // Quadratic problem
  - 4:    $\mathbf{z}^2 := \text{grad descent}_{\mathbf{z}} J_1(\mathbf{x}_n, \mathbf{z})$ , starting from  $\mathbf{z} = \mathbf{z}^1$
  - 5:    $\mathbf{z}_{n+1} := \arg \min_{\mathbf{z} \in \{\mathbf{z}^0, \mathbf{z}^1, \mathbf{z}^2\}} J_1(\mathbf{x}_n, \mathbf{z})$
  - 6:    $\mathbf{x}_{n+1} := \arg \min_{\mathbf{x}} J_1(\mathbf{x}, \mathbf{z}_{n+1})$  // Quadratic problem
  - 7: **end for**
  - 8: **return**  $\mathbf{x}_{n+1}, \mathbf{z}_{n+1}$
- 

Algorithm 2 provides also a useful tool for diagnostics. Indeed, the comparison of the evaluation of  $J_1(\mathbf{x}_n, \mathbf{z})$  in  $\mathbf{z}^0, \mathbf{z}^1, \mathbf{z}^2$  performed in step 5 permits to assess the evolution of the approximation of  $J_1$  by  $J_2$ .

Our experiments with Algorithm 2 (Section 3.2) show that during the first few iterations (where the approximation provided by  $J_2$  is good enough)  $\mathbf{z}^2$  reaches convergence faster than  $\mathbf{z}^0$ . After a critical number of iterations the opposite is true (the initialization provided by the previous iteration is better than the  $J_2$  approximation, and  $\mathbf{z}^0$  converges faster).

These observations suggest that a faster execution, with the same convergence properties, can be achieved by the variant in Algorithm 3.

The fastest alternative is equivalent to Algorithm 1 as long as the approximate energy minimization decreases the actual energy. When this is not the case it will take a slower route similar to Algorithm 2.

Algorithm 2 is still quite fast when  $J_2$  provides a sufficiently good approximation. Even if we cannot give a precise definition of what *sufficiently good* means, the sample comparison of  $K_\phi$  and  $H_\theta$  as functions of  $\mathbf{z}$ , displayed in Figure 1(c), shows that the approximation is fair enough in the sense that it preserves the global structure of  $J_2$ . The same behavior was observed for a large number of

---

**Algorithm 3** Joint posterior maximization - approximate case (faster version)

---

**Require:** Measurements  $\mathbf{y}$ , Autoencoder parameters  $\theta, \phi$ , Initial conditions  $\mathbf{x}_0, \mathbf{z}_0$ , iterations  $n_{\min} < n_{\max}$

**Ensure:**  $\hat{\mathbf{x}}, \hat{\mathbf{z}} = \arg \max_{\mathbf{x}, \mathbf{z}} p_{X, Z|Y}(\mathbf{x}, \mathbf{z} | \mathbf{y})$

- 1: **for**  $n := 0$  **to**  $n_{\max}$  **do**
- 2:    $\mathbf{z}^1 := \arg \min_{\mathbf{z}} J_2(\mathbf{x}_n, \mathbf{z})$  // Quadratic problem
- 3:    $\mathbf{z}^0 := \mathbf{z}_n$
- 4:    $\mathbf{z}^* := \arg \min_{\mathbf{z} \in \{\mathbf{z}^0, \mathbf{z}^1\}} J_1(\mathbf{x}_n, \mathbf{z})$
- 5:   **if**  $J_1(\mathbf{x}_n, \mathbf{z}^1) > J_1(\mathbf{x}_n, \mathbf{z}_n)$  **or**  $n > n_{\min}$  **then**
- 6:      $\mathbf{z}_{n+1} := \text{grad descent}_{\mathbf{z}} J_1(\mathbf{x}_n, \mathbf{z})$ , starting from  $\mathbf{z} = \mathbf{z}^*$
- 7:   **else**
- 8:      $\mathbf{z}_{n+1} := \mathbf{z}^1$  // Faster alternative while  $J_2$  is good enough
- 9:   **end if**
- 10:    $\mathbf{x}_{n+1} := \arg \min_{\mathbf{x}} J_1(\mathbf{x}, \mathbf{z}_{n+1})$  // Quadratic problem
- 11: **end for**
- 12: **return**  $\mathbf{x}_{n+1}, \mathbf{z}_{n+1}$

---

random tests. In particular, these simulations show that for every tested  $\mathbf{x}$ , the function  $\mathbf{z} \mapsto J_1(\mathbf{x}, \mathbf{z})$  exhibits a unique global minimizer. This justifies the following assumption (which is nevertheless much weaker than the previous Assumption 1):

**Assumption 2.**

(A) *The  $z$ -minimization algorithm  $\text{grad descent}_{\mathbf{z}} J_1(\mathbf{x}, \mathbf{z})$  converges to a global minimizer of  $\mathbf{z} \mapsto J_1(\mathbf{x}, \mathbf{z})$ , when initialized at  $\mathbf{z}^1 = \arg \min_{\mathbf{z}} J_2(\mathbf{x}, \mathbf{z})$  or at any  $\mathbf{z}$  such that  $J_1(\mathbf{x}, \mathbf{z}) \leq J_1(\mathbf{x}, \mathbf{z}^1)$ .*

(B) *The map  $\mathbf{z} \mapsto J_1(\mathbf{x}, \mathbf{z})$  has a single global minimizer.*

Under this assumption we have the following result for Algorithm 2:

**Proposition 2** (Convergence of Algorithms 2 and 3). *Under Assumption 2A we have that:*

1. *The sequence  $\{J_1(\mathbf{x}_n, \mathbf{z}_n)\}$  generated by Algorithms 2 and 3 converges monotonically when  $n \rightarrow \infty$ . The sequence  $\{(\mathbf{x}_n, \mathbf{z}_n)\}$  has at least one accumulation point.*
2. *All accumulation points are partial optima of  $J_1$  and they all have the same function value.*

*If in addition  $J_1$  is continuously differentiable and Assumption 2B holds, then:*

3. *All accumulation points are stationary points of  $J_1$ .*

The proof of this proposition is detailed in the appendix. The first two parts are similar to the proof of Proposition 1, but the last part uses a different argument. Indeed we cannot use (Gorski et al., 2007, Thm 4.9), because we do not assume here that  $J_1$  is bi-convex.

## 3 EXPERIMENTAL RESULTS

### 3.1 AUTOENCODER AND DATASET

In order to test our joint prior maximization model we first train a Variational Autoencoder like in (Kingma and Welling, 2013) on the training data of MNIST handwritten digits (Lecun et al., 1998).

The *stochastic encoder* takes as input an image  $\mathbf{x}$  of  $24 \times 24 = 784$  pixels and produces as an output the mean and (diagonal) covariance matrix of the Gaussian distribution  $q_{\phi}(z|\mathbf{x})$ , where the latent variable  $\mathbf{z}$  has dimension 12. The architecture of the encoder is composed of 4 fully connected layers with ELU activations (to preserve continuous differentiability). The sizes of the layers are as follows:

$$784 \rightarrow 500 \rightarrow 500 \rightarrow (12 + 12).$$

Note that the output is of size  $12 + 12$  in order to encode the mean and diagonal covariance matrix, both of size 12.



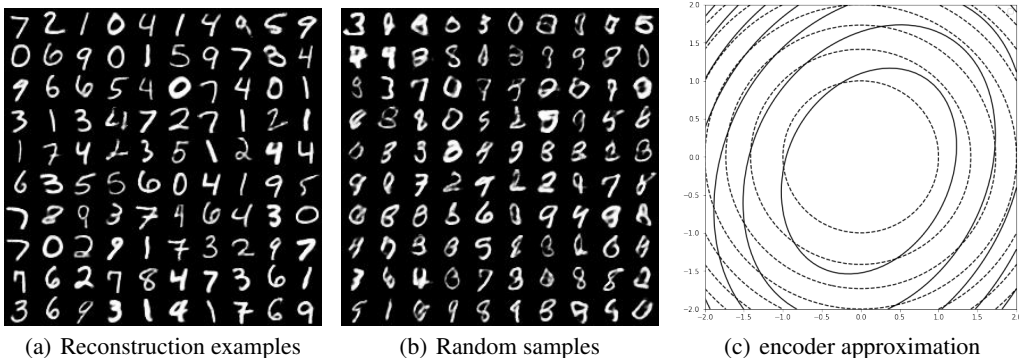


Figure 1: *Visualisation of the trained VAE on MNIST.* (a) Reconstruction examples, *i.e.*  $\mu_\theta(\mu_\phi(\mathbf{x}))$  for 100 samples of  $\mathbf{x}$  from the test set. (b) Random samples, *i.e.*  $\mu_\theta(\mathbf{z})$  for 100 random samples  $\mathbf{z} \sim \mathcal{N}(0, Id)$ . (c) *Encoder approximation:* Contour plots of  $-\log p_\theta(\mathbf{x}|\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|^2$  and  $-\log q_\phi(\mathbf{z}|\mathbf{x})$  for a fixed  $\mathbf{x}$  and for a random 2D subspace in the  $\mathbf{z}$  domain (the plot shows  $\pm 2\Sigma_\phi^{1/2}$  around  $\mu_\phi$ ). Observe the relatively small gap between the true posterior  $p_\theta(\mathbf{z}|\mathbf{x})$  and its variational approximation  $q_\phi(\mathbf{z}|\mathbf{x})$ . This figure shows some evidence of partial  $\mathbf{z}$ -convexity of  $J_1$  around the minimum of  $J_2$ , but it does not show how far is  $\mathbf{z}^1$  from  $\mathbf{z}^2$ .

The *stochastic decoder* takes as an input the latent variable  $\mathbf{z}$  and outputs the mean and covariance matrix of the Gaussian distribution  $p_\theta(\mathbf{x}|\mathbf{z})$ . Following (Dai and Wipf, 2019) we chose here an isotropic covariance  $\Sigma_\theta(\mathbf{z}) = \gamma I$  where  $\gamma > 0$  is trained, but independent of  $\mathbf{z}$ . This choice simplifies the minimization problem (9), because the term  $\det \Sigma_\theta(\mathbf{z})$  (being constant) has no effect on the  $\mathbf{z}$ -minimization.

The architecture of the decoder is also composed of 4 fully connected layers with ELU activations (to preserve continuous differentiability). The sizes of the layers are as follows:

$$12 \rightarrow 500 \rightarrow 500 \rightarrow 784.$$

Note that the covariance matrix is constant, so it does not augment the size of the output layer which is still  $784 = 24 \times 24$  pixels.

We train this architecture using TensorFlow<sup>1</sup> with batch size 64 and Adam algorithm for 400 epochs with learning rate 0.0001 (halving every 150 epochs) and rest of the parameters as default.

The subjective quality of the trained VAE is illustrated in Figure 1, including reconstruction examples (Figure 1(a)) and random samples (Figure 1(b)). Figure 1(c) shows that the encoder approximation  $q_\phi(\mathbf{z}|\mathbf{x})$  of the true posterior  $p_\theta(\mathbf{z}|\mathbf{x})$  is not perfect but is quite tight. It also shows that the true posterior  $p_\theta(\mathbf{z}|\mathbf{x})$  is pretty close to log-concave near the maximum of  $q_\phi(\mathbf{z}|\mathbf{x})$ .

### 3.2 EMPIRICAL VALIDATION OF ASSUMPTION 2

As we have shown in Section 2, our simplest Algorithm 1 is only guaranteed to converge when the encoder approximation is exact thus ensuring bi-convexity of  $J_1 = J_2$ .

In practice an exact autoencoder approximation is difficult to achieve, and in particular the VAE trained in the previous section has a small gap between  $J_1$  and  $J_2$ . To consider this case we proposed Algorithms 2 and 3 which are guaranteed to converge under weaker quasi-bi-convex conditions stated in Assumption 2.

In this section we experimentally check that the VAE we trained in the previous section actually verifies such conditions. We do so by selecting a random  $\mathbf{x}_0$  from MNIST test set and by computing  $\mathbf{z}^*(\mathbf{z}_0) := \text{grad descent}_{\mathbf{z}} J_1(\mathbf{x}_0, \mathbf{z})$  with different initial values  $\mathbf{z}_0$ . These experiments were performed using the Adam minimization algorithm with learning rate equal to 0.01.

Figures 2(a) and 2(c) show that  $\mathbf{z}^*(\mathbf{z}_0)$  reaches the global optimum for initializations  $\mathbf{z}_0$  chosen under far less restrictive conditions than those required by Assumption 2(A). Indeed from 200

<sup>1</sup>Code reused from <https://github.com/daib13/TwoStageVAE> (Dai and Wipf, 2019)

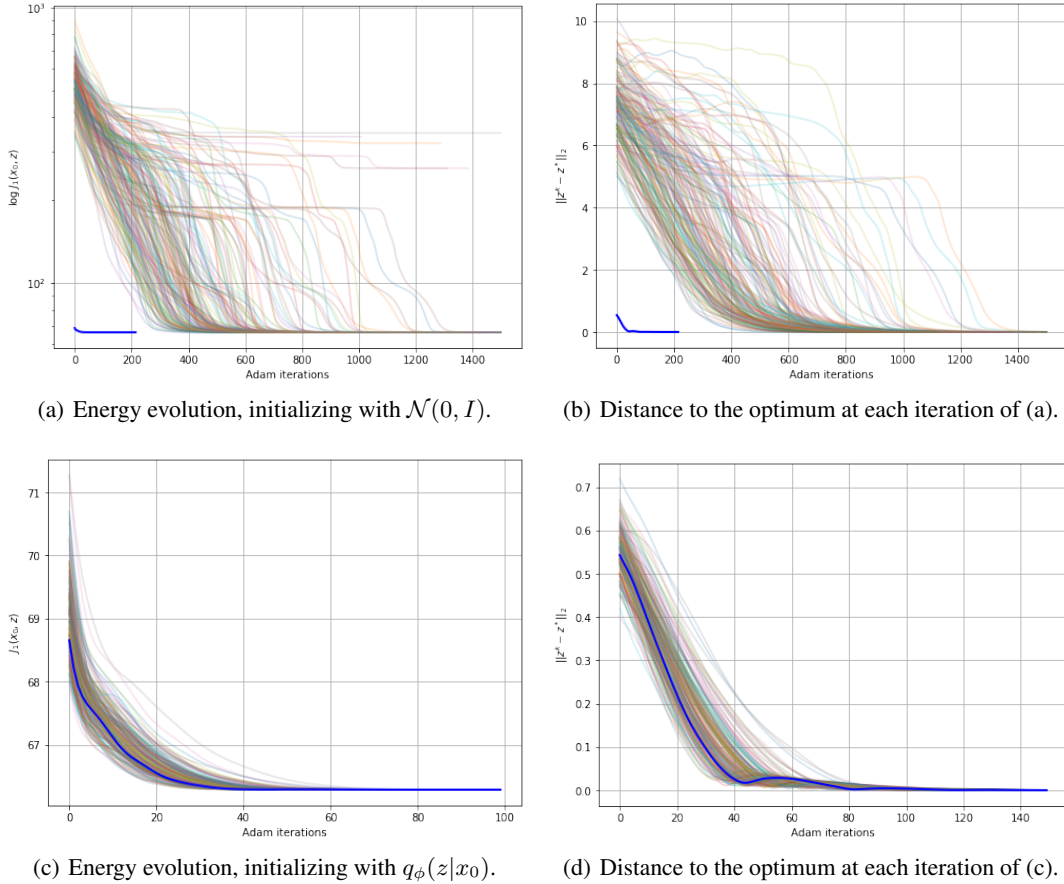


Figure 2: *Experimental validation of Assumption 2:* We take  $x_0$  from the *test* set of MNIST and minimize  $J_1(\mathbf{x}_0, \mathbf{z})$  with respect to  $\mathbf{z}$  using gradient descent from different initializations  $\mathbf{z}_0$ . The blue thick curve represents the trajectory if we initialize at the encoder approximation  $\mathbf{z}^1 = \arg \min_{\mathbf{z}} J_2(\mathbf{x}_0, \mathbf{z}) = \boldsymbol{\mu}_\phi(\mathbf{x}_0)$ . (a) and (c): Plots of the energy iterates  $J_1(\mathbf{x}_0, \mathbf{z}_k)$ . (b) and (d):  $\ell^2$  distances of each trajectory with respect to the global optimum  $\mathbf{z}^*$ . (a) and (b): Evolution of Adam on 200 random Gaussian initializations. (c) and (d): Initializing with random samples taken from the posterior approximation  $q_\phi(\mathbf{z}|\mathbf{x}_0)$  given by the encoder. *Conclusion:* Observe that all initializations  $\mathbf{z}^0$  such that  $J_1(\mathbf{x}_0, \mathbf{z}_0) \leq J_1(\mathbf{x}_0, \mathbf{z}^1)$  do converge to a unique global minimizer, and so do many other initializations.

random initializations  $z_0 \sim \mathcal{N}(0, I)$ , 195 reach the same global minimum, whereas 5 get stuck at a higher energy value. However these 5 initial values have energy values  $J_1(x_0, z_0) \gg J_1(x_0, z^1)$  far larger than those of the encoder initialization  $z^1 = \mu_\phi(x_0)$ , and are thus never chosen by Algorithms 2 and 3.

This experiment validates Assumption 2(A). In addition, it shows that we cannot assume  $z$ -convexity: The presence of plateaux in the trajectories of many random initializations as well as the fact that a few initializations do not lead to the global minimum indicates that  $J_1$  may not be everywhere convex with respect to  $z$ . However it still satisfies the weaker Assumption 2(A) which is sufficient to prove convergence in Theorem 2.

In Figures 2(b) and 2(d) we display the distances of each trajectory (except for the 5 outliers) to the global optimum  $z^*$  (taken as the median over all initializations  $z_0$  of the final iterates  $z^*(z_0)$ ); note that this optimum is always reached, which proves that  $z \mapsto J_1(x_0, z)$  has unique global minimizer.

### 3.3 IMAGE RESTORATION EXPERIMENTS

*Choice of  $x_0$ :* In the previous section, our validation experiments used a random  $x_0$  from the data set as initialization. When dealing with an image restoration problem, Algorithms 2 and 3 require an initial value of  $x_0$  to be chosen. In all experiments we choose this initial value as the simplest possible inversion algorithm, namely the regularized pseudo-inverse of the degradation matrix:

$$x_0 = A^\dagger y = (A^T A + \varepsilon Id)^{-1} A^T y.$$

*Choice of  $n_{\min}$ :* After a few runs of Algorithm 2 we find that in most cases, during the first two or three iterations  $z^1$  decreases the energy with respect to the previous iteration. But after at most five iterations the autoencoder approximation is no longer good enough and we need to perform gradient descent on  $z$  in order to further decrease the energy. Based on these findings we set  $n_{\min} = 5$  in Algorithm 3 for all experiments.

Figure 3 displays some selected results of compressed sensing and inpainting experiments on MNIST using the proposed approach. Figure 3(a) shows an *inpainting* experiment with 80% of missing pixels and Gaussian white noise with  $\sigma = 2/255$ . Figure 3(b) shows a *compressed sensing* experiment with  $m = 100$  random measurements and Gaussian white noise with  $\sigma = 2/255$ . For comparison we provide also the result of another decoupled approach proposed by Bora et al. (2017) with  $\lambda = 0.1$  as suggested in the paper, which uses the same generative model to compute the MAP estimator as in Equation (2), but does not make use of the encoder.<sup>2</sup>

As we can see in Figure 3, the proposed method significantly outperforms CSGM. There are still some failure cases (see last column in Figure 3(b)). However, in the vast majority of cases our alternate minimization scheme does not get stuck in local optima, as CSGM does.

## 4 CONCLUSIONS AND FUTURE WORK

In this work we presented a new framework to solve convex inverse problems with priors learned in the latent space via variational autoencoders. Unlike similar approaches like CSGM (Bora et al., 2017) which learns the prior based on generative models, our approach is based on a generalization of alternate convex search to quasi-biconvex functionals. This quasi-biconvexity is the result of considering the joint posterior distribution of latent and image spaces. As a result, the proposed approach provides stronger convergence guarantees. Experiments on inpainting and compressed sensing confirm this, since our approach gets stuck much less often in spurious local minima than CSGM, which is simply based on gradient descent of a highly non-convex functional. This leads to restored images which are significantly better in terms of MSE.

The present paper provides a first proof of concept of our framework, on a very simple dataset (MNIST) with a very simple VAE. More experiments are needed to:

- Verify that the framework preserves its qualitative advantages on more high-dimensional datasets (like CelebA, Fashion MNIST, etc.), and a larger selection of inverse problems.

<sup>2</sup>Since Bora et al. (2017) does not provide code, we implemented our own version of their algorithm and utilize the same trained VAE as encoder  $\phi$  for both algorithms.

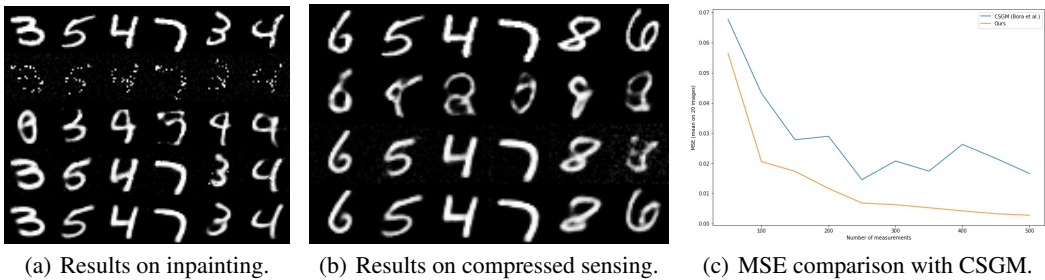


Figure 3: *Experimental results on MNIST. Comparison with CSGM algorithm (Bora et al., 2017).* (a) Some selected results from the inpainting experiment with  $m = 100$  (12.8%) known pixels. From top to bottom: original image  $x^*$ , corrupted image  $\tilde{x}$ , restored by (Bora et al., 2017), restored image  $\hat{x}$  by our framework (with Gaussian decoder) and the reconstruction of the original image by the VAE  $\mu_\theta(\mu_\phi(x^*))$  which can be seen as the best possible reconstruction if we use this model. (b) Same as (a) in a compressed sensing experiment with  $m = 100$  (12.8%) random measurements (without showing the second row). (c) MSE mean of the reconstruction of 20 images for the compressed sensing experiment, varying the number of measurements  $m$ . *Conclusion:* Our algorithm performs consistently better than CSGM. In addition our algorithm gets less often stuck in spurious local minima.

- Improve the quality of the prior model by using more elaborate variations of variational autoencoders which mix the VAE framework with normalizing flows (Dai and Wipf, 2019), adversarial training (Pu et al., 2017a,b; Zhang et al., 2019), or BiGANs (Donahue and Simonyan, 2019).

All variations of the VAE framework cited above have the potential to improve the quality of our generative model, and to reduce the gap between  $J_1$  and  $J_2$ . In particular, the adversarially symmetric VAE (Pu et al., 2017a,b) proves that when learning reaches convergence the autoencoder approximation is exact, meaning that Assumption 2 would become true.

When compared to other decoupled plug & play approaches that solve inverse problems using NN-based priors, our approach is constrained in different ways:

(a) In a certain sense our approach is *less constrained* than existing decoupled approaches since we do not require to retrain the NN-based denoiser to enforce any particular property to ensure convergence: Ryu et al. (2019) requires the denoiser’s residual operator to be non-expansive, and Gupta et al. (2018) and Shah and Hegde (2018) require the denoiser to act as a projector. The effect of these modifications to the denoiser on the quality of the underlying image prior has never been studied in detail and chances are that such constraints degrade it. Our method only requires a variational autoencoder without any further constraints, and the quality and expressiveness of this prior can be easily checked by sampling and reconstruction experiments. Checking the quality of the prior is a much more difficult task for Gupta et al. (2018), Ryu et al. (2019), and Shah and Hegde (2018) which rely on an implicit prior, and do not provide a generative model.

(b) On the other hand our method is *more constrained* in the sense that it relies on a generative model of a *fixed size*. Even if the generator and encoder are both convolutional neural networks, training and testing the same model on images of different sizes is a priori not possible because the latent space has a fixed dimension and a fixed distribution. As a future work we plan to explore different ways to address this limitation. The most straightforward way is to use our model to learn a prior of image patches of a fixed size and stitch this model via aggregation schemes like in EPLL (Zoran and Weiss, 2011) to obtain a global prior model for images of any size. Alternatively we can use hierarchical generative models like in (Karras et al., 2017) or resizable ones like in (Bergmann et al., 2017), and adapt our framework accordingly.

## 5 ACKNOWLEDGEMENTS

This work was partially funded by ECOS-Sud Project U17E04, by the French-Uruguayan Institute of Mathematics and interactions (IFUMI), by CSIC I+D (Uruguay) and by ANII (Uruguay) under Grant

---

FCE\_1\_2017\_1\_135458. The authors would like to thank Warith Harchaoui, Alasdair Newson and Said Ladjal for useful discussions.

---

## A CONVERGENCE PROOFS

Gorski et al. (2007) establishes a general result on the simple alternate convex search (ACS) algorithm which consist in iterating the wollowing two steps:

- $\mathbf{x}$ -minimization i.e.  $\mathbf{x}_{n+1} = \arg \min_{\mathbf{x}} J_1(\mathbf{x}, \mathbf{z}_n)$  (i.e. solve problem (8))
- $\mathbf{z}$ -minimization i.e.  $\mathbf{z}_{n+1} = \arg \min_{\mathbf{z}} J_1(\mathbf{x}_{n+1}, \mathbf{z})$  (i.e. solve problem (9))

until convergence for any continuous functional  $J_1$  which is either bi-convex or which allows to solve both partial minimization subproblems exactly.

Under certain conditions, Algorithms 1, 2, and 3 are actually implementations of the ACS algorithm, and the general result stated in (Gorski et al., 2007, Theorem 4.9) holds.

In the sequel we state the general theorem and then we verify the validity of the different hypotheses.

**Theorem 1** (Convergence of ACS). *Let  $X \subseteq \mathbb{R}^n$  and  $Z \subseteq \mathbb{R}^m$  be closed sets and  $J_1 : X \times Z \rightarrow \mathbb{R}$  be continuous. Let the optimization problems (8) and (9) be solvable.*

1. *If the sequence  $\{(\mathbf{x}_n, \mathbf{z}_n)\}_{n \in \mathbb{N}}$  generated by ACS is contained in a compact set then the sequence has at least one accumulation point.*
2. *In addition suppose that for each accumulation point  $(\mathbf{x}^*, \mathbf{z}^*)$ : either the optimal solution of (8) with  $\mathbf{z} = \mathbf{z}^*$  is unique or the optimal solution of (9) with  $\mathbf{x} = \mathbf{x}^*$  is unique. Then all accumulation points are partial optima and have the same function value.*
3. *Furthermore if*
  - (i)  $J_1$  *is differentiable and bi-convex, and*
  - (ii) *for each accumulation point  $(\mathbf{x}^*, \mathbf{z}^*)$  the optimal solutions of both (8) with  $\mathbf{z} = \mathbf{z}^*$  and (9) with  $\mathbf{x} = \mathbf{x}^*$  are unique;**then:*
  - (a) *the set of accumulation points is a connected compact set, and*
  - (b) *all accumulation points in the interior of  $X \times Z$  are stationary points*

*Proof.* This is the central result in (Gorski et al., 2007), proven in Theorem 4.9 and corollary 4.10.  $\square$

In the sequel we adopt the common assumption that all neural networks used in this work are composed of a finite number  $d$  of layers, each layer being composed of: (a) a linear operator (e.g. convolutional or fully connected layer), followed by (b) a non-linear  $L$ -Lipschitz component-wise activation function with  $0 < L < \infty$ .

Therefore we have the following property:

**Property 1.** *For any neural network  $f_\theta$  with parameters  $\theta$  having the structure described above: There exists a constant  $C_\theta$  such that  $\forall \mathbf{u}$ ,*

$$\|f_\theta(\mathbf{u})\|_2 \leq C_\theta \|\mathbf{u}\|_2.$$

Concerning activation functions we use two kinds:

- continuously differentiable activations like ELU, or
- continuous but non-differentiable activations like ReLU

**Lemma 1.**  $J_1$  *is continuous. In addition  $\mathbf{x} \mapsto J_1(\mathbf{x}, \mathbf{z})$  is convex for all  $\mathbf{z}$ , and the map  $\mathbf{x}_{\min} : \mathbf{z} \mapsto \arg \min_{\mathbf{x}} J_1(\mathbf{x}, \mathbf{z})$  is single-valued. In addition for continuously differentiable activation functions,  $J_1$  is continuously differentiable.*

*Proof.* By construction  $J_1$  is a composition of neural networks (which are composed of linear operators and continuous activations), linear and quadratic operations. Hence  $J_1$  is continuous with respect to both variables, and continuously differentiable if the activation functions are so. In addition

$J_1$  is quadratic in  $\mathbf{x}$  for any fixed  $\mathbf{z}$ . The closed form in equation (8) shows that the mapping  $\mathbf{x}_{\min}$  is single-valued. It is also continuous because  $\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta$  are continuous neural networks.  $\square$

**Lemma 2.**  $J_2$  is continuous. In addition  $\mathbf{z} \mapsto J_2(\mathbf{x}, \mathbf{z})$  is convex for all  $\mathbf{x}$ , and the map  $\mathbf{z}_{\min} : \mathbf{x} \mapsto \arg \min_{\mathbf{z}} J_2(\mathbf{x}, \mathbf{z})$  is single-valued.

*Proof.* By construction  $J_2$  is a composition of neural networks (which are composed of linear operators and continuous activations), linear and quadratic operations. Hence  $J_1$  is continuous with respect to both variables. In addition  $J_2$  is quadratic in  $\mathbf{z}$  for any fixed  $\mathbf{x}$ . The closed form in equation (14) shows that the mapping is single-valued. It is also continuous because  $\boldsymbol{\mu}_\phi, \boldsymbol{\Sigma}_\phi$  are CNNs composed of convolutions and ReLUs, which are continuous functions.  $\square$

**Lemma 3.**  $J_1(\mathbf{x}, \mathbf{z})$  is coercive.

*Proof.* If it was not coercive, then we could find a sequence  $(\mathbf{x}_k, \mathbf{z}_k) \rightarrow \infty$  such that  $J_1(\mathbf{x}_k, \mathbf{z}_k)$  is bounded. As a consequence all three (non-negative) terms are bounded. In particular  $\|\mathbf{z}_k\|$  is bounded, which means that  $\mathbf{x}_k \rightarrow \infty$ .

From Property 1,  $\boldsymbol{\mu}_\theta(\mathbf{z}_k)$  and  $\boldsymbol{\Sigma}_\theta(\mathbf{z}_k)$  are bounded for bounded  $\mathbf{z}_k$ .

Now since  $\boldsymbol{\mu}_\theta(\mathbf{z}_k)$  and  $\boldsymbol{\Sigma}_\theta(\mathbf{z}_k)$  are bounded and  $\mathbf{x}_k \rightarrow \infty$ , we conclude that  $H_\theta(\mathbf{x}_k, \mathbf{z}_k) \rightarrow \infty$  which contradicts our initial assumption.

As a consequence  $J_1$  is coercive.  $\square$

**Lemma 4 (Monotonicity).** *The sequence generated by Algorithms 2 and 3 is non-increasing. Under Assumption 1, Algorithm 1 is also non-increasing.*

*Proof.* Under assumption 1, Algorithm 1 is exactly the ACS algorithm which obviously ensures monotonicity.

Step 6 in Algorithm 2 obviously makes the energy decrease. So does step 5 because

$$J_1(\mathbf{x}_n, \mathbf{z}_{n+1}) \leq J_1(\mathbf{x}_n, \mathbf{z}^0) \leq J_1(\mathbf{x}_n, \mathbf{z}_n)$$

Step 10 in Algorithm 3 ensures that the  $\mathbf{x}$ -update decreases the energy. Step 5 in Algorithm 3 ensures that the  $\mathbf{z}$ -update decreases the energy.  $\square$

**Proof of proposition 1.**

1. *Compact domain and accumulation points*

Theorem 1 applies to closed subsets  $X$  and  $Z$  whereas our algorithm does not restrict the domain.

This is not a big issue because the monotonicity and coercivity allow to show that the sequence is actually bounded.

Indeed, by definition of the coercivity (Lemma 3), the level set  $S = \{(x, z) \mid J(x, z) \leq J_1(x_0, z_0)\}$  is bounded. The monotonicity of the sequence  $(J_1(x_k, z_k))$  (Lemma 4) implies that  $(x_k, z_k) \in S$  for any  $k$ . Moreover,  $S$  is closed, thus compact, since  $J_1$  is continuous. The existence of an accumulation point is straight-forward.

This proves the first part in Proposition 1.

2. *Properties of the set of accumulation points.*

Since the sequence  $\{(\mathbf{x}_n, \mathbf{z}_n)\}$  is bounded, there exist compact sets  $X_0$  and  $Z_0$  which contain the entire sequence. Now consider the dilated compact sets

$$X = \{\mathbf{x} : \exists \mathbf{e} \in \mathbb{R}^n, \|\mathbf{e}\| \leq \epsilon, \text{ and } (\mathbf{x} + \mathbf{e}) \in X_0\}$$

$$Z = \{z : \exists e \in \mathbb{R}^m, \|e\| \leq \epsilon, \text{ and } (z + e) \in Z_0\}.$$

Then the sequence  $\{(x_n, z_n)\}$  and all its accumulation points are all contained in the interior of  $X \times Z$ .

Now we can apply Theorem 1 to  $J_1$  on the restricted domain  $X \times Z$ . Indeed, from Assumption 1 we know that  $J_1 = J_2$ , which means that Algorithm 1 is an implementation of ACS. In addition Lemmas 1 and 2 show that optimization subproblems (8) and (9) are solvable and have unique solutions.

Therefore all hypotheses of Theorem 1 are met, which shows the second part of Proposition 1.  $\square$

### ***Proof of proposition 2.***

#### *1. Compact domain and accumulation points*

First note that Algorithms 2 and 3 (for  $n > n_{\min}$ ) are particular implementations of ACS on  $J_1$ . Therefore the monotonicity of the sequence  $J_1(x_n, z_n)$  (Lemma 4) and the coercivity of  $J_1$  are sufficient to show the first part of proposition 2 exactly like in proposition 1. This also shows that the sequence  $\{(x_n, z_n)\}$  is bounded.

#### *2. Accumulation points are partial optima*

Since the sequence  $\{(x_n, z_n)\}$  is bounded, there exist compact sets  $X_0$  and  $Z_0$  which contain the entire sequence. Now consider the dilated compact sets

$$X = \{x : \exists e \in \mathbb{R}^n, \|e\| \leq \epsilon, \text{ and } (x + e) \in X_0\}$$

$$Z = \{z : \exists e \in \mathbb{R}^m, \|e\| \leq \epsilon, \text{ and } (z + e) \in Z_0\}.$$

Then the sequence  $\{(x_n, z_n)\}$  and all its accumulation points are all contained in the interior of  $X \times Z$ .

Now we can apply parts 2 and 3 of Theorem 1 to  $J_1$  on the restricted domain  $X \times Z$ .

Indeed, from Lemma 1, the  $x$ -minimization subproblem (8) is solvable with unique solution. In addition from Assumption 2(A), the  $z$ -minimization subproblem (9) is solvable and algorithms 2 and 3 provide that solution for each fixed  $x$ .

Therefore all hypotheses for part 2 of Theorem 1 are met, which shows that all accumulation points are partial optima, and they all have the same function value.

#### *3. Accumulation points are stationary points.*

Known facts:

1.  $J_1$  differentiable:  $\nabla J_1(x_k, z_k) = \left( \frac{\partial J_1}{\partial x}(x_k, z_k), \frac{\partial J_1}{\partial z}(x_k, z_k) \right)$
2.  $x \mapsto J_1(x, z)$  convex:

$$\forall y, J_1(y, z) \geq J_1(x, z) + \left\langle \frac{\partial J_1}{\partial x}(x, z), y - x \right\rangle$$

Hence a minimizer is a stationary point.

3. Unicity of partial minimizers: in particular, thanks to the convexity of  $J_1$  – stationary points are minimizers

$$\frac{\partial J_1}{\partial x}(x^*, \hat{z}) = 0 \quad \implies \quad \frac{\partial J_1}{\partial x}(x, \hat{z}) \neq 0 \quad \text{if } x \neq x^*$$

4.  $J_1$  coercive and descent scheme:  $(x_k, z_k)_k$  and  $(x_{k+1}, z_k)_k$  are bounded thus have convergent subsequences



5. Decrease property:

$$J_1(x_k, z_k) \geq J_1(x_{k+1}, z_k) \geq J_1(x_{k+1}, z_{k+1})$$

If  $J_1$  is lowerbounded,

$$\lim_{k \rightarrow +\infty} J_1(x_k, z_k) = \lim_{k \rightarrow +\infty} J_1(x_{k+1}, z_k) = \lim_{k \rightarrow +\infty} J_1(x_{k+1}, z_{k+1})$$

6. Fermat's rule:

$$\frac{\partial J_1}{\partial x}(x_{k+1}, z_k) = 0 \quad \text{and} \quad \frac{\partial J_1}{\partial z}(x_{k+1}, z_{k+1}) = 0$$

- Let  $(x^*, z^*)$  be an accumulation point of  $(x_k, z_k)_k$ . If  $(x_{k_j+1}, z_{k_j+1})_j$  converges to  $(x^*, z^*)$ ,  $(x_{k_j})_j$  and  $(z_{k_j})_j$  are bounded thus have convergent subsequences. So there exists a sequence  $\{j_n\}_n$  such that (double extraction of subsequence)

$$\begin{aligned} \lim_{n \rightarrow +\infty} x_{k_{j_n}+1} &= x^* & \text{and} & & \lim_{n \rightarrow +\infty} x_{k_{j_n}} &= \hat{x} \\ \lim_{n \rightarrow +\infty} z_{k_{j_n}+1} &= z^* & \text{and} & & \lim_{n \rightarrow +\infty} z_{k_{j_n}} &= \hat{z} \end{aligned}$$

According to Fact 5,  $J_1(x^*, z^*) = J_1(x^*, \hat{z}) = J_1(\hat{x}, \hat{z})$ .

- Partial minimizers:

$$\forall x, J_1(x, z_{k_{j_n}}) \geq J_1(x_{k_{j_n}+1}, z_{k_{j_n}})$$

By taking the limit (since  $J_1$  is continuous)

$$\forall x, J_1(x, \hat{z}) \geq J_1(x^*, \hat{z})$$

Thus,  $x^*$  is the unique minimizer of  $J_1(\cdot, \hat{z})$ . By unicity of the partial minimizer,  $\hat{x} = x^*$ .

$$\forall z, J_1(x_{k_{j_n}+1}, z) \geq J_1(x_{k_{j_n}+1}, z_{k_{j_n}+1})$$

By taking the limit (since  $J_1$  is continuous)

$$\forall z, J_1(x^*, z) \geq J_1(x^*, z^*)$$

Thus,  $z^*$  is the unique minimizer of  $J_1(x^*, \cdot)$ . By unicity of the partial minimizer (Assumption 2(B)),  $\hat{z} = z^*$ .

- Since  $J_1(\cdot, z^*)$  is convex,  $x^*$  is the minimizer of  $J_1(\cdot, z^*)$  iff  $\frac{\partial J_1}{\partial x}(x^*, z^*) = 0$
- If  $z \mapsto J_1(x, z)$  is continuously differentiable, then

$$\lim_{n \rightarrow +\infty} \frac{\partial J_1}{\partial z}(x_{k_{j_n}+1}, z_{k_{j_n}+1}) = \frac{\partial J_1}{\partial z}(x^*, z^*) = 0$$

- As a consequence,

$$\nabla J_1(x^*, z^*) = 0$$

Otherwise said,

If  $J_1(\cdot, z)$  is convex,  $J_1$  is continuously differentiable and coercive, and if the partial minimizers are all unique, then any accumulation point of the sequence  $(x_k, z_k)_k$  is a stationary point.

□

---

## REFERENCES

- Aguerreberre, Cecilia, Andres Almansa, Julie Delon, Yann Gousseau, and Pablo Muse (2017). “A Bayesian Hyperprior Approach for Joint Image Denoising and Interpolation, With an Application to HDR Imaging”. In: *IEEE Transactions on Computational Imaging* 3.4, pp. 633–646. ISSN: 2333-9403. DOI: 10.1109/TCI.2017.2704439. arXiv: 1706.03261. URL: [https://nounsse.github.io/HBE\\_project/](https://nounsse.github.io/HBE_project/) (cit. on p. 6).
- Bergmann, Urs, Nikolay Jetchev, and Roland Vollgraf (2017). “Learning Texture Manifolds with the Periodic Spatial GAN”. In: *(ICML) International Conference on Machine Learning* 1, pp. 722–730. arXiv: 1705.06566 (cit. on p. 11).
- Bigdeli, Siavash Arjomand and Matthias Zwicker (2017). *Image Restoration using Autoencoding Priors*. Tech. rep. arXiv: 1703.09964 (cit. on p. 3).
- Bigdeli, Siavash Arjomand, Meiguang Jin, Paolo Favaro, and Matthias Zwicker (2017). “Deep Mean-Shift Priors for Image Restoration”. In: *(NIPS) Advances in Neural Information Processing Systems* 30, pp. 763–772. arXiv: 1709.03749. URL: <http://papers.nips.cc/paper/6678-deep-mean-shift-priors-for-image-restoration> (cit. on p. 3).
- Bora, Ashish, Ajil Jalal, Eric Price, and Alexandros G Dimakis (2017). “Compressed sensing using generative models”. In: *(ICML) International Conference on Machine Learning*. Vol. 2. JMLR.org, pp. 537–546. ISBN: 9781510855144. arXiv: arXiv:1703.03208v1 (cit. on pp. 3, 4, 10, 11).
- Chambolle, A (2004). “An algorithm for total variation minimization and applications”. In: *Journal of Mathematical Imaging and Vision* 20, pp. 89–97. DOI: 10.1023/B:JMIV.0000011325.36760.1e (cit. on p. 3).
- Chan, S. H., X. Wang, and O. A. Elgendy (2017). “Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications”. In: *IEEE Transactions on Computational Imaging* 3.1, pp. 84–98. ISSN: 2333-9403. DOI: 10.1109/TCI.2016.2629286. arXiv: 1605.01710 (cit. on p. 3).
- Chen, Yunjin and Thomas Pock (2017). “Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6, pp. 1256–1272. ISSN: 01628828. DOI: 10.1109/TPAMI.2016.2596743. arXiv: 1508.02848 (cit. on p. 3).
- Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter (2016). “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)”. In: *(ICLR) International Conference on Learning Representations*. arXiv: 1511.07289 (cit. on p. 6).
- Dai, Bin and David Wipf (2019). “Diagnosing and Enhancing VAE Models”. In: *ICLR*, pp. 1–42. arXiv: 1903.05789. URL: <https://openreview.net/forum?id=B1e0X3C9tQ> (cit. on pp. 8, 11).
- Diamond, Steven, Vincent Sitzmann, Felix Heide, and Gordon Wetzstein (2017). “Unrolled optimization with deep priors”. In: arXiv: 1705.08041 (cit. on p. 3).
- Donahue, Jeff and Karen Simonyan (2019). “Large Scale Adversarial Representation Learning”. In: arXiv: 1907.02544. URL: <http://arxiv.org/abs/1907.02544> (cit. on p. 11).
- Dong, Chao, Chen Change Loy, Kaiming He, and Xiaoou Tang (2014). “Learning a deep convolutional network for image super-resolution”. In: *European conference on computer vision*. Springer, pp. 184–199 (cit. on p. 3).
- Gao, Hongyun, Xin Tao, Xiaoyong Shen, and Jiaya Jia (2019). “Dynamic scene deblurring with parameter selective sharing and nested skip connections”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3848–3856 (cit. on p. 3).
- Gharbi, Michaël, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand (2016). “Deep joint demosaicking and denoising”. In: *ACM Transactions on Graphics (TOG)* 35.6, p. 191 (cit. on p. 3).
- Gilton, Davis, Greg Ongie, and Rebecca Willett (2019). “Neumann networks for inverse problems in imaging”. In: arXiv: 1901.03707 (cit. on p. 3).
- Gorski, Jochen, Frank Pfeuffer, and Kathrin Klamroth (2007). “Biconvex sets and optimization with biconvex functions: a survey and extensions”. In: *Mathematical Methods of Operations Research* 66.3, pp. 373–407. ISSN: 1432-2994. DOI: 10.1007/s00186-007-0161-1 (cit. on pp. 4, 6, 7, 13).
- Gregor, Karol and Yann LeCun (2010). “Learning fast approximations of sparse coding”. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, pp. 399–406 (cit. on p. 3).

- 
- Gupta, Harshit, Kyong Hwan Jin, Ha Q Nguyen, Michael T McCann, and Michael Unser (2018). “CNN-based projected gradient descent for consistent CT image reconstruction”. In: *IEEE transactions on medical imaging* 37.6, pp. 1440–1453. DOI: 10.1109/TMI.2018.2832656. arXiv: 1709.01809 (cit. on pp. 3, 11).
- Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen (2017). “Progressive Growing of GANs for Improved Quality, Stability, and Variation”. In: *(ICLR) International Conference on Learning Representations* 10.2, pp. 327–331. arXiv: 1710.10196. URL: <https://openreview.net/forum?id=Hk99zCeAb> (cit. on p. 11).
- Kingma, Diederik P and Max Welling (2013). “Auto-Encoding Variational Bayes”. In: *(ICLR) International Conference on Learning Representations*. ML, pp. 1–14. ISBN: 1312.6114v10. DOI: 10.1051/0004-6361/201527329. arXiv: 1312.6114 (cit. on pp. 4, 5, 7).
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. (2012). “Imagenet classification with deep convolutional neural networks”. In: *(NIPS) Advances in neural information processing systems*, pp. 1097–1105. ISSN: 10495258. arXiv: 1102.0183 (cit. on p. 3).
- Lecun, Yann, Leon Bottou, Yoshua Bengio, and P. Haffner (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. ISSN: 00189219. DOI: 10.1109/5.726791. arXiv: 1102.0183 (cit. on p. 7).
- Louchet, Cécile and Lionel Moisan (2013). “Posterior expectation of the total variation model: Properties and experiments”. In: *SIAM Journal on Imaging Sciences* 6.4, pp. 2640–2684. ISSN: 19364954. DOI: 10.1137/120902276 (cit. on p. 3).
- Meinhardt, Tim, Michael Moller, Caner Hazirbas, and Daniel Cremers (2017). “Learning proximal operators: Using denoising networks for regularizing inverse imaging problems”. In: *(ICCV) International Conference on Computer Vision*, pp. 1781–1790. DOI: 10.1109/ICCV.2017.198. URL: [http://openaccess.thecvf.com/content\\_iccv\\_2017/html/Meinhardt\\_Learning\\_Proximal\\_Operators\\_ICCV\\_2017\\_paper.html](http://openaccess.thecvf.com/content_iccv_2017/html/Meinhardt_Learning_Proximal_Operators_ICCV_2017_paper.html) (cit. on p. 3).
- Pereyra, Marcelo (2016). “Proximal Markov chain Monte Carlo algorithms”. In: *Statistics and Computing* 26.4, pp. 745–760. ISSN: 0960-3174. DOI: 10.1007/s11222-015-9567-4. arXiv: 1306.0187. URL: <http://dx.doi.org/10.1007/s11222-015-9567-4> (cit. on p. 3).
- Pu, Yunchen, Weiyao Wang, Ricardo Henao, Liqun Chen, Zhe Gan, Chunyuan Li, and Lawrence Carin (2017a). “Adversarial symmetric variational autoencoder”. In: *(NIPS) Advances in Neural Information Processing Systems*, pp. 4331–4340 (cit. on p. 11).
- (2017b). “Adversarial symmetric variational autoencoder”. In: *(NIPS) Advances in Neural Information Processing Systems*. Vol. 2017-Decem, pp. 4331–4340 (cit. on p. 11).
- Reehorst, Edward T and Philip Schniter (2018). “Regularization by denoising: Clarifications and new interpretations”. In: *IEEE Transactions on Computational Imaging* 5.1, pp. 52–67. DOI: 10.1109/TCI.2018.2880326. arXiv: 1806.02296 (cit. on p. 3).
- Rudin, Leonid I., Stanley Osher, and Emad Fatemi (1992). “Nonlinear total variation based noise removal algorithms”. In: *Physica D: Nonlinear Phenomena* 60.1-4, pp. 259–268. ISSN: 01672789. DOI: 10.1016/0167-2789(92)90242-F (cit. on p. 3).
- Ryu, Ernest K., Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin (2019). “Plug-and-Play Methods Provably Converge with Properly Trained Denoisers”. In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 5546–5557. arXiv: 1905.05406. URL: <http://proceedings.mlr.press/v97/ryu19a.html> (cit. on pp. 3, 11).
- Schwartz, Eli, Raja Giryes, and Alex M Bronstein (2018). “DeepISP: Toward learning an end-to-end image processing pipeline”. In: *IEEE Transactions on Image Processing* 28.2, pp. 912–923 (cit. on p. 3).
- Shah, Viraj and Chinmay Hegde (2018). “Solving linear inverse problems using gan priors: An algorithm with provable guarantees”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4609–4613 (cit. on pp. 3, 11).
- Teodoro, Afonso M., José M. Bioucas-Dias, and Mário A. T. Figueiredo (2018). *Scene-Adapted Plug-and-Play Algorithm with Guaranteed Convergence: Applications to Data Fusion in Imaging*. arXiv: 1801.00605 (cit. on p. 3).
- Zhang, Kai, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang (2017a). “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising”. In: *IEEE Transactions on Image Processing* 26.7, pp. 3142–3155 (cit. on p. 3).

- 
- Zhang, Kai, Wangmeng Zuo, Shuhang Gu, and Lei Zhang (2017b). “Learning Deep CNN Denoiser Prior for Image Restoration”. In: *(CVPR) IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2808–2817. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.300. arXiv: 1704.03264. URL: [http://openaccess.thecvf.com/content\\_cvpr\\_2017/html/Zhang\\_Learning\\_Deep\\_CNN\\_CVPR\\_2017\\_paper.html](http://openaccess.thecvf.com/content_cvpr_2017/html/Zhang_Learning_Deep_CNN_CVPR_2017_paper.html) (cit. on p. 3).
- Zhang, Kai, Wangmeng Zuo, and Lei Zhang (2018). “FFDNet: Toward a fast and flexible solution for CNN-based image denoising”. In: *IEEE Transactions on Image Processing* 27.9, pp. 4608–4622 (cit. on p. 3).
- Zhang, Zijun, Ruixiang Zhang, Zongpeng Li, Yoshua Bengio, and Liam Paull (2019). “Perceptual Generative Autoencoders”. In: *(ICLR) International Conference on Learning Representations*, pp. 1–7. arXiv: 1906.10335. URL: <https://github.com/zj10/PGA> (cit. on p. 11).
- Zoran, Daniel and Yair Weiss (2011). “From learning models of natural image patches to whole image restoration”. In: *2011 International Conference on Computer Vision*. IEEE, pp. 479–486. ISBN: 978-1-4577-1102-2. DOI: 10.1109/ICCV.2011.6126278. URL: <http://people.csail.mit.edu/danielzoran/EPLLICCVCameraReady.pdf> (cit. on pp. 3, 11).