

A Generative Variational Model for Inverse Problems in Imaging*

Andreas Habring[†] and Martin Holler[†]

Abstract. This paper is concerned with the development, analysis, and numerical realization of a novel variational model for the regularization of inverse problems in imaging. The proposed model is inspired by the architecture of generative convolutional neural networks; it aims to generate the unknown from variables in a latent space via multilayer convolutions and nonlinear penalties, and penalizes an associated cost. In contrast to conventional neural-network-based approaches, however, the convolution kernels are learned directly from the measured data such that no training is required. The present work provides a mathematical analysis of the proposed model in a function space setting, including proofs for regularity and existence/stability of solutions, and convergence for vanishing noise. Moreover, in a discretized setting, a numerical algorithm for solving various types of inverse problems with the proposed model is derived. Numerical results are provided for applications in inpainting, denoising, deblurring under noise, superresolution, and JPEG decompression with multiple test images.

Key words. inverse problems, data science, mathematical imaging, deep image prior, generative neural networks, machine learning

AMS subject classifications. 49J27, 65J20, 68T07, 94A08

DOI. 10.1137/21M1414978

1. Introduction. Inverse problems require the inversion of a forward model $A : X \rightarrow Y$ that describes the relation of an unknown $u \in X$ to some given (possibly noisy) data $y \in Y$, i.e., they require the solution of $A(u) \approx y$ for $u \in X$. Characteristic about inverse problems is that this inversion is ill-posed, meaning that a classical inverse of A is not well-defined and/or, even more severely, that a direct inversion of A is unstable in the sense that small deviations (such as noise) in the given data can lead to large deviations in a corresponding solution.

Variational regularization is a well-established method for solving inverse problems. It defines an approximate solution of an inverse problem as a solution of a minimization problem of the form

$$(1.1) \quad \min_u \mathcal{D}_y(A(u)) + \lambda \mathcal{R}(u),$$

where $\mathcal{D}_y : Y \rightarrow [0, \infty]$ is a discrepancy term measuring the distance of $A(u)$ to the given data, $\mathcal{R} : X \rightarrow [0, \infty]$ is a regularization functional, and $\lambda > 0$ is a (regularization) parameter. Advantages of such an approach are as follows:

*Received by the editors April 26, 2021; accepted for publication (in revised form) November 3, 2021; published electronically March 3, 2022.

<https://doi.org/10.1137/21M1414978>

Funding: The authors received funding from the Austrian Research Promotion Agency (FFG) (project 881561). The second author is a member of NAWI Graz (<https://www.nawigraz.at>) and BioTechMed Graz (<https://biotechmedgraz.at>).

[†]Institute of Mathematics and Scientific Computing, University of Graz, Graz, Austria (andreas.habring@uni-graz.at, martin.holler@uni-graz.at, <https://imsc.uni-graz.at/holler.m>).

- (i) With a suitable choice of \mathcal{R} , the mapping $y \mapsto [u \text{ solving (1.1)}]$ can be shown to be continuous (in an appropriate sense), meaning that variational methods allow one to obtain a continuous mapping from given data to a corresponding solution. This is most evident when analyzing inverse problems in infinite dimensional spaces, as only in this case is there a clear distinction between continuous and noncontinuous solution methods even in a linear setting.
- (ii) The functional \mathcal{R} allows one to incorporate any prior knowledge that one might have on the unknown u . This is in particular important in imaging, where the unknown corresponds to image data that is typically highly structured. Exploiting such structure in the unknown via suitable choices of \mathcal{R} has proven to be highly beneficial for the quality of the obtained reconstruction and can be seen as one of the main reasons for the success of variational methods for inverse problems.

In the past decades, a main focus of research on variational methods for inverse problems in imaging and beyond was on sparsity-based modeling of the unknown, either via sparsity in some transform domain [30] or sparsity of (higher-order) derivatives [34, 9]. In this context, significant progress has been made in terms of both practical performance and analytic understanding of variational methods, with examples for the latter being, for instance, recovery guarantees, a description of the structure of solutions, or convergence results for vanishing noise.

More recently, (machine-)learning-based approaches have become increasingly popular also in inverse problems, mainly for their practical performance, for instance, in image processing [3]. The latter is particularly striking for more complex image data containing repeating patterns or texture, a class of image data, where classical variational models still show limitations and are outperformed also by approaches such as dictionary- or patch-based methods [25]. So far, however, the level of analytic understanding of machine learning methods, in particular in view of an analysis for inverse problems in function space, is far from the one of variational methods. Furthermore, the requirement of large sets of training data is a limiting factor for the application of machine learning methods to inverse problems, where the availability of training data is often very limited.

The present work is a step toward overcoming these shortcomings; it is concerned with the development, analysis, and numerical realization of a novel variational model for inverse problems in imaging. The proposed approach uses techniques from generative models in machine learning to overcome a limited performance of existing variational methods in particular for highly structured image data such as texture; see Figure 1 for an inpainting example. In contrast to existing works on conventional machine learning approaches, however, our approach does not rely on training data, but enforces regularity purely by the architecture of the underlying network itself. Further, our model is amenable to analysis in function space, such that analytic guarantees on well-posedness, stability, and regularity of solutions for general inverse problems can be ensured independently of discretizations.

Analytically, our work can be seen as a multilayer extension of [13], where our numerical experiments confirm a significantly improved practical performance resulting from the introduction of a multilayer structure. Conceptually, our approach is also closely related to the

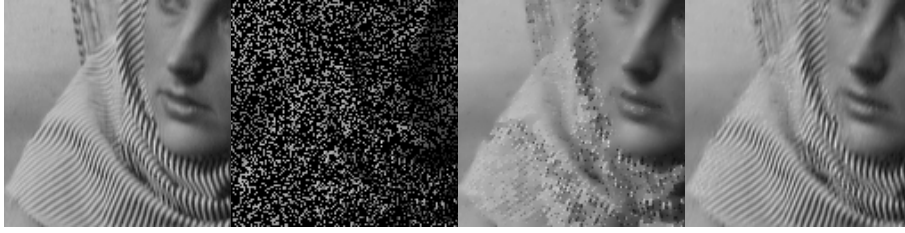


Figure 1. Inpainting from 30% known pixels. Left to right: ground truth, corrupted, reconstruction with total generalized variation regularization [10], reconstruction with the proposed method.

recent works on *deep image priors* (DIPs) [26], and we refer to subsection 1.2 for a detailed discussion of relevant works in that context.

1.1. The proposed approach. We consider a class of methods where a low dimensional variable in a so-called latent space Z is mapped to an image $u \in X$ via a neural network. Trained appropriately, these networks can be regarded as parametrizations of certain classes of images. Due to their ability to generate images of a given class/distribution we refer to these networks as *generative*. In view of inverse problems, a direct application of a pretrained generative neural network, denoted by $g_\theta : Z \rightarrow X$, where θ summarizes all pretrained parameters, would be to solve

$$\min_z \mathcal{D}_y(A(g_\theta(z)))$$

for reconstruction, where again A defines the forward model and \mathcal{D}_y is a data discrepancy term.

In order to avoid training, works such as Lempitsky, Vedaldi, and Ulyanov [26] instead learn also the parameters θ directly from the measured data by solving

$$\min_{z, \theta} \mathcal{D}_y(A(g_\theta(z))).$$

(Note that the original work [26] leaves z fixed in practice.) Surprisingly, this yields rather impressive results for different inverse problems in imaging, suggesting that the architecture of the generator comprises a good prior for image data.

A problem with this approach, however, is that a generator of sufficient size may be able to produce almost any given image, even random noise [38], meaning that the above model has no regularizing effect at all. To account for that, [26] relies on early stopping of the numerical solution algorithm (as well as different heuristics), arguing that the generator usually fits natural images quite fast, but needs more time to fit noise.

Noting the relation of such approaches to convolutional sparse coding methods [13, 32], our approach realizes a similar idea, but within the framework of variational methods in function space, which are amenable to mathematical analysis. As regularization, we define a generative prior $\mathcal{G} : X \rightarrow [0, \infty]$ that generates the unknown $u \in X$ from multilayer convolutions of a

variable in latent space in a way that is optimal with respect to an associated cost. In detail, we define \mathcal{G} as

(1.2)

$$\mathcal{G}(u) = \inf_{\substack{\mu = (\mu^l)_{l=1}^L, \\ \theta = (\theta^l)_{l=1}^L}} \|\mu\|_{\mathcal{M}} + \sum_{k,l} \mathcal{J}_{l-1} \left(\mu_k^{l-1} - \sum_{n=1}^{N_l} \mu_n^l * \theta_{n,k}^l \right) \quad \text{subject to} \quad \begin{cases} u = \sum_{n=1}^{N_1} \mu_n^1 * \theta_{n,1}^1, \\ \|\theta_{n,k}^l\|_2 \leq 1 \quad \text{for all } n, k, l; \\ Z\theta = 0 \end{cases}$$

see Figure 2 for a visualization. Here, the latent variables μ^l are modeled as measures, the norm $\|\cdot\|_{\mathcal{M}}$ denotes the Radon norm (the measure-space counterpart of the L^1 -norm), and the θ^l denote filter kernels (modeled as L^2 functions), which are normalized by the constraint $\|\theta_{n,k}^l\|_2 \leq 1$. The functionals \mathcal{J}_l can, for instance, be indicator functions of $\{0\}$, i.e., $\mathcal{J}_l(x) = 0$ if $x = 0$ and $\mathcal{J}_l(x) = \infty$ otherwise, to ensure that u is indeed generated from a sequence of multi-layer convolutions of the latent variable μ^L in the deepest layer. However, we allow for more general \mathcal{J}_l , including also smooth relaxations, since our numerical implementation relies on differentiability of the \mathcal{J}_l . Nonlinearities in the multilayer convolution are replaced by non-linear penalties on the latent variables of the intermediate layers. The last constraint $Z\theta = 0$, with Z any linear, bounded operator, allows us to add some additional constraints on the filter kernels, e.g., to have zero mean. For a detailed introduction of the generative prior \mathcal{G} , we refer to section 2.2.1.

In this work \mathcal{G} as in (1.2) is analyzed and applied as regularization for general inverse problems. Our motivation for introducing \mathcal{G} is to capture image structure, such as repeating patterns, that is not captured well by classical variational methods such as Tikhonov or total variation (TV) regularization. Moreover, the multilayer approach inspired by the architecture of convolutional neural networks is shown to be advantageous in practice compared to a single layer convolutional model (see section 3).

On the other hand, many existing variational methods are already very well-suited for simpler image structures such as piecewise smooth regions. Furthermore, our analysis in section 2 shows that generative priors such as \mathcal{G} will always produce continuous image structures as output. Accounting for these facts, we allow for the infimal convolution of \mathcal{G} with a second

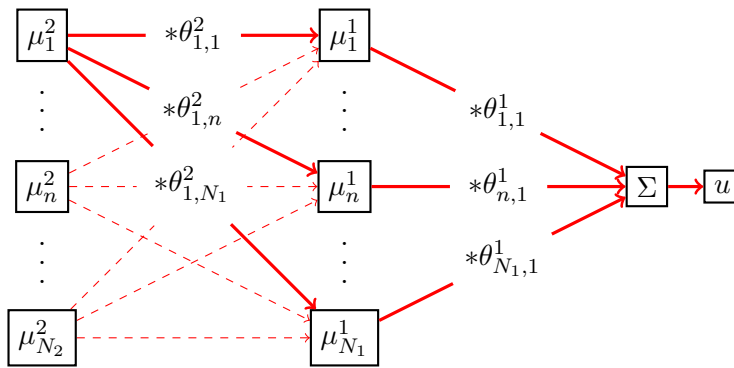


Figure 2. Sketch of the generative network with $L = 2$ layers.



Figure 3. Inpainting from 30% known pixels with the proposed method. Left to right: ground truth image, corrupted image, reconstruction u , generative part v , remaining part $u - v$.

regularization functional (such as the total variation functional [34]) in our overall approach to the regularization of inverse problems. This results in a variational energy of the form

$$(1.3) \quad \min_{u,v} \lambda \mathcal{D}_y(A(u)) + s_{\mathcal{R}}(\nu) \mathcal{R}(u - v) + s_{\mathcal{G}}(\nu) \mathcal{G}(v)$$

with $\lambda \in (0, \infty)$ a regularization parameter and $s_{\mathcal{R}}(\nu) = \frac{\nu}{\min(\nu, 1-\nu)}$ and $s_{\mathcal{G}}(\nu) = \frac{1-\nu}{\min(\nu, 1-\nu)}$ for $\nu \in (0, 1)$ being parametrizations of a ν -dependent balancing between \mathcal{R} and \mathcal{G} . In (1.3) \mathcal{R} can be the indicator function of $\{0\}$, i.e., $\mathcal{R}(u) = 0$ if $u = 0$ and ∞ otherwise, resulting in a model that only uses \mathcal{G} for regularization. Or, alternatively, \mathcal{R} can be chosen to be any other suitable regularization functional, such as the TV functional [34], and the resulting infimal convolution of the two functionals yields a decomposition $u = v + (u - v)$ of the solution u of (1.3) as shown, for example, in Figure 3 for an inpainting experiment.

Contributions. By developing an energy-based regularization approach for general inverse problems in function space, our work provides the following contributions.

- *Well-posedness:* Analytically, we prove properties of \mathcal{G} such as coercivity and lower semicontinuity in function space that yield well-posedness (including stability, but not uniqueness) for the application of our model to any linear inverse problem, without any assumptions, and to nonlinear inverse problems under weak standard assumptions [22]. Specifically, we provide the following results:
 - The infimum over (μ, θ) in the definition of \mathcal{G} as in (1.2) is attained, and \mathcal{G} is coercive and weakly lower semicontinuous; see Lemmas 2.6 to 2.8.
 - Problem (1.3) admits a solution; see Theorem 2.12.
 - Solutions of (1.3) are weakly, subsequentially stable up to shifts of u in a subspace where both A and \mathcal{R} are invariant (Theorem 2.15), and for vanishing noise and appropriately chosen regularization parameters, weak, subsequential convergence of the solutions to solutions of an appropriate limit problem can be guaranteed (Theorem SM1.2).
- *Regularity:* We provide a regularity result for the generative prior in function space. That is, we show that our convolutional network inspired architecture generates only continuous functions as outputs whenever at least two layers are used, i.e., $\mathcal{G}(u) < \infty$ only if u is continuous; see Proposition 2.2. This is an important observation as it has several crucial implications: (i) Our prior is well-suited for removing noise-like artifacts, as it is not capable of fitting highly discontinuous noise. (ii) It is very reasonable to combine our generative prior, as well as other related generative

neural-network-based priors, with a second term that allows for discontinuous structures. In our work, this is achieved via the infimal convolution with a TV-like functional \mathcal{R} . (iii) It is the penalization of latent variables via norms, as done in this work, that yields well-posedness and regularity of solutions. For other works that do not penalize latent variables, such as the DIP [26], similar results cannot be expected. This confirms the need for early stopping, as done in those works, in order to avoid the approximation of noise.

- *Generality:* Conceptually, the proposed approach is applicable to general inverse problems and we provide numerical results showing state-of-the-art performance for inpainting, denoising, deblurring under noise, superresolution, and JPEG decompression.
- *Model complexity:* While our approach still requires the solution of a nonconvex optimization problem, for which global optimality cannot be guaranteed, we still believe that our energy-based formulation and the fact that we do not require any heuristics is a big advantage in terms of simplicity and interpretability of our model compared to existing deep learning methods. In [26], for instance, the authors propose to use skip connections in the network, a noise-based regularization that is perturbing the network input with a random noise at each iteration of the fitting process, and also a method of reverting to previous iterates during the minimization. All of these features improve practical performance but are difficult to interpret or understand. We do not use any such heuristics in the implementation and all the theoretical results proven in the paper directly apply to the experiments.

In particular, we highlight that we are able to achieve results that are at least comparable to the ones of the DIP, while our model reduces the number of effective degrees of freedom, e.g., by a factor of 100 (from $\approx 3 \times 10^6$ to $\approx 3 \times 10^4$) for inpainting with the *Barbara* image as shown in Figure 1 (or by a factor of 10, e.g., to 2×10^5 , if one counts also auxiliary variables introduced in order to smooth hard constraints in the numerical realization). In this context we note that also the work [19] reduces model complexity for the DIP, and we refer to subsection 1.2 for a comparison to [19].

- *Foundations for further analysis:* By developing a generative prior in function space that delivers state-of-the-art numerical results, we enable a further analysis and mathematical development of such models with respect to, for instance, convex relaxations as done in [13] for a single-layer version, or a further understanding of structural properties of solutions using results from [6, 7].

1.2. Related works. The DIP [26] has been a great incentive for research in the direction the present work is headed. While most works in that context are focused on experimental results, some also take a more mathematical point of view.

In [13], a single layer version of the proposed model, using only one convolutional layer, is introduced and analyzed as well as tested on images, showing promising results for images composed of texture and piecewise smooth parts. In [20], the authors investigate how convolutional neural networks act as successful priors for image reconstruction solely via their architecture, as empirically shown by [26]. The authors prove that fitting a convolutional neural network with fixed convolution kernels via early stopped gradient descent to a signal

denoises it. In [19], the authors propose the deep decoder, an underparametrized, untrained generative network for inverse problems in imaging. To be precise, the authors use a very simple, nonconvolutional network (consisting only of a pixelwise linear combination of different channels, upsampling, nonlinearities, and normalization layers) as an image generator. The network architecture is chosen such that the number of parameters is less than the image dimensionality. It is shown theoretically and empirically that the underparametrization acts as regularization by removing noise, without compromising the reconstruction error too much. These results are in line with our observations in the sense that we could also reduce the number of parameters and model complexity significantly compared to the original DIP. In [15], a different point of view is introduced, showing that, under some conditions, a DIP can be interpreted as learning an optimal Tikhonov functional instead of training a neural network. In [23], algorithmic guarantees for solving inverse problems using a DIP are provided.

In [5, 12, 31, 38, 4, 39], neural networks are combined with additional regularization terms for solving inverse problems. In [5], the authors consider a problem of the form

$$(1.4) \quad \min_{\theta} \mathcal{D}_y(Ag_{\theta}(z)) + \lambda \mathcal{R}(g_{\theta}(z))$$

with a regularizing functional \mathcal{R} . Existence, stability, and convergence of solutions are obtained by imposing suitable assumptions. Moreover, also different learned approaches to obtain a better initialization for the proposed method are used. A main difference to our method is that we do not penalize $g_{\theta}(z)$, but θ and z instead. Also in [12] a problem of the form (1.4) with $\mathcal{D}_y(z) = \frac{1}{2}\|z - y\|^2$ and \mathcal{R} a weighted TV is considered and solved using the alternating direction method of multipliers framework.

The problem formulation in [31] is of the form

$$(1.5) \quad \min_z \|A\phi_{\alpha}(z) - y\|^2 + \alpha \|z\|_{1,w},$$

where $(\phi_{\alpha})_{\alpha \geq 0}$ is a family of functions modeling (pretrained) neural networks and $\|\cdot\|_{1,w}$ denotes a weighted 1-norm. The authors provide proofs of existence, stability, and convergence of solutions of the method in an abstract setting employing a list of assumptions on the involved functions and spaces.

In [38], a DIP is combined with TV regularization and a regularization of the network parameters leading to

$$(1.6) \quad \min_{\theta} \|Ag_{\theta}(z) - y\|^2 + \lambda \text{TV}(g_{\theta}(z)) + \mu \mathcal{R}(\theta),$$

where \mathcal{R} is a learned regularization functional. Contrary to the present work, here the TV is applied to the output of the neural network, which in our model shall in particular contain oscillatory features of the image. In [4, 39], the authors use a pretrained invertible generative network, $g_{\theta} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ with θ learned, as a prior. In [4], the authors consider

$$\min_z \|Ag_{\theta}(z) - y\|_2^2 + \lambda \|z\|_2^2$$

and in [39] this ansatz is further generalized. Both works use a likelihood-based approach and discuss recovery guarantees.

A field that is also related to our work due to its generative modeling is sparse coding. Sparse coding aims to find a representation of a vector $y \in \mathbb{R}^m$ in terms of a (typically pre-trained) dictionary $D \in \mathbb{R}^{m \times n}$ and a sparse coefficient vector $x \in \mathbb{R}^n$, such that $y = Dx$. In convolutional sparse coding, the dictionary matrix D is further restricted to represent a convolution operator.

In [37], a multilayer convolutional sparse coding method is discussed, i.e., instead of the single dictionary D , a composition of convolutional dictionaries $D_N D_{N-1} \dots D_1$ is used. The works [40, 32] present methods for cartoon texture decomposition via single-layer convolutional sparse coding combined with a TV penalty, resembling our combination of the generative prior \mathcal{G} with a second regularization. In [40], the dictionary is pretrained, but in [32], this results in a minimization problem similar to a single-layer version of the proposed method.

All the mentioned approaches using neural networks were generative in the sense that the network is used to generate the unknown. To the contrary, in [29, 28] the authors use pretrained neural networks directly as regularization functionals. More precisely, a neural network $\psi_\theta : X \rightarrow \mathbb{R}$ with parameters θ is trained to map an image to a scalar penalty. Then, once the parameters are fixed, ψ_θ is applied for solving inverse problems via

$$\min_u \mathcal{D}_y(Au) + \lambda \psi_\theta(u).$$

In both works the authors also present theoretical analyses of the models relying on assumptions on the trained network. Main differences to our work are that we use the neural network as an image generator and that our network is untrained. Moreover, contrary to [29, 28], we do not make any additional assumptions on our network for the analysis.

1.3. Outline of the paper. In section 2, we introduce and analyze the proposed method in an infinite dimensional setting proving, in particular, results on existence, stability, and regularity of solutions. In section 3, we provide numerical results obtained with our method and a comparison to the state of the art for different imaging applications.

2. The generative model in function spaces. In order to properly introduce the proposed model and prove theoretical results, we first need some preliminaries.

2.1. Notation and preliminaries. In the following, $\Omega, \Sigma \subset \mathbb{R}^d$ are bounded domains, i.e., bounded, open, and simply connected sets in \mathbb{R}^d , with $0 \in \Sigma$. Moreover, we denote $\Omega_\Sigma = \Omega - \Sigma = \{x - y \mid x \in \Omega, y \in \Sigma\}$. For $q \in [1, \infty]$, we denote the Hölder conjugate exponent as q' , i.e., $q' \in [1, \infty]$ is, such that $\frac{1}{q} + \frac{1}{q'} = 1$, using the convention that $\frac{1}{\infty} = 0$. For $\omega \subset \mathbb{R}^d$, $f \in L^q(\omega)$ and $g \in L^{q'}(\omega)$, we use the notation $\langle f, g \rangle := \int_\omega f(x)g(x) dx$ for the standard dual pairing and $\|f\|_q^q = \int_\omega |f(x)|^q dx$ for the L^q -norm. We define the zero extension of any function $f : \omega \rightarrow \mathbb{R}$ as

$$\tilde{f} : \mathbb{R}^d \rightarrow \mathbb{R}, \quad \tilde{f}(x) = \begin{cases} f(x) & \text{if } x \in \omega, \\ 0 & \text{else.} \end{cases}$$

Further, we define $C_0(\omega)$ as the closure of $C_c(\omega)$ with respect to the uniform norm and the space $(\mathcal{M}(\omega), \|\cdot\|_{\mathcal{M}})$ as the dual of $C_0(\omega)$. Note that $\mathcal{M}(\omega)$ coincides with the space of finite Radon measures and $\|\cdot\|_{\mathcal{M}}$ with the TV norm, whenever ω is a locally compact, separable

metric space [2, Theorem 1.54, Remark 1.57], a property that holds for all domains considered in this work. For ω bounded, we will sometimes identify a function $f \in L^q(\omega)$ with an element of $\mathcal{M}(\omega)$ via $C_0(\omega) \ni \phi \mapsto \langle f, \phi \rangle$. For normed spaces X, Y , we denote the space of all linear, bounded operators from X to Y as $\mathcal{L}(X, Y)$. Last, we denote the indicator function on a set M as \mathcal{I}_M , i.e., $\mathcal{I}_M(x) = 0$ if $x \in M$ and ∞ otherwise. We also need a notion of convolution.

Definition and Proposition 2.1. *Let $q \in (1, 2]$, $g \in L^2(\Sigma)$ and $\mu \in \mathcal{M}(\Omega_\Sigma)$. Then there exists a unique function denoted by $\mu * g \in L^q(\Omega)$ such that for all $\phi \in C_c(\Omega)$,*

$$(2.1) \quad \langle \mu * g, \phi \rangle = \int_{\Omega_\Sigma} \int_{\Sigma} \tilde{\phi}(x+y) g(x) dx d\mu(y).$$

*In particular, the right-hand side is well-defined for all such ϕ . Moreover, there exists $C > 0$ such that $\|\mu * g\|_q \leq C \|\mu\|_{\mathcal{M}} \|g\|_2$.*

Proof. This is shown, for instance, in [13, section 3.1]. ■

Note that in case g and μ are sufficiently regular, by Fubini's theorem, $\mu * g$ can be identified with the classical convolution given as

$$(\mu * g)(x) = \int_{\Omega_\Sigma} g(x-y) d\mu(y).$$

2.2. Modeling and analysis. With the proposed method, we aim to solve inverse problems in imaging by (partially) generating the unknown image from an architecture inspired by generative convolutional neural networks. We also allow for an optional combination of this approach via infimal convolution with a second regularization term that models parts of the unknown image that are not well-described by the generative approach. To be precise, let $q \in (1, 2]$, Y a Banach space, $y \in Y$ the given data, and $A : L^q(\Omega) \rightarrow Y$ a forward operator; then we aim to find $u \in L^q(\Omega)$ with $A(u) \approx y$ by solving

$$(P(y)) \quad \min_{u, v \in L^q(\Omega)} \mathcal{E}_y(u, v) \text{ with } \mathcal{E}_y(u, v) := \lambda \mathcal{D}_y(A(u)) + s_{\mathcal{R}}(\nu) \mathcal{R}(u-v) + s_{\mathcal{G}}(\nu) \mathcal{G}(v),$$

where $\lambda \in (0, \infty)$, $\nu \in (0, 1)$, $s_{\mathcal{R}}(\nu) = \frac{\nu}{\min(\nu, 1-\nu)}$, $s_{\mathcal{G}}(\nu) = \frac{1-\nu}{\min(\nu, 1-\nu)}$, and

- \mathcal{D}_y is a data fidelity term penalizing the discrepancy between $A(u)$ and the data y ,
- \mathcal{G} is a *generative prior* inspired by the architecture of generative convolutional neural networks, introduced in detail in subsection 2.2.1 below, and
- \mathcal{R} is any other regularization functional satisfying standard conditions as stated in Assumption 2.9 below. We think of \mathcal{R} as a functional that enforces piecewise smoothness such as $\mathcal{R} = \text{TV}$, but its choice is rather flexible and also $\mathcal{R} = \mathcal{I}_{\{0\}}$ is feasible, reducing our model to using only \mathcal{G} for regularization.

In $(P(y))$, u is the image, v as the *generative part* of u , which shall in particular contain oscillatory features of the image, and $u-v$ contains all information not captured by the generative part u . The parameters ν and λ balance the different regularization terms and the data fidelity, respectively, where the parametrizations $s_{\mathcal{R}}(\nu)$ and $s_{\mathcal{G}}(\nu)$ are introduced to achieve a ν -dependent balancing between \mathcal{R} and \mathcal{G} , that does not interfere with the balancing between data fidelity and regularization.

2.2.1. The generative prior \mathcal{G} . We model the generative part $v \in L^q(\Omega)$ in $(P(y))$ to be the output of a neural-network-inspired generative architecture as follows: Let L be the number of hidden layers, let N_l be the number of latent variables in layer $l = 1, \dots, L$, and set $N_0 = 1$. Then we define, for $l = 1, \dots, L$, via

$$E^l \subset \{(n, k) \mid n \in \{1, \dots, N_l\}, k \in \{1, \dots, N_{l-1}\}\}$$

a set of all connections between latent variables of layer l and of layer $l-1$; see Figure 2 for a visualization of a corresponding generative network. In order to avoid latent variables that do not influence the output, we assume that each latent variable of layer l is connected to at least one latent variable of layer $l-1$, i.e., for each $n \in \{1, \dots, N_l\}$ there exists $k \in \{1, \dots, N_{l-1}\}$ such that $(n, k) \in E^l$.

Given the domains $\Sigma, \Omega \subset \mathbb{R}^d$, we recursively define domains Ω_l via $\Omega^1 = \Omega - \Sigma$ and $\Omega^{l+1} = \Omega^l - \Sigma$ for $l = 1, \dots, L-1$. Further, we denote

- $M^l := \mathcal{M}(\Omega^l)^{N_l}$,
- $M := M^1 \times M^2 \times \dots \times M^L$

to be the spaces of latent variables $\mu = (\mu^1, \dots, \mu^L) \in M$ with $\mu^l = (\mu_{n,k}^l)_{(n,k) \in E^l} \in M^l$, and

- $\Theta^l := L^2(\Sigma)^{|E^l|}$,
- $\Theta := \Theta^1 \times \Theta^2 \times \dots \times \Theta^L$

to be the spaces of filter kernels $\theta = (\theta^1, \dots, \theta^L) \in \Theta$ with $\theta^l = (\theta_{n,k}^l)_{(n,k) \in E^l} \in \Theta^l$.

Our generative prior will constrain the input v to be written as a sum of multilayer convolutions of the latent variable $\mu^L \in M^L$. To this aim, we define vectorial convolution operators $*_l$ via

$$\begin{aligned} *_1 : M^1 \times \Theta^1 &\rightarrow L^q(\Omega), & *_l : M^l \times \Theta^l &\rightarrow M^{l-1}, \\ (\mu, \theta) &\mapsto \sum_{n=1}^{N_1} \mu_n * \theta_{n,1}, & (\mu, \theta) &\mapsto \left(\sum_{(n,k) \in E^l} \mu_n * \theta_{n,k} \right)_{k=1}^{N_{l-1}}, \end{aligned}$$

for $l = 2, \dots, L$, where we note that, by Definition and Proposition 2.1, the convolutions $\mu^l *_l \theta^l$ are actually contained in more regular L^q -spaces. The functional $\mathcal{G} : L^q(\Omega) \rightarrow [0, \infty]$ is then defined as

(GEN)

$$\mathcal{G}(v) = \inf_{\substack{\mu \in M, \\ \theta \in \Theta}} \|\mu\|_{\mathcal{M}} + \sum_{l=2}^L \mathcal{J}_{l-1} \left(\mu^{l-1} - \mu^l *_l \theta^l \right) \quad \text{s.t.} \quad \begin{cases} v = \mu^1 *_1 \theta^1, \\ \|\theta_{n,k}^l\|_2 \leq 1 & \text{for all } n, k, l, \\ Z\theta = 0, \end{cases}$$

where $\|\mu\|_{\mathcal{M}} = \sum_{l=1}^L \sum_{n=1}^{N_l} \|\mu_n^l\|_{\mathcal{M}}$ denotes a componentwise Radon norm, $Z \in \mathcal{L}(\Theta, \mathbb{R}^{N_c})$ with $N_c \in \mathbb{N}$, and $\mathcal{G}(v) = \infty$ in case the constraints in (GEN) cannot be met by any μ, θ . Note that while our interest is to enforce the constraint $\mu^{l-1} = \mu^l *_l \theta^l$ exactly, we allow also a relaxed version of this constraint via incorporating functionals $\mathcal{J}_l : M^l \rightarrow [0, \infty]$, that are weak* lower semicontinuous and such that $\mathcal{J}_l(0) = 0$. This is a generalization, as the choice $\mathcal{J}_l = \mathcal{I}_{\{0\}}$ is feasible and allows us to enforce hard constraints.

We can make the following observations regarding the generative prior \mathcal{G} : The latent variables μ^l are penalized with the L^1 -type norm $\|\cdot\|_{\mathcal{M}}$. In our approach, this is the replacement for nonlinearities in the network and aims to enforce sparsity of the μ^l . A relation to nonlinearities in neural networks is given indirectly via the soft-thresholding operator resulting from $\|\cdot\|_{\mathcal{M}}$, which acts on the μ^l within our numerical solution algorithm. In order to ensure well-posedness, and in particular that (GEN) admits a minimum, we also need to penalize the filter kernels. Indeed, note that due to the bilinearity of the convolution, it is possible to shift a nonzero scalar α between μ and θ without changing the outcome, that is, $\mu^l *_l \theta^l = (\frac{1}{\alpha}\mu^l) *_l (\alpha\theta^l)$. This would, however, decrease the objective functional value for $\alpha > 1$. Therefore, without a penalty on θ , one could always let $\alpha \rightarrow \infty$ above and hence (GEN) would not admit a minimum. We decided to use a norm constraint on the filter kernels instead of an additive term since again by the bilinearity of the convolution this is not limiting the model, but avoids ambiguities. We allow for additional, linear constraints $Z\theta = 0$ on the filter kernels. In our applications later on, where \mathcal{G} is combined with a TV-type functional, this will be used to enforce zero mean of the filter kernels of the last layer, i.e., $\int_{\Sigma} \theta_{n,1}^1 = 0$ for all $n = 1, \dots, N_1$. In the context of TV-type functionals this is natural. The TV does not penalize constant translations, therefore we want them to be contained in $u - v$ and thus v shall have zero mean. In the general model, however, any type of (weakly continuous) constraints, in particular also no constraints ($Z = 0$), are possible.

In order to gain more understanding of the regularizing properties of the proposed network architecture, we will in the following discuss regularity of the output v . To simplify notation, we denote for $\theta \in \Theta$ and $\mu \in M$,

$$(2.2) \quad G(\mu, \theta) = \|\mu\|_{\mathcal{M}} + \sum_{l=2}^L \mathcal{J}_{l-1} \left(\mu^{l-1} - \mu^l *_l \theta^l \right),$$

and, for $v \in L^q(\Omega)$, the feasible set as

$$F(v) := \{(\mu, \theta) \in M \times \Theta \mid (\mu, \theta) \text{ satisfies the constraints in (GEN)}\},$$

such that $\mathcal{G}(v) = \inf_{(\mu, \theta) \in F(v)} G(\mu, \theta)$.

Proposition 2.2. *Let $\mathcal{G} : L^q(\Omega) \rightarrow [0, \infty]$ be given as in (GEN) with $L \geq 2$ and assume additionally that \mathcal{J}_l is such that $\mathcal{J}_l(\mu) < \infty$ implies $\mu \in L^2(\Omega^l)^{N_l}$. Then, if $\mathcal{G}(v) < \infty$, it holds true that $v \in C(\overline{\Omega})$.*

Proof. Let $\mathcal{G}(v) < \infty$. This implies the existence of $(\mu, \theta) \in F(v)$ such that $G(\mu, \theta) < \infty$. From Definition and Proposition 2.1, it follows that $\mu^L *_L \theta^L \in L^2(\Omega^{L-1})^{N_{L-1}}$. Together with the assumption on \mathcal{J}_l , this implies $\mu^{L-1} \in L^2(\Omega^{L-1})^{N_{L-1}}$. Repeating this argument, in case $L > 2$, yields $\mu^1 \in L^2(\Omega^1)^{N_1}$. Finally, $v = \mu^1 *_1 \theta^1 \in C(\overline{\Omega})$ as it is a sum of convolutions of two L^2 functions; see, for instance, [11, Theorem 3.14]. ■

Note that this result requires a network depth of at least $L = 2$; the single-layer variant with $L = 1$ can also produce discontinuous outputs in general. Also note that the assumption on \mathcal{J}_l in Proposition 2.2 is in particular true for $\mathcal{J}_l = \mathcal{I}_{\{0\}}$ as well as for the relaxed version of $\mathcal{I}_{\{0\}}$, which is introduced in Example 2.11 and used in the applications in section 3.

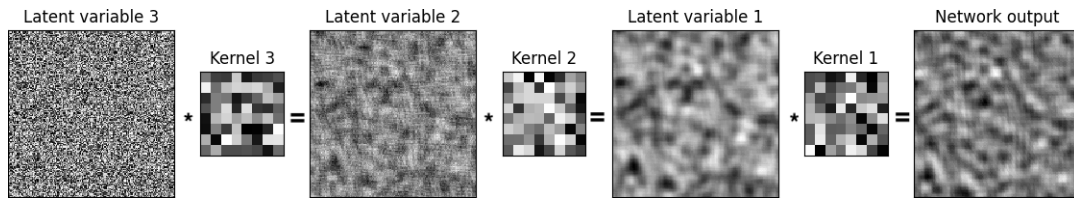


Figure 4. Smoothing effect of the convolution with increasing network depth. After randomly initializing all kernels and the latent variables in the deepest layer (latent variable 3), this figure shows, from left to right, the resulting variables of intermediate layers and the network output. As can be observed visually, the smoothness of the variables increases with an increased number of subsequent convolutions.

Remark 2.3 (insights from function space). In our opinion, an important reason for the success of using deeper generative networks in imaging (as opposed to single-layer models) lies in the fact that the convolution increases regularity, as proved in Proposition 2.2, and as can also be practically observed in Figure 4, where randomly initialized latent variables and filter kernels are convolved. Proposition 2.2 in particular shows that the output of our convolutional network is continuous whenever the network has at least two layers. This has several important implications. First, it explains the efficacy of the proposed generative prior \mathcal{G} for removing noise-like artifacts, which are typically highly discontinuous. Second, Proposition 2.2 also suggests to combine \mathcal{G} with a second regularizing functional \mathcal{R} , which allows for jump discontinuities (e.g., $\mathcal{R} = TV$), since otherwise our model would not be able to reconstruct these.

If, on the contrary, also the filter kernels θ are not more regular than belonging to $\mathcal{M}(\Sigma)$, it is easy to see that the convolution does, in general, not increase regularity as we can only expect $\mu^{L-1} \in \mathcal{M}(\Omega^{L-1})^{N_{L-1}}$. This has an interesting consequence for generative-neural-network based models: In case no regularization of the latent variables μ and the kernels θ is used, as done in the original DIP [26] and most of the subsequent works, the learned latent variables and kernels can be arbitrarily irregular and the resulting network might generate noise. This explains why, indeed, it makes sense to use early stopping as proposed in [26].

We now begin our analysis of the functional \mathcal{G} . Our main goal is to prove that \mathcal{G} is coercive and lower semicontinuous, which makes it an appropriate regularizing functional. We start with a general continuity result for the convolution.

Lemma 2.4 (sequential weak*-continuity of the convolution). Take $q \in (1, 2]$ arbitrary and let $(g_m)_m, g$ in $L^2(\Sigma)$ and $(\mu_m)_m$, and μ in $\mathcal{M}(\Omega_\Sigma)$ be such that $g_m \rightharpoonup g$ and $\mu_m \xrightarrow{*} \mu$ as $m \rightarrow \infty$. Then, it follows that

$$\mu_m * g_m \rightharpoonup \mu * g \text{ in } L^q(\Omega) \text{ as } m \rightarrow \infty.$$

Proof. The proof can be achieved in three steps. First, use the Arzelà–Ascoli theorem to show that, for $\phi \in C_c(\Omega)$ fixed, a subsequence of $(\langle \mu_m * g_m, \phi \rangle)_m$ converges to $\langle \mu * g, \phi \rangle$. Second, extend this to the entire sequence, and third, via density, conclude weak* convergence as claimed. For the reader’s convenience, this proof is elaborated in detail in section SM1 of the supplementary material for this paper. ■

Based on this, we can now establish lower semicontinuity of \mathcal{G} in an appropriate topology. In this context, all notions of convergence in the product spaces M and Θ , such as weak* convergence in M or weak convergence in Θ , refer to componentwise convergence in the underlying \mathcal{M} and L^2 , respectively.

Lemma 2.5. *Let $\mathcal{G} : L^q(\Omega) \rightarrow [0, \infty]$ be given as in (GEN), where $q \in (1, 2]$, $Z \in \mathcal{L}(\Theta, \mathbb{R}^{N_c})$ and the $\mathcal{J}_l : M^l \rightarrow [0, \infty]$ are weak* lower semicontinuous with $\mathcal{J}_l(0) = 0$.*

Then, for sequences $(v_m)_m$ in $L^q(\Omega)$, $(\mu_m)_m$ in M , and $(\theta_m)_m$ in Θ such that

$$v_m \rightharpoonup v \text{ in } L^q(\Omega), \quad \mu_m \xrightarrow{*} \mu \text{ in } M, \text{ and } \theta_m \rightharpoonup \theta \text{ in } \Theta,$$

it holds that

- (i) $G(\mu, \theta) \leq \liminf_{m \rightarrow \infty} G(\mu_m, \theta_m)$ and
- (ii) if $(\mu_m, \theta_m) \in F(v_m)$ for all m , then also $(\mu, \theta) \in F(v)$.

Proof. We start by proving (i). By Lemma 2.4, for all $l = 2, \dots, L$,

$$(\mu_m)^l *_l (\theta_m)^l \rightharpoonup \mu^l *_l \theta^l \text{ in } L^2(\Omega^{l-1})^{N_{l-1}} \text{ as } m \rightarrow \infty.$$

This implies also convergence in the weak* sense in M^{l-1} . Further, \mathcal{J}_l is weak* lower semicontinuous in M^l by assumption and $\|\cdot\|_{\mathcal{M}}$ is weak* lower semicontinuous as being a dual norm. Hence, G is lower semicontinuous as claimed.

To prove (ii), we have to show that (μ, θ) satisfies the three constraints in (GEN). The first constraint $v = \mu^1 *_1 \theta^1$ holds true by uniqueness of weak limits and by weak convergence of $(\mu_m)^1 *_1 (\theta_m)^1$ to $\mu^1 *_1 \theta^1$ as shown in Lemma 2.4. The constraints $\|\Theta_{n,k}^l\|_2 \leq 1$ and $Z\theta = 0$ follow from weak convergence of $(\theta_m)_m$ using weak lower semicontinuity of $\|\cdot\|_2$ and weak-to-weak continuity of Z , respectively. ■

Lemma 2.6. *With the assumptions of Lemma 2.5, for any $v \in L^q(\Omega)$ with $\mathcal{G}(v) < \infty$, the infimum in $\mathcal{G}(v)$ is attained, i.e., there exists $(\mu, \theta) \in F(v)$, such that $\mathcal{G}(v) = G(\mu, \theta)$.*

Proof. Since $\mathcal{G}(v) < \infty$, we pick a minimizing sequence $(\mu_m, \theta_m)_m$ in $F(v)$ such that the sequence $(G(\mu_m, \theta_m))_m$ is bounded. From this, the definitions of G and $F(v)$ imply that the sequences $(\mu_m)_m \subset M$ and $(\theta_m)_m \subset \Theta$ are bounded. By the theorem of Banach and Alaoglu applied to the space of Radon measures and reflexivity of L^2 , we can hence find (nonrelabeled) subsequences such that $\mu_m \xrightarrow{*} \mu$ in M and $\theta_m \rightharpoonup \theta$ in Θ as $m \rightarrow \infty$. Lemma 2.5 yields that also the limit (μ, θ) is feasible, i.e., $(\mu, \theta) \in F(v)$, and that

$$G(\mu, \theta) \leq \liminf_{m \rightarrow \infty} G(\mu_m, \theta_m) = \mathcal{G}(v),$$

which concludes the proof. ■

Lemma 2.7. *With the assumptions of Lemma 2.5, the generative prior $\mathcal{G} : L^q(\Omega) \rightarrow [0, \infty]$ is weakly lower semicontinuous.*

Proof. Let $v_m \rightharpoonup v$ in $L^q(\Omega)$. We want to show that $\mathcal{G}(v) \leq \liminf_{m \rightarrow \infty} \mathcal{G}(v_m)$ for which we can assume that $\liminf_{m \rightarrow \infty} \mathcal{G}(v_m) < \infty$, as otherwise there is nothing to prove. By moving to a (nonrelabeled) subsequence, we can further assume that $\liminf_{m \rightarrow \infty} \mathcal{G}(v_m) = \lim_{m \rightarrow \infty} \mathcal{G}(v_m)$ and that $\mathcal{G}(v_m) < \infty$ for all m . By Lemma 2.6, for every m , there exists $(\mu_m, \theta_m) \in F(v_m)$,

such that $\mathcal{G}(v_m) = G(\mu_m, \theta_m)$. The fact that $(\mathcal{G}(v_m))_m$ is bounded and the definitions of G and $F(v_m)$ imply that the sequences $(\mu_m)_m$ and $(\theta_m)_m$ are bounded in M and Θ , respectively. As in the proof of [Lemma 2.6](#), we can extract a further subsequence, again not relabeled, and find μ, θ such that $\mu_m \xrightarrow{*} \mu$ in M and $\theta_m \rightarrow \theta$ in Θ as $m \rightarrow \infty$. [Lemma 2.5](#) then implies that $(\mu, \theta) \in F(v)$ and

$$\mathcal{G}(v) \leq G(\mu, \theta) \stackrel{\text{Lemma 2.5}}{\leq} \liminf_{m \rightarrow \infty} G(\mu_m, \theta_m) = \liminf_{m \rightarrow \infty} \mathcal{G}(v_m) = \lim_{m \rightarrow \infty} \mathcal{G}(v_m). \quad \blacksquare$$

Lemma 2.8. *With the assumptions of [Lemma 2.5](#), $\mathcal{G} : L^q(\Omega) \rightarrow [0, \infty]$ is proper and coercive.*

Proof. In order to show that \mathcal{G} is proper, we simply note that $\mathcal{G}(0) = 0$. To prove coercivity, let $(v_m)_m \subset L^q(\Omega)$ such that $\|v_m\|_q \rightarrow \infty$. We have to show that $\mathcal{G}(v_m) \rightarrow \infty$ as well. Assuming the contrary, there is a subsequence $(v_{m_k})_k$ of $(v_m)_m$ such that $(\mathcal{G}(v_{m_k}))_k$ is bounded. By [Lemma 2.6](#), we can pick $(\mu_{m_k}, \theta_{m_k}) \in F(v_{m_k})$ such that $\mathcal{G}(v_{m_k}) = G(\mu_{m_k}, \theta_{m_k})$. This implies again that $(\mu_{m_k})_k$ and $(\theta_{m_k})_k$ are bounded, which, from [Definition and Proposition 2.1](#), in turn implies that also v_{m_k} is bounded in $L^q(\Omega)$, contradicting our assumption. \blacksquare

2.2.2. Analysis for inverse problems. The goal of this section is to prove that $(P(y))$ admits a stable minimum, which converges for vanishing noise. This will be done for \mathcal{G} as defined in [section 2.2.1](#) and for rather general functionals \mathcal{R} and \mathcal{D}_y . For the sake of simplicity, we will provide the results in this section for linear forward operators $A \in \mathcal{L}(L^q(\Omega), Y)$, as this is possible without any additional assumptions on the forward model. Building on the properties of \mathcal{G} as we have established in [section 2.2.1](#), a generalization of these results to nonlinear inverse problems is possible under standard assumptions; see, for instance, [\[22, 36\]](#). Also for the case of linear A , our proofs on existence, stability, and convergence for vanishing noise can essentially be derived from the obtained properties of \mathcal{G} with classical techniques (see, for instance, [\[22, 36\]](#)). However, as the infimal-convolution-based approach in our model as well as our technique of factoring out the kernel of the forward operator A in order to avoid additional assumptions require special care, we provide short self-contained proofs in the following.

The assumptions used in this setting are summarized as follows, with concrete examples of Z , \mathcal{J}_l , \mathcal{R} , and \mathcal{D}_y fulfilling these assumptions provided in [Example 2.11](#) below.

Assumption 2.9 (assumptions for existence/stability). Let $q \in (1, 2]$ and further assume the following:

1. The data fidelity functional $\mathcal{D}_y : Y \rightarrow [0, \infty]$ is proper, coercive, lower semicontinuous, and convex, and $A \in \mathcal{L}(L^q(\Omega), Y)$ is the forward operator.
2. The generative prior $\mathcal{G} : L^q(\Omega) \rightarrow [0, \infty]$ is given as in [\(GEN\)](#), where $Z \in \mathcal{L}(\Theta, \mathbb{R}^{N_c})$ and the $\mathcal{J}_l : M^l \rightarrow [0, \infty]$ are weak* lower semicontinuous with $\mathcal{J}_l(0) = 0$.
3. The prior $\mathcal{R} : L^q(\Omega) \rightarrow [0, \infty]$ is proper and weakly lower semicontinuous, and there exists a closed subspace $U \subset L^q(\Omega)$ and a continuous linear projection $P_U : L^q(\Omega) \rightarrow U$ such that for every $u \in L^q(\Omega)$, $v \in U$, $\mathcal{R}(u) = \mathcal{R}(u + v)$ and $\|u - P_U u\|_q \leq C\mathcal{R}(u)$ with $C > 0$ and
 - (i) U is finite dimensional or
 - (ii) $U \cap \ker(A)$ admits a complement Z in U and $\|u\|_q \leq D\|Au\|_Y$ for all $u \in Z$ and some $D > 0$.

Remark 2.10. Note that in [Assumption 2.9](#), 3, (i) implies (ii), so we will assume (ii) in the following. Indeed, if U is finite dimensional, then $U \cap \ker(A)$ admits a complement Z in U . Moreover, the restriction $A|_Z : Z \rightarrow A(Z)$ is a bijective, linear operator between finite dimensional spaces, which implies $\|u\|_q = \|(A|_Z)^{-1}A|_Z u\|_q \leq \|(A|_Z)^{-1}\| \|A|_Z u\|_Y = \|(A|_Z)^{-1}\| \|Au\|_Y$ for all $u \in Z$.

Example 2.11. We list some examples of functionals \mathcal{D}_y , \mathcal{G} , and \mathcal{R} satisfying [Assumption 2.9](#). For the *data discrepancy* term \mathcal{D}_y , power-of-norm discrepancies such as $\mathcal{D}_y(z) = \frac{1}{r} \|z - y\|_Y^r$ with $r \in [1, \infty)$ are feasible. This includes $Y = L^2(\Xi)$ with $\Xi \subset \mathbb{R}^{N_D}$ and $\mathcal{D}_y(z) = \frac{1}{2} \|z - y\|_{L^2(\Xi)}^2$, the standard choice in case the data is perturbed by Gaussian noise. For inpainting, $\mathcal{D}_y(z) = \mathcal{I}_{\{y\}}(z)$ is feasible and, in case the data is perturbed by Poisson distributed noise, it is feasible to choose $\mathcal{D}_y = \text{KL}(\cdot, y)$, with KL being the Kullback–Leibler divergence. For properties of KL that indeed ensure [Assumption 2.9](#) to hold, we refer to [9].

Regarding \mathcal{G} , possible specifications of Z and \mathcal{J}_l are $Z = 0$, $Z\theta = (\int_{\Sigma} \theta_{n,1}^1)_{n=1}^{N_1}$, or $Z\theta = (\int_{\Sigma} \theta_{n,k}^l)_{n,k,l}$, the last two yielding zero-mean constraints for the filter kernels of the last or of all layers, respectively. The \mathcal{J}_l can, for example, be chosen as $\mathcal{J}_l = \mathcal{I}_{\{0\}}$, yielding the constraint $\mu^{l-1} = \mu^l *_l \theta^l$, or as $\mathcal{J}_l = \frac{\gamma}{2} \mathcal{V}_l^2$ with $\gamma > 0$, where, for $\mu = (\mu_1, \dots, \mu_{N_l}) \in M^l$,

$$\mathcal{V}_l(\mu) = \sup \left\{ \sum_{n=1}^{N_l} \int_{\Omega^l} \phi_n(x) d\mu_n(x) \mid \phi = (\phi_1, \dots, \phi_{N_l}) \in C_0(\Omega^l)^{N_l}, \|\phi\|_2 \leq 1 \right\}.$$

Indeed, the \mathcal{V}_l (and, consequently, the \mathcal{J}_l) are weak* lower semicontinuous as being the pointwise supremum of weak* continuous functions. Also note that $\mathcal{V}_l(\mu) = \|\mu\|_2$ in case $\mu \in L^2(\Omega)^{N_l}$, and $\mathcal{V}_l(\mu) = \infty$ else, yielding $\mathcal{J}_l(\mu^{l-1} - \mu^l *_l \theta^l) = \frac{\gamma}{2} \|\mu^{l-1} - \mu^l *_l \theta^l\|_2^2$, in the case $\mu^{l-1} \in L^2(\Omega^{l-1})^{N_{l-1}}$, a relaxed version of the constraint $\mu^{l-1} = \mu^l *_l \theta^l$.

Possible choices for the regularizing functional \mathcal{R} are, for instance, $\mathcal{R} = \mathcal{I}_{\{0\}}$, which means that only \mathcal{G} is used for regularization, or $\mathcal{R} = \text{TV} : L^1(\Omega) \rightarrow [0, \infty]$ in case $1 < q \leq \frac{d}{d-1}$, where d is the dimension of Ω (necessary for Poincaré's inequality). More generally, in the latter case a feasible choice is also $\mathcal{R} = J^{**}$, the bipolar/biconjugate of J , where, for $j : \mathbb{R}^d \rightarrow [0, \infty)$ coercive, convex, Lipschitz continuous, and with linear growth,

$$J : L^1(\Omega) \rightarrow [0, \infty],$$

$$u \mapsto \begin{cases} \int_{\Omega} j(\nabla u) dx & \text{if } u \in W^{1,1}(\Omega), \\ \infty & \text{else.} \end{cases}$$

In this case, J^{**} coincides with the lower semicontinuous regularization of J . For background information and application-specific details in this context we refer to [16] and [17], respectively. Lower semicontinuity and convexity of \mathcal{R} hold by definition and the subspace U from [Assumption 2.9](#) is the space of all constant functions, as in the case $\mathcal{R} = \text{TV}$. For further generalizations, allowing also for a spatial dependence of j in Ω , we refer to [1, 21].

Under [Assumption 2.9](#), existence can now be guaranteed as follows.

Theorem 2.12 (existence of solutions). *Suppose that [Assumption 2.9](#) holds true and assume there exists $\hat{u} \in L^q(\Omega)$ such that $\mathcal{R}(\hat{u}) < \infty$ and $\mathcal{D}_y(A\hat{u}) < \infty$. Then $(P(y))$ admits a minimum.*

Proof. Since, by assumption, $0 \leq \mathcal{E}_y(\hat{u}, 0) < \infty$, we can find a minimizing sequence $(u_m, v_m)_m$ such that $\lim_{m \rightarrow \infty} \mathcal{E}_y(u_m, v_m) = \inf_{(u,v)} \mathcal{E}_y(u, v) \in [0, \infty)$. Since \mathcal{D}_y and \mathcal{R} are nonnegative, the coercivity of \mathcal{G} implies that $(v_m)_m$ is bounded in $L^q(\Omega)$. The sequence $(u_m)_m$, however, is not bounded in general, but we can construct a modified minimizing sequence that is bounded as follows: With Z the complement of $U \cap \ker(A)$ as in [Assumption 2.9](#), let $P_Z : U \rightarrow Z$ be a continuous, linear projection onto Z such that $I - P_Z : U \rightarrow U \cap \ker(A)$ is the corresponding projection onto $U \cap \ker(A)$ (here I denotes the identity). With P_U the projection onto U from [Assumption 2.9](#), we define

$$w_m := u_m - (I - P_Z)P_U u_m = u_m - P_U u_m + P_Z P_U u_m.$$

Since $(I - P_Z)P_U u_m \in U \cap \ker(A)$, we have that $Au_m = Aw_m$ as well as $\mathcal{R}(u_m - v_m) = \mathcal{R}(w_m - v_m)$ and thus $\mathcal{E}_y(u_m, v_m) = \mathcal{E}_y(w_m, v_m)$. Hence, $(w_m, v_m)_m$ is also a minimizing sequence for \mathcal{E}_y . Additionally, we can now show that $(w_m)_m$ is bounded: By [Assumption 2.9](#), we have

$$(2.3) \quad \|(u_m - v_m) - P_U(u_m - v_m)\|_q \leq C \mathcal{R}(u_m - v_m) \leq \frac{C}{s_{\mathcal{R}}(\nu)} \mathcal{E}_y(u_m, v_m)$$

for $C > 0$; therefore, $((u_m - v_m) - P_U(u_m - v_m))_m$ is bounded. Since $(v_m)_m$ is bounded, also $(v_m - P_U v_m)_m$ is bounded and, consequently, so is $(u_m - P_U u_m)_m$. Further, we find that, for constants $C, D > 0$,

$$\begin{aligned} \|P_Z P_U u_m\|_q &\leq D \|AP_Z P_U u_m\|_Y = D \|AP_U u_m\|_Y \leq D (\|A(u_m - P_U u_m)\|_Y + \|Au_m\|_Y) \\ &\leq D (\underbrace{\|A\| \|u_m - P_U u_m\|_q}_{\text{a)}} + \underbrace{\|Au_m\|_Y}_{\text{b)}}), \end{aligned}$$

where (a) is bounded as just shown and (b) is bounded by the coercivity of \mathcal{D}_y and nonnegativity of \mathcal{R} and \mathcal{G} . Hence, $(w_m, v_m)_m$ is a bounded minimizing sequence and since $L^q(\Omega)$ is reflexive, we can pick (nonrelabeled) weakly convergent subsequences of $(w_m)_m$ and $(v_m)_m$ such that $w_m \rightharpoonup w$ and $v_m \rightharpoonup v$ with $w, v \in L^q(\Omega)$. In particular, this also implies $w_m - v_m \rightharpoonup w - v$ and $Aw_m \rightharpoonup Aw$ by weak-to-weak continuity of A . Since all involved functionals \mathcal{D}_y , \mathcal{R} , and \mathcal{G} are weakly lower semicontinuous and $s_{\mathcal{R}}(\nu), s_{\mathcal{G}}(\nu), \lambda > 0$, we find that

$$\mathcal{E}_y(u, v) \leq \liminf_{m \rightarrow \infty} \mathcal{E}_y(w_m, v_m) = \inf_{\tilde{u}, \tilde{v} \in L^q(\Omega)} \mathcal{E}_y(\tilde{u}, \tilde{v}),$$

which concludes the proof. ■

In our applications, we will use $\mathcal{R} = J^{**}$ with an appropriate functional $J : L^1(\Omega) \rightarrow [0, \infty]$ and $\mathcal{J}_l = \frac{\gamma}{2} \mathcal{V}_l^2$ as mentioned in [Example 2.11](#). For this setting, the regularity of solutions of $(P(y))$ can be described as follows.

Proposition 2.13. *In case the network has $L \geq 2$ layers, $\mathcal{R} = \text{TV}$ or, more generally, $\mathcal{R} = J^{**}$ as in [Example 2.11](#) and \mathcal{J}_l is such that $\mathcal{J}_l(\mu) < \infty$ implies $\mu \in L^2(\Omega^l)^{N_l}$, any solution u of $(P(y))$ can be written as $u = (u - v) + v$ with $(u - v) \in \text{BV}(\Omega)$ and $v \in C(\bar{\Omega})$.*

Proof. This is a direct consequence of [Proposition 2.2](#) and the fact that $\mathcal{R}(u - v) < \infty$ implies $u - v \in \text{BV}(\Omega)$; see, for instance, [\[2, 17\]](#). ■

In order to discuss stability of the solution, we first introduce notions of convergence and coercivity of the data fidelity term for varying data, which are taken from [9]. With $(y_m)_m$ and y in Y , we say \mathcal{D}_{y_m} converges to \mathcal{D}_y and write $\mathcal{D}_{y_m} \rightarrow \mathcal{D}_y$ if

$$(2.4) \quad \begin{cases} \mathcal{D}_y(z) \leq \liminf_{m \rightarrow \infty} \mathcal{D}_{y_m}(z_m) & \text{for every } z_m \rightarrow z \text{ in } Y, \\ \mathcal{D}_y(z) \geq \limsup_{m \rightarrow \infty} \mathcal{D}_{y_m}(z) & \text{for every } z \in Y. \end{cases}$$

Moreover, we say that $(\mathcal{D}_{y_m})_m$ is equicoercive if there exists a coercive function $\mathcal{D}_0 : Y \rightarrow [0, \infty]$ such that $\mathcal{D}_0 \leq \mathcal{D}_{y_m}$ for all m .

Note that the notions of convergence and equicoercivity ensure the properties of the data term that are necessary for stability estimates, without requiring us to explicitly consider the interplay of a convergence of $(y_m)_m$ in Y and properties of the data term. For standard choices of \mathcal{D}_y , they can be ensured via convergence of $(y_m)_m$ as follows: In case $\mathcal{D}_y(z) = (1/q)\|y - z\|_Y^q$, $\mathcal{D}_{y_m} \rightarrow \mathcal{D}_y$ and equicoercivity hold, whenever $y_m \rightarrow y$ in Y as $m \rightarrow \infty$. In case $\mathcal{D}_y(z) = \text{KL}(z, y)$ with $Y = L^1(\Xi)$, $\mathcal{D}_{y_m} \rightarrow \mathcal{D}_y$ and equicoercivity hold, whenever $\text{KL}(y, y^m) \rightarrow 0$ as $m \rightarrow \infty$ and $y_m \leq Cy$ for some $C > 0$ and all m ; see [9, Example 2.16].

Within the proof of stability, we will further make use of the following lemma.

Lemma 2.14. *Let $(a_m)_m$ and $(b_m)_m$ be real sequences and $a, b \in \mathbb{R}$. Assume further, $a_m + b_m \rightarrow a + b$ and $a \leq \liminf_{m \rightarrow \infty} a_m$, $b \leq \liminf_{m \rightarrow \infty} b_m$. Then, $a_m \rightarrow a$ and $b_m \rightarrow b$.*

Proof. We simply compute

$$a \leq \liminf_{m \rightarrow \infty} (a_m + b_m - b_m) = a + b + \liminf_{m \rightarrow \infty} (-b_m) = a + b - \limsup_{m \rightarrow \infty} b_m.$$

Hence, $\limsup_{m \rightarrow \infty} b_m \leq b \leq \liminf_{m \rightarrow \infty} b_m$ and accordingly $b_m \rightarrow b$. As a result, also $a_m \rightarrow a$. ■

Theorem 2.15 (stability). *Let Assumption 2.9 hold and assume that $\mathcal{D}_{y_m} \rightarrow \mathcal{D}_y$ in the sense of (2.4) and that $(\mathcal{D}_{y_m})_m$ is equicoercive. Let, for each m , (u_m, v_m) be a minimizer of \mathcal{E}_{y_m} . Then we have either*

- (i) $\mathcal{E}_{y_m}(u_m, v_m) \rightarrow \infty$ as $m \rightarrow \infty$ and $\mathcal{E}_y(u, v) = \infty$ for any u, v , or
- (ii) $\mathcal{E}_{y_m}(u_m, v_m) \rightarrow \min_{\tilde{u}, \tilde{v} \in L^q(\Omega)} \mathcal{E}_y(\tilde{u}, \tilde{v}) < \infty$ as $m \rightarrow \infty$ and, up to shifts in $(U \cap \ker(A)) \times \{0\}$, $(u_m, v_m)_m$ admits a weak accumulation point $(u, v) \in L^q(\Omega)^2$ that minimizes \mathcal{E}_y .

Moreover, in the latter case, for each subsequence $(u_{m_k}, v_{m_k})_k$ converging weakly to some $(u, v) \in L^q(\Omega)^2$, we have that (u, v) is a minimizer of \mathcal{E}_y and $\mathcal{D}_{y_{m_k}}(Au_{m_k}) \rightarrow \mathcal{D}_y(Au)$, $\mathcal{R}(u_{m_k} - v_{m_k}) \rightarrow \mathcal{R}(u - v)$ and $\mathcal{G}(v_{m_k}) \rightarrow \mathcal{G}(v)$ as $m \rightarrow \infty$. If \mathcal{E}_y admits a unique minimizer, then $(u_m, v_m) \rightarrow (u, v)$.

Proof. Assume first that $\mathcal{E}_{y_m}(u_m, v_m) \rightarrow \infty$ as $m \rightarrow \infty$. In case there exist $u, v \in L^q(\Omega)$ such that $\mathcal{E}_y(u, v) < \infty$, from convergence of the data fidelity term it follows that

$$\limsup_{m \rightarrow \infty} \mathcal{E}_{y_m}(u_m, v_m) \leq \underbrace{\limsup_{m \rightarrow \infty} \mathcal{E}_{y_m}(u, v)}_{\text{optimality}} \leq \mathcal{E}_y(u, v) < \infty,$$

which contradicts our assumption. Therefore, \mathcal{E}_y cannot admit a finite value, showing (i).

Now assume to the contrary $\mathcal{E}_{y_m}(u_m, v_m) \not\rightarrow \infty$, that is, $\liminf_{m \rightarrow \infty} \mathcal{E}_{y_m}(u_m, v_m) < \infty$, and let $(u_{m_k}, v_{m_k})_k$ be a subsequence such that

$$\liminf_{m \rightarrow \infty} \mathcal{E}_{y_m}(u_m, v_m) = \lim_{k \rightarrow \infty} \mathcal{E}_{y_{m_k}}(u_{m_k}, v_{m_k}).$$

We define $w_m = u_m - (I - P_Z)P_U u_m$ as in the proof of [Theorem 2.12](#) such that for each m , (w_m, v_m) is again a minimizer of \mathcal{E}_{y_m} . Using the equicoercivity of $(\mathcal{D}_{y_m})_m$, we find that $(w_{m_k}, v_{m_k})_k$ is bounded and hence admits a (nonrelabeled) subsequence weakly converging to some (u, v) . Now, let $\tilde{u}, \tilde{v} \in L^q(\Omega)$ be arbitrary. Using the convergence of the data term, weak-to-weak continuity of the operator A , and the weak lower semicontinuity of \mathcal{R} and \mathcal{G} , we obtain

$$\begin{aligned} \mathcal{E}_y(u, v) &\leq \liminf_{k \rightarrow \infty} \mathcal{E}_{y_{m_k}}(w_{m_k}, v_{m_k}) = \liminf_{m \rightarrow \infty} \mathcal{E}_{y_m}(w_m, v_m) \\ (2.5) \quad &\leq \limsup_{m \rightarrow \infty} \mathcal{E}_{y_m}(w_m, v_m) \underbrace{\leq}_{\text{optimality}} \limsup_{m \rightarrow \infty} \mathcal{E}_{y_m}(\tilde{u}, \tilde{v}) \leq \mathcal{E}_y(\tilde{u}, \tilde{v}). \end{aligned}$$

This shows that (u, v) is a minimizer of \mathcal{E}_y . Moreover, if we plug in $(\tilde{u}, \tilde{v}) = (u, v)$, we find that

$$\min_{\tilde{u}, \tilde{v} \in L^q(\Omega)} \mathcal{E}_y(\tilde{u}, \tilde{v}) = \mathcal{E}_y(u, v) = \lim_{m \rightarrow \infty} \mathcal{E}_{y_m}(u_m, v_m),$$

which finishes the proof of (ii).

To conclude the proof, assume that $(u_{m_k}, v_{m_k})_k$ is a subsequence of $(u_m, v_m)_m$, weakly converging to (u, v) . Then

$$\mathcal{E}_y(u, v) \leq \liminf_{k \rightarrow \infty} \mathcal{E}_{y_{m_k}}(u_{m_k}, v_{m_k}) = \lim_{m \rightarrow \infty} \mathcal{E}_{y_m}(u_m, v_m) = \min_{\tilde{u}, \tilde{v} \in L^q(\Omega)} \mathcal{E}_y(\tilde{u}, \tilde{v}).$$

Hence, (u, v) is a minimizer of \mathcal{E}_y and [Lemma 2.14](#) implies the convergence of the three parts $(\mathcal{D}_{y_{m_k}}(Au_{m_k}))_k$, $(\mathcal{R}(u_{m_k}))_k$ and $(\mathcal{G}(u_{m_k}))_k$ individually. Finally, assume (u, v) is the unique minimizer of \mathcal{E}_y . Then we find that $U \cap \ker(A) = \{0\}$, since if there was $0 \neq w \in U \cap \ker(A)$, then $(u + w, v)$ would be a second, distinct solution. Therefore,

$$u_m = u_m - (I - P_Z)P_U u_m$$

and consequently $(u_m, v_m)_m$ is already bounded. Thus, every subsequence of $(u_m, v_m)_m$ contains another subsequence weakly converging to (u, v) , which implies $(u_m, v_m) \rightharpoonup (u, v)$ by a standard contradiction argument. ■

Convergence of solutions for vanishing noise can be proven with similar techniques as in [Theorem 2.15](#) and we refer to section [SM1](#) of the supplementary material for an additional result in that direction.

3. Numerical results. In this section, we provide numerical results for imaging applications such as inpainting, denoising, deconvolution, and JPEG decompression. Further results on superresolution can be found in the supplement subsection [SM4.4](#). For all results, we use a particular instance of the proposed approach with the following architecture:

- (i) $L = 3$ and $N_l = 8$ for each $l = 1, 2, 3$.
- (ii) $E_l = \{(n, n) \mid n \in \{1, \dots, 8\}\}$ for $l = 2, 3$ and $E_1 = \{(n, 1) \mid n \in \{1, \dots, 8\}\}$.
- (iii) $\mathcal{R} = J^{**}$, where, for $\epsilon > 0$, $J : L^1(\Omega) \rightarrow [0, \infty]$ is given as

$$(3.1) \quad J(u) = \begin{cases} \frac{1}{|\Omega|} \int_{\Omega} \sqrt{|\nabla u|^2 + \epsilon} \, dx & \text{if } u \in W^{1,1}(\Omega), \\ \infty & \text{else.} \end{cases}$$

- (iv) For $l = 1, 2$, as detailed in [Example 2.11](#),

$$\mathcal{J}_l(\mu) = \begin{cases} \frac{\gamma}{2} \|\mu\|_2^2 & \text{if } \mu \in L^2(\Omega^l)^{N_l}, \\ \infty & \text{else.} \end{cases}$$

The choices (i) and (ii) were made as they result in a rather simple network that still delivers significantly improved results compared, e.g., to a single-layer setting [13]. Note in particular that, with our choice of E_l , different latent variables are only connected after the last layer. Besides aiming at a simple model with a reduced number of parameters, this choice is motivated by the observation that for deeper networks, as opposed to fully connected latent variables, it allows for a clear differentiation of features; see [Figure 8\(a\)](#) for an example.

The choices of \mathcal{R} and \mathcal{J}_l are a compromise to allow for a rather simple numerical solution algorithm with convergence guarantees. From the modeling perspective, for \mathcal{R} choices like TV or higher-order derivative-based regularization functionals (see [9]) as well as the choice $\mathcal{J}_l = \mathcal{I}_{\{0\}}$ seem preferable but are more difficult to realize numerically, as we are dealing with a nonsmooth, nonconvex optimization problem.

Using this particular architecture, the energy in $(P(y))$ is discretized for images $u \in \mathbb{R}^{N_x \times N_y}$, where the forward operator A and the data discrepancy \mathcal{D}_y are chosen application specific as provided in subsections 3.1 to 3.4. In addition, as further means of dimensionality reduction in the network, we introduce a stride in the convolutions between different layers and denote the corresponding strided upconvolution as $*_{\sigma}$; see section [SM2](#) for details.

Replacing v in $(P(y))$ with its representation by (μ, θ) and using the same notation for the discretized functionals as for the continuous counterparts, the corresponding minimization problem used in the implementation reads as

$$\begin{aligned} (\text{DP}(y)) \quad \min_{u, \mu, \theta} \quad & \lambda \mathcal{D}_y(Au) + s_{\mathcal{R}}(\nu) J \left(u - \sum_{n=1}^{N_1} \mu_n^1 *_{\sigma} \theta_n^1 \right) + s_{\mathcal{G}}(\nu) \sum_{l=1}^L \sum_{n=1}^{N_l} \|\mu_n^l\|_1 \\ & + \gamma \sum_{l=2}^L \sum_{n=1}^{N_l} \frac{1}{2} \|\mu_n^{l-1} - \mu_n^l *_{\sigma} \theta_n^l\|_2^2 \end{aligned}$$

subject to

$$\begin{cases} \|\theta_n^l\|_2^2 \leq 1 & \text{for all } l, n, \\ \sum_{i,j=1}^r (\theta_n^1)_{i,j} = 0 & \text{for all } n. \end{cases}$$



Figure 5. Test images. From left to right, we refer to them as Barbara, mix, fish (*“Pomocanthus imperator facing right,”* by Albert Kok, licensed under CC BY-SA 4.0), patchtest, and zebra [27].

The discretizations of the ∇ , the different norms, and integrals are rather standard and we refer to the supplement section SM2 for details.

A numerical solution of the discretized problem is obtained using the iPALM algorithm [33], for which, in case of boundedness, convergence to a stationary point can be ensured. We implemented the method in Python for GPUs based on PyOpenCL [24]. For additional details on the discretization, the algorithm, and the implementation, we refer to the supplementary material of this paper and the publicly available source code [18]. We also note that, in the discrete setting, we use a strided convolution (see (SM2.3)) as a means of increasing the resolution of the latent variables from the deepest to the last layer, thereby reducing the dimensionality of the network.

The test images we use for the numerical experiments can be found in Figure 5, and all experiments for the proposed generative prior as shown in this paper can be reproduced with the publicly available source code [18]. Note that the experiments with the *patchtest* and *zebra* images are omitted here for the sake of brevity and can be found in the supplementary material section SM4.

Parameter choice. Our method has three classes of parameters: (i) The network architecture, which is fixed for all experiments and was chosen in view of obtaining a rather simple network, that still delivers good results; see Table SM1. (ii) The parameters ϵ and $\gamma > 0$, which are necessary to achieve a smoothing of nonsmooth functionals and constraints. We found their choice to be rather uncritical as long as they are sufficiently small and large, respectively, and again fixed those parameters for all experiments in the paper; see again Table SM1. (iii) The parameters λ and ν , which are the two effective parameters of our method. They define the trade-off between data fidelity and regularization and between the priors \mathcal{R} and \mathcal{G} , respectively. The choice of a regularization parameter λ is necessary for any regularization method and can be made, e.g., according to classical parameter choice strategies, such as the discrepancy principle or the L-curve approach. The parameter ν can be interpreted as a model parameter that can be fixed for a class of images under consideration, independent of the noise level. In our experiments, for each application, we allow only a single parameter to vary (in order to obtain visually optimal results), while the other, if applicable, was fixed; see Table SM2.

Comparison to related methods. We compare our method to the DIP method [26], to convex learning (CL) [13], which can be seen as convex relaxation of a single-layer version of the proposed method, and to total generalized variation (TGV) regularization [10]. Details on the implementation used for these methods, as well as the parameter choice, can be found in section SM4, and in particular Table SM3. In short, for each competing method,

parameters were chosen in order to obtain visually optimal results, and for DIP in particular, also the network architecture was adapted to each experiment as suggested by the authors. Results for a larger experiment for inpainting with fixed parameters on subsets of the ImageNet ILSVRC2017 DET test data set [35] can be found in Table 2; see subsection 3.1 for details. Note that we did not compare deconvolution to DIP since this experiment was also not presented in the original paper. Moreover, we compare to CL only on images which are also used in the original paper since the authors published their code for these images with optimal parameters.

3.1. Inpainting. In the case of inpainting, we aim to reconstruct an image from only a part of its pixels. Formally, given $\mathcal{M} \in \{0, 1\}^{N_x \times N_y}$, such that $\mathcal{M}_{i,j} = 1$ whenever the image at pixel (i, j) is known and $\mathcal{M}_{i,j} = 0$ else, we define

$$Au = \mathcal{M} \odot u, \quad \mathcal{D}_y(z) = \mathcal{I}_{\{y\}},$$

where \odot denotes the Hadamard product and \mathcal{M} is initialized such that around 70% of the entries are zero. Results can be found in Figure 6 (and Figure SM3, including the *patchtest* and *zebra* images) with corresponding PSNR and SSIM values shown in Table 1.

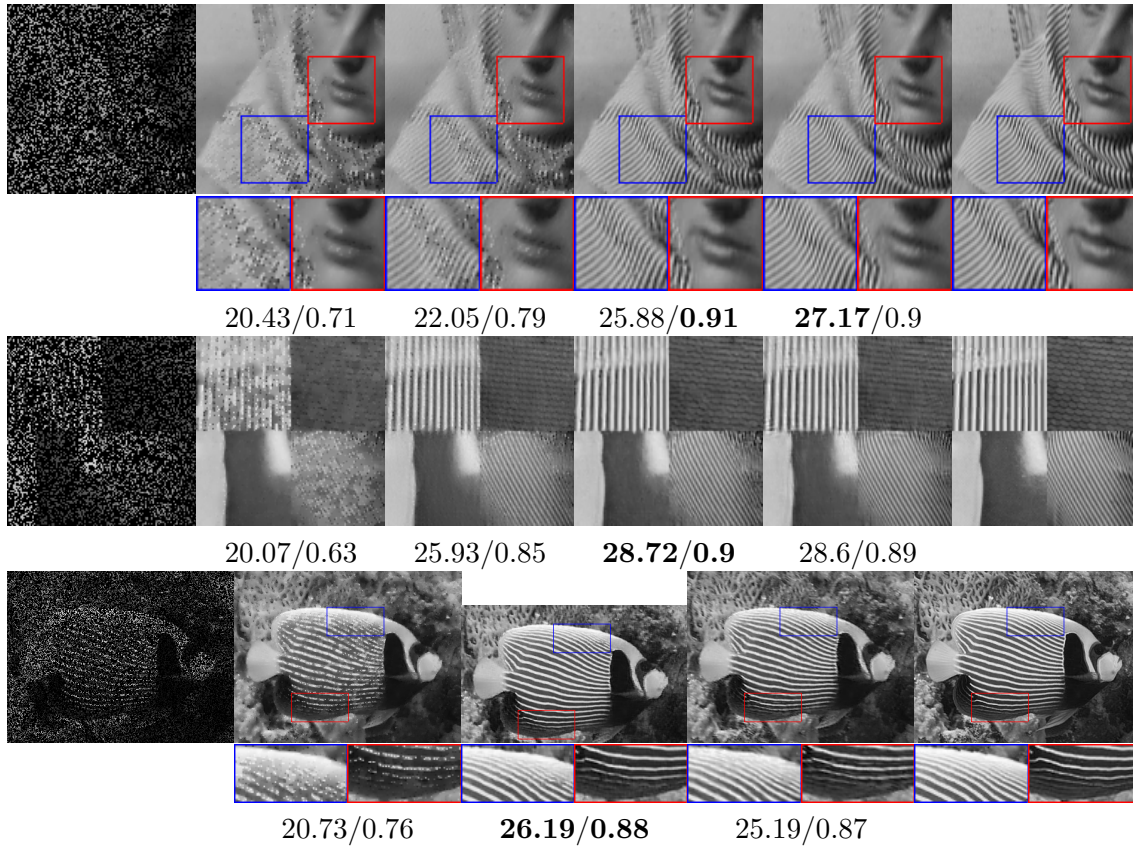


Figure 6. Inpainting from 30% known pixels with closeups. First two rows, from left to right: Data, TGV, CL, DIP, proposed, ground truth. Last row, from left to right: Data, TGV, DIP, proposed, ground truth. PSNR/SSIM values below images, bold indicates best result.

Table 1

PSNR/SSIM values of all results shown in the paper. Bold indicates the best result. A “-” indicates that the experiment was not carried out since it was not part of the original reference used for comparison or that no ground truth is available.

	<i>Barbara</i>	<i>mix</i>	<i>patchtest</i>	<i>fish</i>	<i>zebra</i>
Inpainting					
TGV	20.43/0.71	20.07/0.63	19.21/0.64	20.73/0.76	25.66/0.83
CL	22.05/0.79	25.93/0.85	26.13/0.90	-	-
DIP	25.88/ 0.91	28.72/0.9	25.78/0.9	26.19/0.88	27.73/0.86
proposed	27.17/0.9	28.6/0.89	26.97/0.91	25.19/0.87	28.59/0.91
Denoising					
TGV	23.64/0.75	22.88/0.62	21.7/0.69	25.06/0.82	26.4/0.79
CL	23.24/0.72	25.41/0.75	24.66/0.83	-	-
DIP	26.68/0.82	26.21/0.76	26.54/0.84	26.8/0.85	28.16/0.80
proposed	26.21/0.81	26.83/0.81	25.64/ 0.84	25.52/0.82	27.19/ 0.81
Deconvolution					
TGV	22.47/0.73	23.56/0.70	23.97/ 0.84	23.73/0.80	26.44/ 0.83
CL	22.65/0.73	24.46/0.72	-	-	-
proposed	23.96/0.79	25.28/0.77	24.61/0.83	23.77/0.81	26.58/0.82
Super-resolution					
DIP	-	-	-	25.88/0.78	24.33/0.71
proposed	-	-	-	26.11/0.82	23.79/ 0.77
JPEG-decompression					
DIP	24.36/ 0.81	25.19/ 0.81	24.77/0.87	26.43/0.89	29.00/0.86
proposed	24.84/0.80	26.23/0.81	24.3/0.85	25.66/0.85	27.67/0.85

Both visually and in terms of PSNR/SSIM, the methods CL and DIP and the proposed method clearly outperform TGV regularization (in particular in texture parts), which is as expected as the latter constitutes a model for piecewise smooth images. Compared to CL, the DIP and the proposed method also deliver superior results, which is manifested in particular in a further improved reconstruction of texture parts and a smoother result overall. The latter can be attributed to the fact that CL uses a single layer, while DIP and the proposed method use multiple layers. Compared to DIP, the proposed method performs at least equally well, both visually and in terms of PSNR/SSIM, where notable differences are a slightly improved reconstruction of stripe structures with our method and slightly fewer artifacts with the DIP prior, for instance, in the mouth region of the *Barbara* image.

In order to show that our method generalizes well, we also compare on two larger data sets to TGV and DIP. We used images from the ImageNet ILSVRC2017 DET test data set [35]. The data set consists of 5500 images from which we took two different samples. For one sample we picked 100 images at random and for the second sample we handpicked 26 images containing distinctive texture parts. The 26 texture images are not contained in the random sample. We transformed all images to grayscale images and cropped them to a size of 256x256. Again we removed around 70% of known pixels from the images for the inpainting task. We let all three methods run for 5000 iterations. For the proposed method the parameter ν was set to 0.9 for all experiments. The resulting mean PSNR and SSIM scores and

Table 2

PSNR and SSIM values for inpainting from 30% known pixels on subsets of the ImageNet data set. We provide the mean and standard deviation over the data set as mean \pm standard deviation. Bold indicates the best result.

		TGV	DIP	Proposed
PSNR	Random images	24.58 \pm 3.69	25.19 \pm 4.08	25.27 \pm 3.93
	Texture images	22.04 \pm 4.08	23.94 \pm 4.78	23.71 \pm 4.28
SSIM	Random images	0.81 \pm 0.081	0.83 \pm 0.092	0.84 \pm 0.078
	Texture images	0.76 \pm 0.075	0.83 \pm 0.076	0.83 \pm 0.063

corresponding standard deviations with TGV, DIP, and the proposed method can be found in Table 2. Compared to TGV both DIP and the proposed method perform significantly better in particular on the texture images, for which TGV naturally is a suboptimal prior. On the texture images, DIP performs slightly better than the proposed method in terms of PSNR scores and comparably well in terms of SSIM scores. On the other hand, on the random sample, the proposed method performs slightly better in terms of both PSNR and SSIM scores. It can also be noted that the standard deviation of PSNR/SSIM values for the proposed method is consistently slightly below the ones obtained with the DIP, indicating an increased stability of our method. Altogether, these results confirm an at least comparable performance of our method compared to DIP.

3.2. Denoising. In the case of denoising, the forward operator A is set to be the identity and the data fidelity term is given as $\mathcal{D}_y(z) = \frac{1}{2}\|z - y\|_2^2$, where the data is corrupted with Gaussian noise with mean zero and standard deviation 0.1 times the image range. Results can be found in Figure 7 (and Figure SM4, including the *patchtest* and *zebra* images) with corresponding PSNR and SSIM values shown in Table 1.

Similarly as in the inpainting experiment, we observe that the DIP and the proposed method clearly outperform CL and TGV. In terms of PSNR values, the DIP seems slightly superior to the proposed method in this case, while in terms of SSIM values and visually the results seem comparable, again with slightly better results of our method for stripe structures and overall slightly smoother reconstructions obtained with DIP. It is also interesting to note that, in particular in the result for the *Barbara* image, with DIP, CL, and the proposed method, some structures of the texture part are also subtly visible in other parts of the image. This can be seen in particular in the smooth top-left region as well as *Barbara*'s face. While such artifacts can, in part, certainly be attributed to the rather strong noise level of the data in this case, they also seem related to the overall design of the three approaches.

Figure 8(a) shows the generative network, consisting of filter kernels, latent variables, and network output, that was obtained while denoising the *mix* image with the proposed approach. We can observe that different latent variables and filter kernels clearly correspond to different features of the image, and that the main features of the image are captured rather well by the model. This is remarkable in particular in view of the fact that no training was used and that the network can only rely on noisy data to learn these features.

We can also observe, in particular when considering the latent variables μ_2^1 and $\mu_2^2 * \sigma \theta_2^2$ in the second row of the first layer of the network, that the output of the previous layer does not

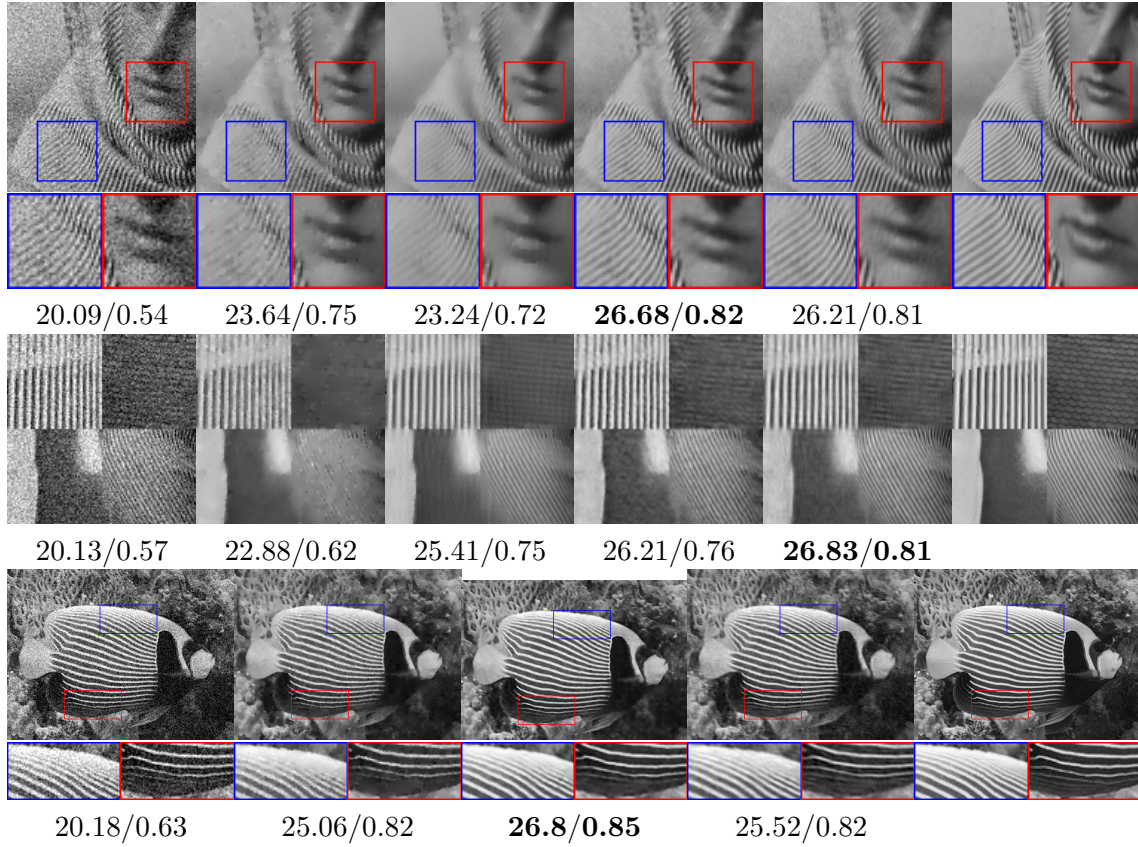


Figure 7. Denoising with Gaussian noise with mean zero and standard deviation 0.1 times image range. First two rows, from left to right: Data, TGV, CL, DIP, proposed, ground truth. Last two row, from left to right: Data, TGV, DIP, proposed, ground truth. PSNR/SSIM values below images, bold indicates best result.

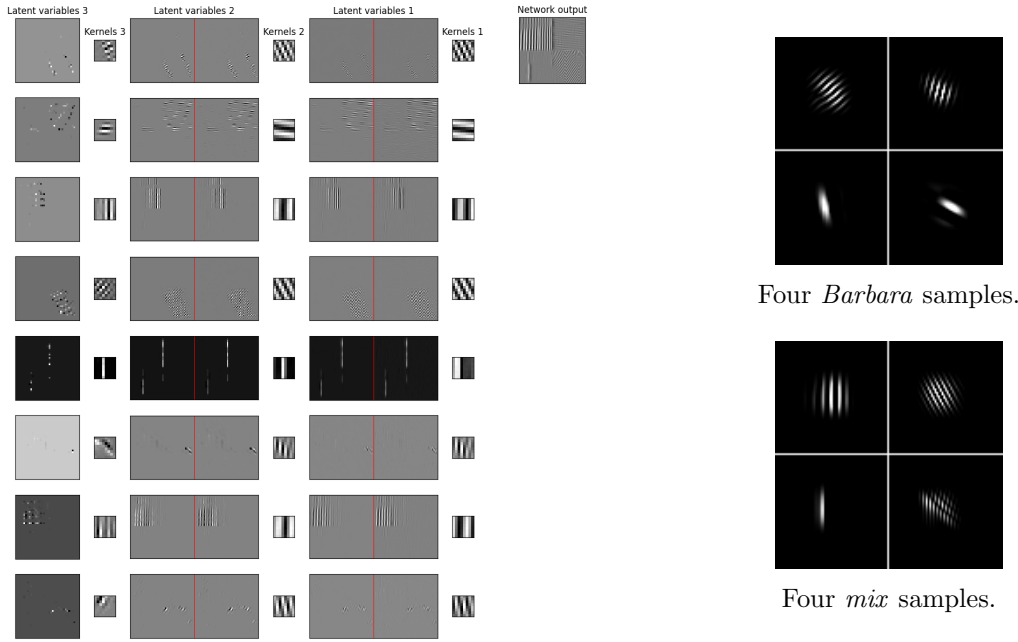
always coincide exactly with the latent variable that is passed forward to the next layer. This is a result of using $\mathcal{J}_l = \frac{\gamma}{2} \|\cdot\|_2^2$ rather than $\mathcal{J}_l = \mathcal{I}_{\{0\}}$ in our numerical realization. It is worth noting that the difference of the two, however, mostly corresponds to noise-like artifacts. This indicates that these noise-like artifacts are difficult to generate from consecutive convolutions, due to the smoothing properties of the convolution (see Figure 4). It also suggests that, by allowing for $\mathcal{J}_l = \mathcal{I}_{\{0\}}$ with an improved algorithm, our results might further be improved.

In Figure 8(b), we draw samples from two different learned networks, where the kernels were learned from denoising the *Barbara* and *mix* images.

3.3. Deconvolution. Given a convolution kernel $k \in \mathbb{R}^{(2s+1) \times (2s+1)}$, we define the forward operator as $A : \mathbb{R}^{N_x \times N_y} \rightarrow \mathbb{R}^{N_x \times N_y}$

$$(Au)_{i,j} = \sum_{i'=-s}^s \sum_{j'=-s}^s k_{s+1+i',s+1+j'} u_{i-i',j-j'},$$

where we set $u_{i,j} = 0$, whenever $(i,j) \notin \{1,2,\dots,N_x\} \times \{1,2,\dots,N_y\}$. The data discrepancy is given as $\mathcal{D}_y(z) = \frac{1}{2} \|z - y\|_2^2$. We use a Gaussian convolution kernel, with size 9×9 for the



(a) *Mix* network. The intermediate layers show the output of the previous convolution $\mu_n^{l+1} *_{\sigma} \theta_n^{l+1}$ (left) and the latent variable of the next convolution μ_n^l (right) next to each other, separated by a red line (recall that the discrepancy of these two is penalized in our objective functional by the functional \mathcal{J}_l).

(b) Network samples (images are cropped): The images v were obtained via $v = \mu^1 *_{\sigma} \theta^1$ with $\mu^{l-1} = \mu^l *_{\sigma} \theta^l$ for $l > 1$, with θ obtained from denoising. The μ^L were chosen as delta peaks in different latent variables μ_n^L .

Figure 8. Convolutional networks obtained from denoising. Left: Full network after denoising. Right: Samples from obtained networks with μ^L delta peaks at different latent variables.

Barbara and *mix* images, 13×13 for the *fish* image, and 15×15 for the *zebra* image, and a standard deviation of 0.25. The data is corrupted with Gaussian noise with standard deviation 0.025 times the image range and mean zero. The results of the deconvolution experiment are shown in Figure 9 (and Figure SM5, including the *zebra* image) with PSNR and SSIM values shown in Table 1.

We can observe that, for the *Barbara* and the *mix* image, the results with the proposed method are clearly superior to CL and TGV in terms of PSNR/SSIM values, and also visually the results are superior in particular for the texture parts. For the *fish* image the improvement is only minor both in terms of PSNR and visually. One reason for this might be the rather different size of this image and the fact that we do not adapt our network architecture between different experiments. This might be further improved by automatically adapting, e.g., the filter-kernel size to the image dimension, which is something we will consider in the course of developing improved overall algorithms for the proposed method within the scope of future work.

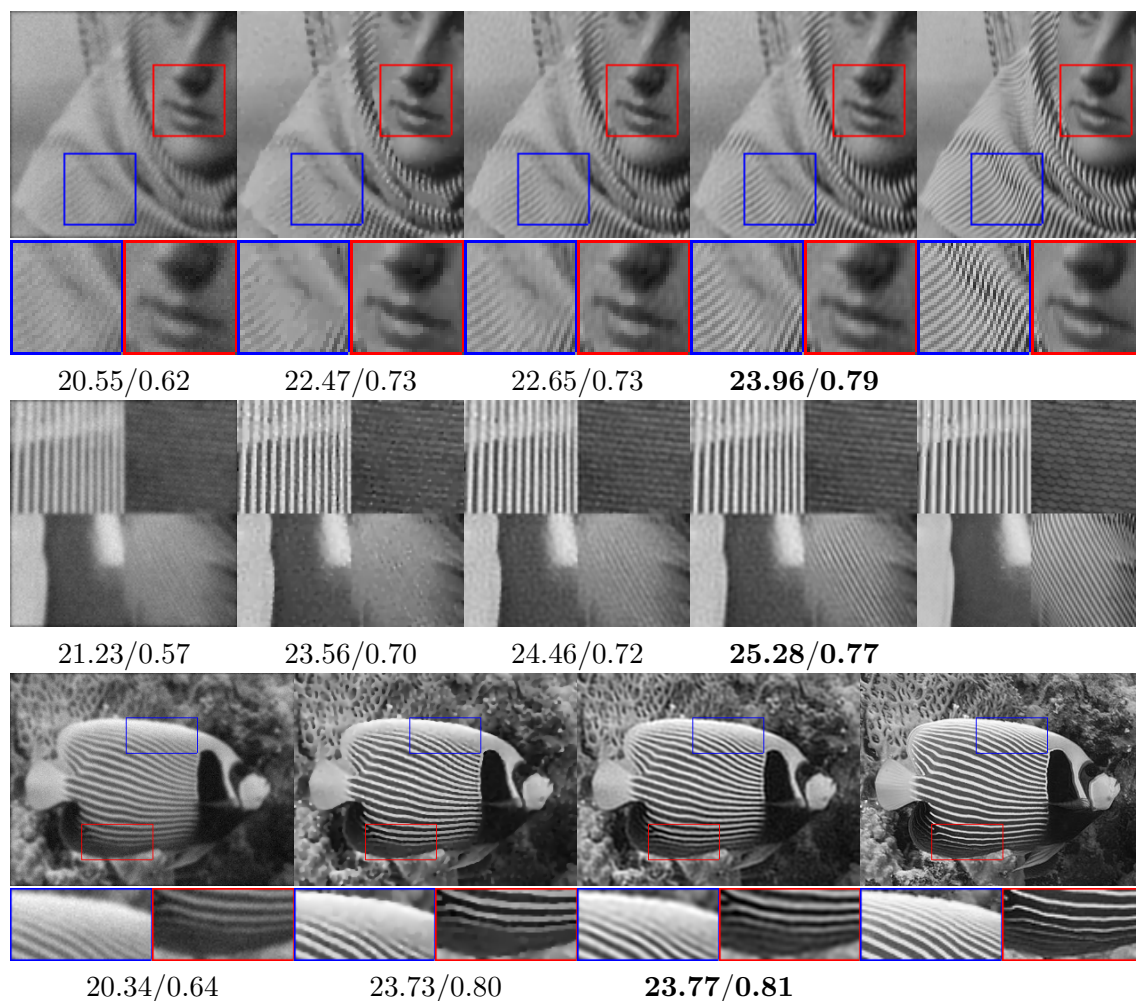


Figure 9. Deconvolution. First two rows from left to right: Data, TGV, CL, proposed, ground truth. Last row from left to right: Data, TGV, proposed, ground truth. PSNR/SSIM values below images, bold indicates best result.

3.4. JPEG decompression. The last experiment we consider is JPEG decompression. Here, the forward operator A is a block-cosine transform, while $\mathcal{D}_y = \mathcal{I}_{D(y)}$, with $D(y)$ containing all cosine coefficients which, when quantized with the given compression rate, would result in the same coefficients as stored in the given JPEG file y . For details we refer to subsection SM4.5 and [8]. The results for JPEG decompression can be found in Figure 10 (and Figure SM7, including the *patchtest* image), and again the corresponding PSNR and SSIM values are provided in Table 1.

For JPEG decompression, it can be observed that the proposed method is rather clearly superior to DIP visually. In particular, JPEG compression artifacts are still visible in the DIP result, but not the results with the proposed method. With respect to PSNR/SSIM values DIP seems to perform slightly better. One should be aware, however, that both these discrepancies in performance might be a result of the different data fidelities used in DIP and our model; see subsection SM4.5 for details.

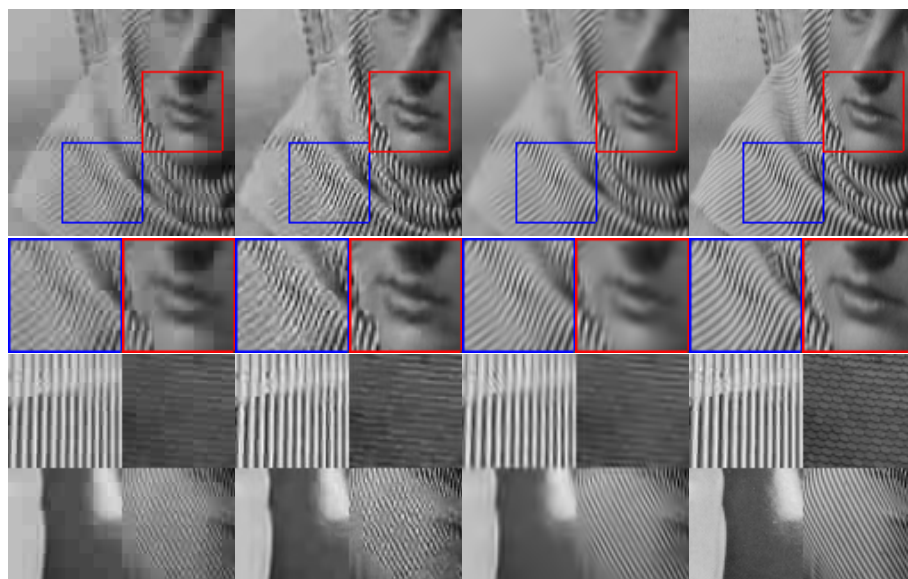


Figure 10. JPEG decomposition. From left to right: Data, DIP, proposed, ground truth.

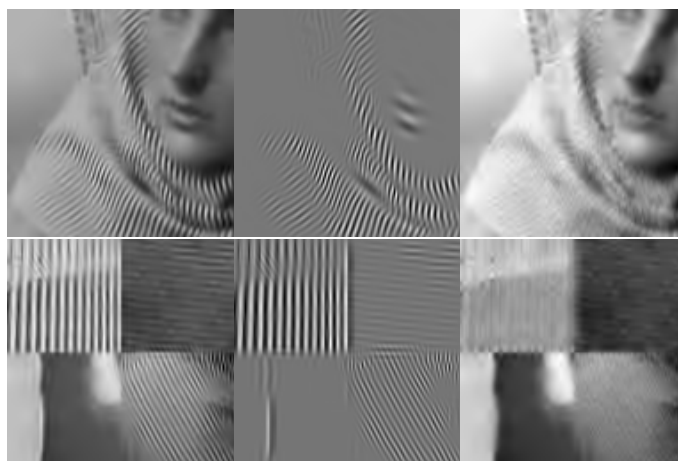


Figure 11. Image decomposition for JPEG decomposition. From left to right: Reconstructed image u , generative part v , remaining part $u - v$.

In Figure 11 one can observe the implicit decomposition of the images into two parts, one being generated from our network and denoted v and the other one being penalized by \mathcal{R} denoted $u - v$. The shown decompositions are obtained from the experiments of Figure 10. One can observe that v contains mostly texture and finer patterns of the image whereas $u - v$ captures piecewise smooth and/or constant regions. For instance, for the *Barbara* image we can clearly see that most of the face is contained in $u - v$ and the texture of the dress is mostly contained in v . As for the *mix* image, a similar decomposition can be observed in that the textures are contained in v and piecewise smooth/constant areas in $u - v$.

4. Discussion. We have introduced and investigated a novel generative variational regularization method for inverse problems in imaging. The method utilizes an energy-based prior whose architecture is inspired by the one of generative neural networks. We have proven theoretical results concerning existence/stability of solutions and convergence for vanishing noise. Moreover, we have shown that the proposed prior always generates continuous functions as outputs, which analytically confirms its efficacy for removing noise-like artifacts and sheds light on different numerical techniques and heuristics used in a large class of existing works related to the deep image prior [26].

For a discretized version of the proposed method, we have derived a numerical solution algorithm that allows a direct application to a rather general class of inverse problems in imaging. Experimentally, we have shown that our method outperforms classical variational methods and competes with recent state-of-the-art deep learning methods such as the DIP, while having the advantage of a significantly reduced number of parameters and of a solid mathematical background.

Our next research goals are to further improve the proposed class of generative variational priors both in terms of analytic understanding, e.g., with respect to the structure of solutions, and in terms of numerical algorithms, e.g., to ensure general, resolution-independent applicability without manual parameter selection. In particular, we aim at developing convex relaxations of the proposed prior in the spirit of [14].

REFERENCES

- [1] M. AMAR, V. DE CICCIO, AND N. FUSCO, *Lower semicontinuity and relaxation results in bv for integral functionals with bv integrands*, ESAIM Control Optim. Calc. Var., 14 (2008), pp. 456–477, <https://doi.org/10.1051/cocv:2007061>.
- [2] L. AMBROSIO, N. FUSCO, AND D. PALLARA, *Functions of Bounded Variation and Free Discontinuity Problems*, Oxford Math. Monogr., Oxford University Press, New York, 2000.
- [3] S. ARRIDGE, P. MAASS, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *Solving inverse problems using data-driven models*, Acta Numer., 28 (2019), pp. 1–174, <https://doi.org/10.1017/S0962492919000059>.
- [4] M. ASIM, M. DANIELS, O. LEONG, A. AHMED, AND P. HAND, *Invertible generative models for inverse problems: Mitigating representation error and dataset bias*, in Proceedings of the International Conference on Machine Learning, PMLR, 2020, pp. 399–409.
- [5] D. O. BAGUER, J. LEUSCHNER, AND M. SCHMIDT, *Computed Tomography Reconstruction Using Deep Image Prior and Learned Reconstruction Methods*, <https://arxiv.org/abs/2003.04989>, 2020.
- [6] C. BOYER, A. CHAMBOLLE, Y. D. CASTRO, V. DUVAL, F. DE GOURNAY, AND P. WEISS, *On representer theorems and convex regularization*, SIAM J. Optim., 29 (2019), pp. 1260–1281, <https://doi.org/10.1137/18M1200750>.
- [7] K. BREDIES AND M. CARIONI, *Sparsity of solutions for variational inverse problems with finite-dimensional data*, Calc. Var. Partial Differential Equations, 59 (2020), pp. 1–26, <https://doi.org/10.1007/s00526-019-1658-1>.
- [8] K. BREDIES AND M. HOLLER, *A total variation-based JPEG decompression model*, SIAM J. Imaging Sci., 5 (2012), pp. 366–393, <https://doi.org/10.1137/110833531>.
- [9] K. BREDIES AND M. HOLLER, *Higher-order total variation approaches and generalisations*, Inverse Problems. Topical Rev., 36 (2020), 123001, <https://doi.org/10.1088/1361-6420/ab8f80>.
- [10] K. BREDIES, K. KUNISCH, AND T. POCK, *Total generalized variation*, SIAM J. Imaging Sci., 3 (2010), pp. 492–526, <https://doi.org/10.1137/090769521>.
- [11] K. BREDIES AND D. LORENZ, *Mathematical Image Processing*, Springer, New York, 2018.
- [12] P. CASCARANO, A. SEBASTIANI, AND M. C. COMES, *ADMM-DIPTV: Combining Total Variation and Deep Image Prior for Image Restoration*, <https://arxiv.org/abs/2009.11380>, 2020.

- [13] A. CHAMBOLLE, M. HOLLER, AND T. POCK, *A convex variational model for learning convolutional image atoms from incomplete data*, J. Math. Imaging Vision, 62 (2020), pp. 417–444, <https://doi.org/10.1007/s10851-019-00919-7>.
- [14] A. CHAMBOLLE, M. HOLLER, AND T. POCK, *Source Code to Reproduce the Results of “A Convex Variational Model for Learning Convolutional Image Atoms from Incomplete Data,”* https://github.com/hollerm/convex_learning, 2020.
- [15] S. DITTMER, T. KLUTH, P. MAASS, AND D. O. BAGUER, *Regularization by architecture: A deep prior approach for inverse problems*, J. Math. Imaging Vision, (2019), pp. 1–15, <https://doi.org/10.1007/s10851-019-00923-x>.
- [16] I. EKELAND, R. TEMAM, AND D. PALLARA, *Convex Analysis and Variational Problems*, SIAM, Philadelphia, 1987.
- [17] A. HABRING, *A Generative Variational Model for Inverse Problems in Imaging*, Master’s thesis, Karl-Franzens Universität Graz, Graz, 2020.
- [18] A. HABRING AND M. HOLLER, *Source Code to Reproduce the Results of “A Generative Variational Model for Inverse Problems in Imaging,”* https://github.com/habring/generative_variational_reg, 2021.
- [19] R. HECKEL AND P. HAND, *Deep Decoder: Concise Image Representations from Untrained Non-convolutional Networks*, <https://arxiv.org/abs/1810.03982>, 2019.
- [20] R. HECKEL AND M. SOLTANOLKOTABI, *Denoising and Regularization via Exploiting the Structural Bias of Convolutional Generators*, <https://arxiv.org/abs/1910.14634>, 2019.
- [21] M. HINTERMÜLLER, M. HOLLER, AND K. PAPAFITSOROS, *A function space framework for structural total variation regularization with applications in inverse problems*, Inverse Problems, 34 (2018), 064002, <https://doi.org/10.1088/1361-6420/aab586>.
- [22] B. HOFMANN, B. KALTENBACHER, C. PÖSCHL, AND O. SCHERZER, *A convergence rates result for Tikhonov regularization in Banach spaces with non-smooth operators*, Inverse Problems, 23 (2007), pp. 987–1010, <https://doi.org/10.1088/0266-5611/23/3/009>.
- [23] G. JAGATAP AND C. HEGDE, *Algorithmic Guarantees for Inverse Imaging with Untrained Network Priors*, <https://arxiv.org/abs/1906.08763>, 2019.
- [24] A. KLÖCKNER, N. PINTO, Y. LEE, B. CATANZARO, P. IVANOV, AND A. FASIH, *PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation*, Parallel Comput., 38 (2012), pp. 157–174, <https://doi.org/10.1016/j.parco.2011.09.001>.
- [25] M. LEBRUN, M. COLOM, A. BUADES, AND J.-M. MOREL, *Secrets of image denoising cuisine*, Acta Numer., 21 (2012), pp. 475–576, <https://doi.org/10.1017/S0962492912000062>.
- [26] V. LEMPITSKY, A. VEDALDI, AND D. ULYANOV, *Deep image prior*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454, <https://doi.org/10.1007/s11263-020-01303-4>.
- [27] V. LEMPITSKY, A. VEDALDI, AND D. ULYANOV, *Source Code for the Deep Image Prior*, <https://github.com/DmitryUlyanov/deep-image-prior>, 2018.
- [28] H. LI, J. SCHWAB, S. ANTHOLZER, AND M. HALTMEIER, *NETT: Solving Inverse Problems with Deep Neural Networks*, <https://arxiv.org/abs/1803.00092>, 2019.
- [29] S. LUNZ, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *Adversarial Regularizers in Inverse Problems*, <https://arxiv.org/abs/1805.11572>, 2019.
- [30] S. MALLAT, *A Wavelet Tour of Signal Processing*, Elsevier, Amsterdam, 2009.
- [31] D. OBMANN, J. SCHWAB, AND M. HALTMEIER, *Deep Synthesis Regularization of Inverse Problems*, <https://arxiv.org/abs/2002.00155>, 2020.
- [32] V. PAPYAN, Y. ROMANO, J. SULAM, AND M. ELAD, *Convolutional dictionary learning via local processing*, in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5296–5304.
- [33] T. POCK AND S. SABACH, *Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems*, SIAM J. Imaging Sci., 9 (2016), pp. 1756–1787, <https://doi.org/10.1137/16m1064064>.
- [34] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268, [https://doi.org/10.1016/0167-2789\(92\)90242-F](https://doi.org/10.1016/0167-2789(92)90242-F).
- [35] O. RUSSAKOVSKY, J. DENG, H. SU, J. KRAUSE, S. SATHEESH, S. MA, Z. HUANG, A. KARPATY, A. KHOSLA, M. BERNSTEIN, A. C. BERG, AND L. FEI-FEI, *ImageNet large scale visual recognition challenge*, Int. J. Comput. Vis., 115 (2015), pp. 211–252, <https://doi.org/10.1007/s11263-015-0816-y>.

- [36] O. SCHERZER, M. GRASMAIR, H. GROSSAUER, M. HALTMEIER, AND F. LENZEN, *Variational Methods in Imaging*, Appl. Math. Sci. 167, Springer, New York, 2008.
- [37] J. SULAM, V. PAPYAN, Y. ROMANO, AND M. ELAD, *Multilayer convolutional sparse modeling: Pursuit and dictionary learning*, IEEE Trans. Signal Process., 66 (2018), pp. 4090–4104, <https://doi.org/10.1109/TSP.2018.2846226>.
- [38] D. V. VEEN, A. JALAL, M. SOLTANOLKOTABI, E. PRICE, S. VISHWANATH, AND A. G. DIMAKIS, *Compressed Sensing with Deep Image Prior and Learned Regularization*, <https://arxiv.org/abs/1806.06438>, 2018.
- [39] J. WHANG, Q. LEI, AND A. G. DIMAKIS, *Compressed Sensing with Invertible Generative Models and Dependent Noise*, <https://arxiv.org/abs/2003.08089>, 2020.
- [40] H. ZHANG AND V. M. PATEL, *Convolutional sparse coding-based image decomposition*, in Proceedings of BMVC, 2016, <https://doi.org/10.5244/C.30.125>.