Deep Unfolding With Normalizing Flow Priors for Inverse Problems

Xinyi Wei, Hans van Gorp, Lizeth Gonzalez-Carabarin, Daniel Freedman, Yonina C. Eldar, *Fellow, IEEE*, and Ruud J. G. van Sloun, *Member, IEEE*

***Abstract*—Many application domains, spanning from computational photography to medical imaging, require recovery of high-fidelity images from noisy, incomplete or partial/compressed measurements. State-of-the-art methods for solving these inverse problems combine deep learning with iterative model-based solvers, a concept known as deep algorithm unfolding or unrolling. By combining a-priori knowledge of the forward measurement model with learned proximal image-to-image mappings based on deep networks, these methods yield solutions that are both physically feasible (data-consistent) and perceptually plausible (consistent with prior belief). However, current proximal mappings based on (predominantly convolutional) neural networks only implicitly learn such image priors. In this paper, we propose to make these image priors fully explicit by embedding deep generative models in the form of normalizing flows within the unfolded proximal gradient algorithm, and training the entire algorithm end-to-end for a given task. We demonstrate that the proposed method outperforms competitive baselines on various image recovery tasks, spanning from image denoising to inpainting and deblurring, effectively adapting the prior to the restoration task at hand.**

***Index Terms*—Deep unfolding, normalizing flows, inverse problem, image reconstruction.**

I. Introduction

IMAGE reconstruction from noisy, partial or limited-bandwidth measurements is an important problem with applications spanning from fast Magnetic Resonance Imaging (MRI) [1] to photography [2]. These reconstruction tasks can be posed as linear inverse problems, which are often ill-posed, with many potential solutions satisfying the measurements. Recovery of a meaningful and plausible solution thus requires adequate statistical priors. Formulating such priors for natural or medical image recovery tasks is however not trivial and often dependent on the recovery task itself.

Manuscript received September 13, 2021; revised March 4, 2022 and May 13, 2022; accepted May 17, 2022. Date of publication June 3, 2022; date of current version June 21, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Laurent Condat. *(Corresponding author: Xinyi Wei.)*

Xinyi Wei, Hans van Gorp, Lizeth Gonzalez-Carabarin, and Ruud J. G. van Sloun are with the Department of Electrical Engineering, Eindhoven University of Technology, 5612AZ Eindhoven, The Netherlands (e-mail: x.wei@student.tue.nl; h.v.gorp@tue.nl; l.gonzalez.carabarin@tue.nl; R.J.G.v.Sloun@tue.nl).

Daniel Freedman is with the Verily Research (formerly Google Life Sciences), Haifa 3190500, Israel (e-mail: danielfreedman@google.com).

Yonina C. Eldar is with the Department of Math and Computer Science, Weizmann Institute of Science, Rehovot 7610001, Israel (e-mail: yonina.eldar@weizmann.ac.il).

Digital Object Identifier 10.1109/TSP.2022.3179807

Traditional convex optimization methods for, e.g., compressed sensing assume sparsity in some transform domain [3], [4]. Choosing an appropriate sparse basis is highly dependent on the application and requires careful analysis of, for example, wavelet or total variation-based regularizers that are hard to tune in practice.

Deep learning [5] is increasingly adopted for image reconstruction, outperforming traditional iterative-based reconstruction methods for tasks such as image denoising [6]–[9], deconvolution [10], inpainting [11], [12] and end-to-end signal recovery [13]–[16]. More specifically, within the framework of compressed sensing, in which signals are to be reconstructed from a set of compressed measurements, deep learning methods have improved both image quality and reconstruction speed [17], [18].

Recent works have shown that, using variable splitting techniques [19], [20], any preferred denoiser can be used within classical model-based optimization methods. Typical denoising architectures are based on convolutional autoencoders, U-Nets [21], or residual networks (ResNets) [22]. An interesting special case is DRUNet, having the ability to handle various noise levels via a single model [23]. Related to this, deep generative models (DGMs), such as generative adversarial networks (GANs) [24], variational autoencoders (VAEs) [25] and normalizing flows [26], can also serve as meaningful priors for inverse problems in imaging [18], [27]–[30]. DGMs generate a complex distribution (e.g. that of natural images) from a simple latent base distribution, e.g. independent Gaussians, using a learned deterministic transformation, obtained by pre-training on a large dataset of clean images. This pre-trained model is subsequently used to solve inverse problems by performing gradient-based optimization in their (possibly lower dimensional) latent space.

While all of these approaches improve upon the hand-crafted sparsity-based priors and exhibit great empirical success, they do not accelerate the optimization process and still rely on time-consuming iterative algorithms. Moreover, their strength, being agnostic of the task and merely concerned with modeling the general image prior, is also a limitation: these approaches do not exploit task-specific statistical properties that can aid the optimization.

Deep algorithm unfolding aims to address these problems by unrolling the iterative optimization algorithm as a feed forward deep neural network [31]–[34]. The result is a deep network that takes the structure of the iterations in, for example, proximal-gradient methods, but allows for learning the parameters and/or

successive "neural" proximal mappings directly from training data [35]. Examples include ADMM-Net, an unfolded version of the iterative ADMM solver [36], ISTA-Net, integrating convolutional networks with sparse-coding-based soft thresholding activations [37], and CORONA, an unfolded robust PCA algorithm for clutter suppression in Ultrasound [38]. Other works include D-AMP [39], [40] inspired by the approximate message passing (AMP) algorithm, neural proximal gradient descent [35], and approaches that unfold primal-dual solvers [41], [42] or half-quadratic splitting methods [43], [44].

However, these fast and task-based neural unrolled proximal gradient descent methods no longer explicitly model the underlying statistical priors as a data-generating probability density function. Instead, current methods rely on "discriminative" network architectures to model the proximal mapping, such as Resnet- or U-Net-based architectures. Despite their success and vast application space, such implicit priors via neural mappings complicate analysis and steer away from a probabilistic interpretation and modeling of the data distribution.

In this paper, we propose an end-to-end deep algorithm unfolding framework that combines neural proximal gradient descent with generative normalizing flows priors. Our approach first pre-trains a generic flow-based model on natural images by direct likelihood maximization, and subsequently fine-tunes the entire pipeline and priors to adapt to specific image reconstruction tasks.

Our main contributions are:

- We propose a new framework for solving linear inverse problems based on deep algorithm unfolding and normalizing flows priors that adapt to the data and task.
- We leverage the generative probabilistic nature of our model to yield a strong initial guess: the maximum likelihood solution of the learned flow prior.
- We demonstrate strong performance gains over state-of-the-art neural proximal gradient descent baselines on a wide range of image restoration problems.

The remainder of this paper is organized as follows. In Section II, we introduce the problem and objective. Then, in Section III, we present our method by first describing generative flow priors in a generic optimization problem, and then proceeding by unrolling the iterations of a proximal gradient algorithm for that optimization problem. In Section IV we turn to describing the specific flow architecture we adopt, and continue by detailing the experimental setup in Section V. The results and specifics for each of the tasks are given in Section VI. Finally, in Section VII we discuss our results and conclude in Section VIII.

II. Problem Formulation

We consider problems of the following form:

$$\mathrm{y} = \mathbf{A}\mathrm{x} + \eta, \tag{1}$$

where $\mathrm{y} \in \mathbb{R}^m$ is a noisy measurement vector, $\mathrm{x} \in \mathbb{R}^n$ is the desired signal/image expressed in vector form, $\eta \in \mathbb{R}^m$ is a noise vector, and $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a measurement matrix, which we here assume to be known. Note that this can easily be generalized to scenarios where $\mathbf{A}$ becomes learnable in the unfolded structure.

For the ill-posed inverse problems that we are interested in, maximum-likelihood estimation, i.e., $\mathrm{argmax}_{\mathrm{x}} p(\mathrm{y}|\mathrm{x})$, is insufficient, yielding solutions that adhere to the measurement model but are visually implausible. By imposing statistical priors, meaningful solutions (i.e., those that fit expected behavior and prior knowledge) can be obtained through maximum a posteriori (MAP) inference:

$$\hat{\mathrm{x}}_{MAP} := \arg\max_{\mathrm{x}} p(\mathrm{x}|\mathrm{y}) \propto \arg\max_{\mathrm{x}} p(\mathrm{y}|\mathrm{x})\, p_\theta(\mathrm{x}), \tag{2}$$

where $p(\mathrm{y}|\mathrm{x})$ is the likelihood according to the linear noisy measurement model, and $p_\theta(\mathrm{x})$ is the signal prior.

The effectiveness of MAP inference strongly depends on the adequacy of the chosen prior. Formalizing such knowledge is challenging and requires careful analysis for each domain, for example natural images or medical images, and recovery task. Assuming a Gaussian distribution of measurement model errors, i.e. $p(\mathrm{y}|\mathrm{x}) \sim \mathcal{N}(\mu = \mathbf{A}\mathrm{x}, \sigma_n^2)$, MAP optimization leads to the following (negative log posterior) minimization problem:

$$\hat{\mathrm{x}} \in \arg\min_{\mathrm{x}} \frac{1}{2\sigma_n^2} \|\mathrm{y} - \mathbf{A}\mathrm{x}\|_2^2 - \log p_\theta(\mathrm{x}). \tag{3}$$

Given (3), our goal is now twofold: 1) to learn a useful prior $p_\theta(\mathrm{x})$, and 2) to accelerate and improve performance of standard gradient-based optimization of (3) using deep algorithm unfolding. We will address the former in the next section, and then proceed with the latter in Section III-B.

III. Deep Algorithm Unfolding With Flow Priors

A. Flow Priors

Normalizing flows [26] are a class of generative models, which are capable of modeling powerful and expressive priors. Normalizing flows transform a base probability distribution $p(\mathrm{z}) \sim \mathcal{N}(0, I)$ into a more complex, possibly multi-modal distribution by a series of composable, bijective, and differentiable mappings.

Due to the invertible nature of normalizing flows models, they can operate in both directions. These are the generative direction, which generates an image from a point in latent space ($\mathrm{x} = g_\theta(\mathrm{z})$), and the flow direction, which maps images into its latent representation ($\mathrm{z} = f_\theta(\mathrm{x})$). To create a normalizing flow model of sufficient capacity, many layers of bijective functions can be composed together:

$$\mathrm{z} = f_\theta(\mathrm{x}) = (\mathrm{f}_1 \circ \mathrm{f}_2 \circ \ldots \circ \mathrm{f}_\mathrm{i})(\mathrm{x}), \tag{4}$$

and,

$$\mathrm{x} = g_\theta(\mathrm{z}) = (\mathrm{f}_\mathrm{i}^{-1} \circ \mathrm{f}_{\mathrm{i}-1}^{-1} \circ \ldots \circ \mathrm{f}_1^{-1})(\mathrm{z}), \tag{5}$$

where '$\circ$' denotes the composition of two functions, and θ are the parameters of the model.

The functions $f_1, f_2, \ldots f_i$ are all layers of a neural network. Each of these layers is invertible and may have trainable parameters. Specifically, each layer is either an actnorm, squeeze, invertible convolution, or affine transformation. Through this lens, f_θ is in fact a neural network of which the inverse also exists. Examples of such networks include iRevNets, inverse autoregressive flow, and GLOW. We refer the readers to [45] for

a comprehensive overview. Here we choose to adopt GLOW: details of the specific normalizing flow model we adopt, and its parameterization, are given in Section IV.

Exact density evaluation of $p_\theta(\mathrm{x})$ is possible through the use of the change of variables formula, leading to:

$$\log p_\theta(\mathrm{x}) = \log p(\mathrm{z}) + \log \mid \mathtt{det} D f_\theta(\mathrm{x}) \mid, \tag{6}$$

where D is the Jacobian, accounting for the change in density caused by the transformation f_θ. With latent z following a normal distribution with zero mean and unit standard deviation, and $\mathrm{x} = g_\theta(\mathrm{z})$, we can then perform proximal updates in z space:

$$\begin{aligned} \hat{\mathrm{z}} &\in \arg\min_{\mathrm{z}} \frac{1}{2\sigma_n^2}\|\mathrm{y} - \mathbf{A}g_\theta(\mathrm{z})\|_2^2 - \log p(\mathrm{z}), \\ &\in \arg\min_{\mathrm{z}} \|\mathrm{y} - \mathbf{A}g_\theta(\mathrm{z})\|_2^2 + \lambda\|\mathrm{z}\|_2^2, \end{aligned} \tag{7}$$

where λ is a parameter that balances the importance of adhering to the measurements (data consistency) and the prior. Note that other simple base priors for z may be adopted as well. For instance, one may assume a Laplace distribution, recasting the ℓ_2 norm in (7) into an ℓ_1 norm.

In our case, we have formulated the problem as (7), finding the optimal latent code z, instead of finding the optimal reconstructions x, as in (3). This is useful as both previous works [30], [46] found it is empirically difficult to optimize in x-space. Because the statistics in z space are the same in all dimensions (standard Normal distribution) and semantically useful relationships are created by the normalizing flow network when going to and from the latent space. In our proximal gradient style algorithm we go back and forth between optimizing in x-space and in z-space. This implies that we chose to ignore the log-determinant term. However, our method works well without it, and including it increases the computational overhead and adds complexity.

B. Unrolled Proximal Gradient Iterations

The optimization problem in (7) can be solved using an iterative proximal-style algorithm that alternates between gradient updates in the direction of the data consistency term and pushing the solution in the proximity of the prior.

To derive our iterative scheme, we will alternate using the problem formulations in (3) and (7). More specifically, at every fold k the network performs data consistency updates using the x-space formulation in (3):

$$\tilde{\mathrm{x}}^{(k+1)} = \mathrm{x}^{(k)} - \mu^{(k)}\mathbf{A}^T(\mathrm{y} - \mathbf{A}\mathrm{x}^{(k)}) \tag{8}$$

where superscript (k) denotes the current fold and $\mu^{(k)}$ is the trainable step size. The signal representation is then converted to the latent space:

$$\tilde{z}^{(k+1)} = f_\theta^{(k+1)}(\tilde{\mathrm{x}}^{(k+1)}). \tag{9}$$

The purpose of this conversion is so that we may perform the proximal update $\mathcal{P}(\cdot)$ using the z-space formulation in (7):

$$z^{(k+1)} = \mathcal{P}^{(k+1)}(\tilde{\mathrm{z}}^{(k+1)}) = \frac{\tilde{\mathrm{z}}^{(k+1)}}{1+\lambda^{(k+1)}} \tag{10}$$

where $\lambda^{(k+1)}$ is a trainable shrinkage parameter. Intuitively, this can be understood as pushing solutions into a high likelihood regime (i.e. closer to the origin in z). Finally, we convert from latent space back to signal space

$$\mathrm{x}^{(k+1)} = g_\theta^{(k+1)}(\mathrm{z}^{(k+1)}) \tag{11}$$

and continue on to the next iteration.

We unfold the above iterative algorithm as a K-fold feedforward neural network that we train end-to-end. After K folds the final estimate $\hat{\mathrm{x}}$ is produced from the latent space after data consistency:

$$\hat{\mathrm{x}} = g_\theta^{(K)}(\tilde{\mathrm{z}}^{(K)}). \tag{12}$$

C. End-to-End Task-Adaptive Training and Initial Guess

We make use of a pre-trained single generative normalizing flow model from a set of clean images to learn a generic density function. After pre-training, we embed these generic priors into the unrolled architecture and tailor it to the specific recovery task at hand using end-to-end supervised learning. This yields an architecture in which each fold has a distinct and task-based flow model. By unrolling and training end-to-end we no longer guarantee nor explicitly promote normality of the latent space of the flow prox at each fold. We will further discuss this and its implications in Section VII.

We also make use of the explicit likelihood modeling of the normalizing flow prior to yield a useful initial guess for x: the maximum likelihood solution according to the clean images, by exploiting the fact that the most likely image is at $z = 0$. This serves as an input to our network and is denoted by $\hat{x}^{(0)} = g_\theta^{(0)}(z^{(0)} = 0)$; see also Fig. 1.

IV. Normalizing Flows Architecture

For the normalizing flow model we make use of GLOW [26], a normalizing flow architecture that uses 1×1 convolutions to permute the dimensions on a multi-scale architecture [48]. Central to GLOW is the affine transformation, which is defined as:

$$\mathrm{y} = s \cdot \mathrm{x} + t, \tag{13}$$

where s is the scale, and t is the translation that is applied to x. In general, each step of GLOW consists of three stages: actnorm, an invertible 1×1 convolution, and an affine coupling layer. The actnorm stage (short for activation normalization) is a normalization scheme that is better adapted to low batch sizes than conventional batch normalization. The actnorm stage is implemented as an affine transformation that acts upon the incoming data, with learnable scale and translation parameters.

After normalization an invertible 1×1 convolution is performed. This convolution can be viewed as a generalization of a permutation operation. Permutation of the dimensions is a vital step in normalizing flows to ensure that each dimension can affect every other dimension, enabling the model to transform complex (data) distributions into normal distributions. By learning this permutation as a convolution, rather than choosing it

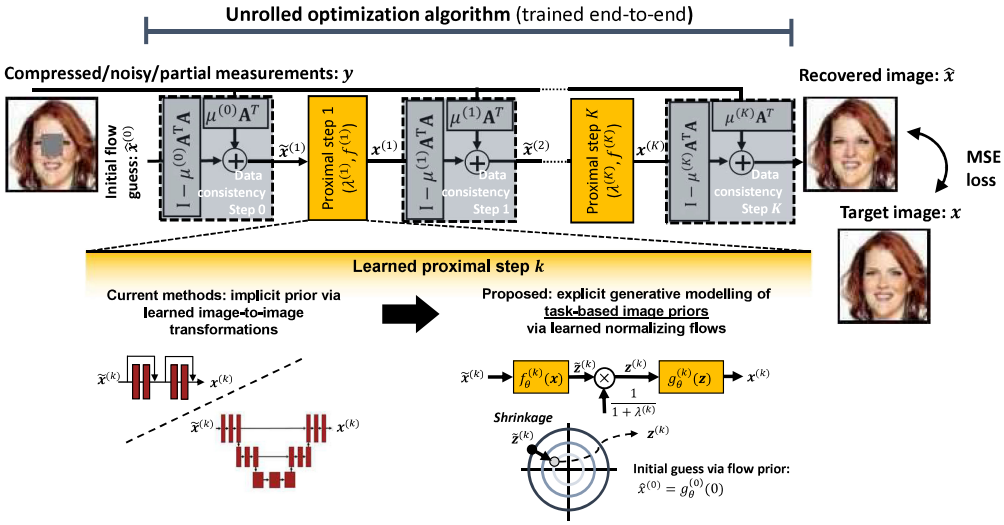

Fig. 1. Overview of the proposed algorithm. We unroll the iterations of a proximal gradient algorithm as a deep neural network. The proximal step performs shrinkage in the latent space of a normalizing flow prior, successively pushing the image likelihood in that distribution across the folds.

a-priori, the permutation can more powerfully adapt to the data and/or task.

Each step of GLOW is concluded with an affine coupling layer. The scale and translation parameters are created from part of the incoming tensor, using:

$$\begin{aligned}
\mathrm{x}_a, \mathrm{x}_b &= \mathrm{split}(\mathrm{x}) \\
(\log s, t) &= \mathrm{NN}(\mathrm{x}_b) \\
s &= \exp(\log s) \\
\mathrm{y}_a &= s \cdot \mathrm{x}_a + t \\
\mathrm{y}_b &= \mathrm{x}_b \\
\mathrm{y} &= \mathrm{concat}(\mathrm{y}_a, \mathrm{y}_b),
\end{aligned}$$

where NN is a neural network that is not required to be invertible. This is because its input is left unchanged by the affine coupling layer, it changes x_a into y_a via a scaling and transformation, but leaves $x_b = y_b$. Thus when going backwards through this layer, we will always know what input was used for this neural network to generate s and t. In order for the network to also affect the 'b' part of the input, the aforementioned 1×1 convolutions are used, which change which parts of the signal are a and which parts are b, thereby slowly allowing GLOW to whiten the entire signal.

All of these steps are combined within a multi-scale architecture [48] having 6 levels and a depth of 32. For details regarding its architecture and implementations (code) we refer the reader to the original paper by Kingma and Dhariwal [26] as well as the paper by Asim *et al.* [30].

V. Experimental Setup

We assess our method's performance for various image recovery tasks using both the in-distribution images as CelebA-HQ [49], and the out-of-distribution images as the Anime Faces set [50]. Across all experiments we used $K = 4$ folds of the unfolded proximal gradient algorithm. We implement the networks and the experiments using Pytorch.

A. Dataset

As in-distribution set we make use of the CelebA-HQ training set. CelebA-HQ consists of 23,000 training, 1,500 validation and 1,500 test images, which are cropped to the faces, and resized to a size of 64×64 pixels with 3 color channels.

As out-distribution set we make use of the Anime Faces set [50]. As this set was only used for out-of-distribution evaluation performance, no train-test split was made. Similar to the CelebA-HQ dataset the faces were detected and centered and the images were then cropped to a size of 64×64 pixels with 3 color channels.

B. Training Strategy

We make full use of the generative capabilities of our GLOW prox, by first pre-training it as a generative model, and consequently re-training it as the proximal operator in an unrolled proximal gradient scheme. The pre-trained GLOW architecture consists of a sequence of affine transformations with a depth of flow 18. The number of multi-scale levels is 4. The model is trained on maximizing the data-likelihood of clean images given by:

$$\log p_\theta(\mathcal{D}) = \sum_{i=1}^{N} \left[\log p_z(f_\theta(\mathrm{x}_i)) + \log \mid \mathtt{det} D f_\theta(\mathrm{x}_i) \mid\right]. \tag{14}$$

After pre-training, we untie the weights of the GLOW model at every fold and use it as the proximal operator. We then train the proximal gradient network again using the CelebA-HQ dataset, but this using corrupted-clean image pairs. We train the proximal gradient network for the specific image recovery tasks using a Mean Square Error (MSE) loss. We employ the Adam optimizer with (lr $= 1e^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and

TABLE I
EXPERIMENTAL RESULTS OF DEEP UNFOLDING WITH NORMALIZING FLOWS PRIORS COMPARED TO TWO STRONG BASELINES. VALUES REPORTED ARE MEAN PEAK SIGNAL TO NOISE RATIO (PSNR) OF THE RECONSTRUCTED IMAGES ACROSS THE TEST SET

Experiment (settings)	**Denoising** $\eta \sim \mathcal{N}(\mu_n = 0, \sigma_n = 0.2)$	**Inpainting** $W = 19 \times 19$	**Deblurring** $\sigma_b = 5$
ResNet Prox [35]	27.054 dB	34.010 dB	35.554 dB
Pre-trained ResNet prox *Pre-trained as a denoiser		33.216 dB	35.703 dB
U-Net Prox [47]	28.180 dB	35.157 dB	38.662 dB
Pre-trained U-Net prox *Pre-trained as a denoiser		35.337 dB	38.046 dB
Pre-trained GLOW prox (Proposed) *Pre-trained by data-likelihood maximization	**28.236** dB	**35.927** dB	**40.549** dB

$\epsilon = 1e-8$). Moreover, we train the learnable step size $\mu^{(k)}$, and shrinkage factor $\lambda^{(k)}$ with a higher learning rate, namely $1e^{-2}$. As before, we perform early stopping based on the validation loss.

Leveraging the invertible nature of the GLOW model, we strongly reduce train-time memory of the full unfolded architecture using the approach by Putzky and Welling [51]; =instead of storing all intermediate activations for back-propagation, we recalculate the gradients given the post-activations during backpropagation.

C. Baselines

We compare our generative flow-based priors, 25.78 M trainable parameters to two alternative neural proximal mappings; one based on ResNets [35], with 0.16 M trainable parameters and one based on a U-Net with 31.04 M trainable parameters.

Note that for fair and direct comparison, we focus on typical alternatives *within* the unfolded proximal gradient framework. This allows for straightforward assessment of the merit of the proposed (task-adapted) normalizing flow prior, beyond the architectural advantages of unfolding the proximal gradient algorithm itself. Training settings and strategy are identical to those described in Section V-B.

Our ResNet proximal baseline follows the structure proposed by Mardani *et al.*, [35]. Each residual block consists of two convolutional layers (3×3 kernel and 128 feature maps), followed by batch normalization and ReLU activation. This is then followed by another 3 convolutional layers with 1×1 kernels. Mardani *et al.* found that using 2 such residual blocks per proximal fold works best, so we follow that here as well. Our second baseline proximal mapping is a standard U-Net [47] implementation. The U-Net is a fully convolutional neural network that consists of a contracting path and an expansive path with skip connections between the two. This allows for learning both low-level and high-level features. We here make use of the Pytorch U-Net implementation.

VI. RESULTS

A. Experiment on In-Distribution Test Images

1) Denoising: We consider noisy measurements $\mathrm{y} = \mathrm{x} + \eta$, where η is an i.i.d. Gaussian noise vector with standard deviation σ_n and mean $\mu_n = 0$. Note that $\mathbf{A}$ in (8) is thus an identity matrix here. We train the networks on denoising level of $\sigma_n = 0.2$ and analyze performance accordingly. The goal of the recovery algorithm is to denoise the image, recovering x from y.

Table I shows that the proposed GLOW prox outperforms the baselines. Qualitatively, the examples displayed in Fig. 2 (top row) show that both the GLOW prox and U-Net prox well preserve the desired features (e.g. the wrinkles around the cheeks).

2) Inpainting: We perform measurements $\mathrm{y} = \mathbf{A}\mathrm{x}$, where $\mathbf{A}$ is a matrix masking the center $W = 19 \times 19$ elements of an image by operating on its vectorized form x. The goal of the recovery algorithm is to "inpaint" the masked pixels as accurately as possible.

For inpainting, the performance gain using GLOW prox is about 0.6 dB (see Table I). Visual inspection of the second inpainting example given in the second row of Fig. 2 shows that the proposed method is better capable of producing high-resolution reconstructions. Note that this apparent higher fidelity is paired with a pixel-wise PSNR improvement: reconstructions are not merely visually pleasing, but also more accurate. We again refer to the challenging examples (row 2): while U-Net prox also can reconstruct the actual shape of the nose, the proposed method does produce improved skin tone compared to the surroundings of the "inpainted" area.

3) Deblurring: We take measurements $\mathrm{y} = \mathbf{A}\mathrm{x}$, where $\mathbf{A}$ is a 2D convolution matrix blurring the image with a Gaussian kernel having standard deviation $\sigma_b = 5$ pixels. We analyze the performance of our method in recovering the original image from the blurred measurements, i.e. deblurring.

As for the other tasks, the proposed GLOW prox outperforms the other baselines (see Table I). While the performance gain is about 1.8 dB PSNR with respect to the ResNet prox and the U-Net prox, minor differences could be visually inspected from Fig. 2 (bottom row).

4) Impact of Pre-Training: We also pre-train the baselines as the denoisers for a fair comparison and evaluate their performances accordingly. Specifically, we first pre-train the baselines in a denoising way using the CelebA-HQ training set. We then embed the denoisers in the unfolded network to train on the image inpainting and deblurring tasks, respectively. Table I indicates that U-Net prox does slightly benefit from such pre-training on the inpainting task, and the pre-trained ResNet prox also has a performance gain on the deblurring task. However, they still do not outperform our proposed method.

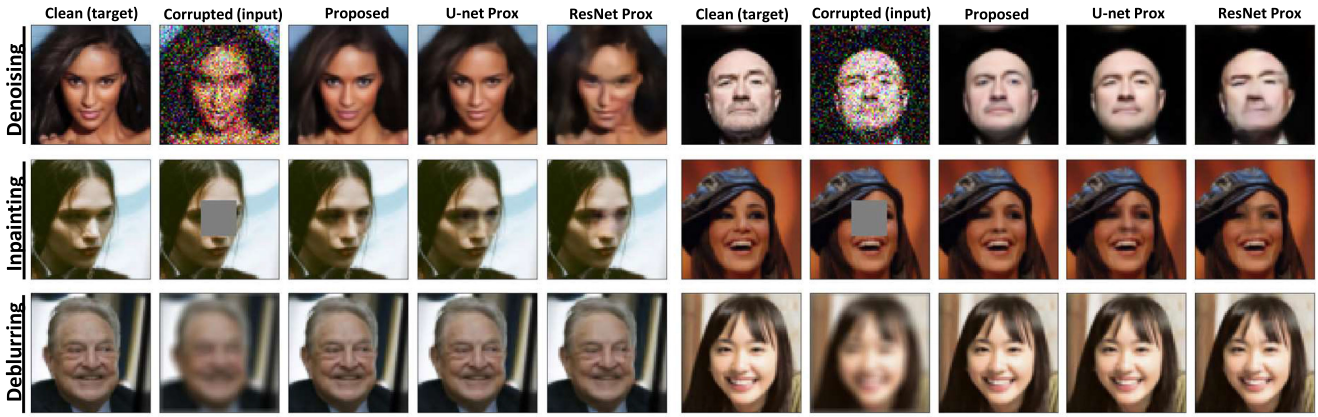

Fig. 2. Comparison of the proposed normalizing flows proximal mapping with baselines based on standard U-Net or ResNet proximal mappings across the in-distribution dataset. Results are shown for 3 image restoration tasks applied to 6 typical in-distribution example images.

TABLE II
RESULTS OF DEEP UNFOLDING WITH NORMALIZING FLOWS PRIORS COMPARED BETWEEN WITH PROPOSED FRAMEWORK AND THE PLUG-AND-PLAY. VALUES REPORTED ARE MEAN PEAK SIGNAL TO NOISE RATIO (PSNR) OF THE RECONSTRUCTED IMAGES ACROSS THE IN-DISTRIBUTION TEST SET

Experiment (settings)	**GLOW Prox** (Plug-and-play)	**GLOW Prox** (Proposed)
Denoising ($\eta \sim \mathcal{N}(\mu_n = 0, \sigma_n = 0.2)$)	8.112 dB	**28.236** dB
Inpainting ($W = 19 \times 19$)	18.960 dB	**35.927** dB
Deblurring ($\sigma_b = 5$)	23.141 dB	**40.549** dB

5) Comparison to the Plug-and-Play Approach: The plug-and-play approach is also a valuable baseline for evaluation. Specifically, we plug the pre-trained GLOW into the unfolded framework and only adapt the step sizes to the specific task in the train time. We observe a significant performance gain (see Table II) of our proposed method to the plug-and-play approach in all the tasks, further confirming the importance of adapting GLOW at each fold to specific tasks.

B. Experiment on Out-of-Distribution Measurement

We again test our proposed network and the baselines on the denoising, inpainting, and deblurring tasks using the CelebA-HQ test set, while with out-of-distribution measurement matrix $\mathbf{A}$ or noise level η. Consequently, for denoising, we test on the noise level η following a standard deviation $\sigma_n = 0.25$ and mean $\mu_n = 0$. Next, we test the images with the data consistency matrix $\mathbf{A}$ at the last fold blocking the center $W = 16 \times 16$ elements for inpainting. Finally, we test the images blurred with a Gaussian kernel $\mathbf{A}$ with a standard deviation $\sigma_b = 8$ pixels for deblurring.

Table III indicates that our GLOW prox achieves performance gains in all the cases. The visual examples in the first row of Fig. 3 strongly confirm that the reconstructions of our method are much sharper and more capable of preserving details (such as hairs). The second row of Fig. 3 again proves that the reconstructions of our method are more accurate, given the well-matched skin tone and shape of the front/side noses of the masked area.

C. Experiment on Out-Of Distribution Test Images

Finally, we evaluate the behaviour of the proposed method on out-of-distribution data x. To that end, we test our models trained on the CelebA-HQ dataset on images from the Anime Faces set [50]. The images in this out-of-distribution set show clear differences with respect to the train set, such as much bigger eyes and much smaller noses. We evaluate performance on 100 images that were randomly selected from the full dataset.

Table IV displays that performance deteriorates for out-of-distribution predictions. U-Net prox outperforms our model by around 0.5 dB in both the image denoising and inpainting tasks, and our model achieves about 1 dB gain to the baselines in the deblurring task. Example images for visual assessment are given in Fig. 4. Interestingly, visual comparison qualitatively shows a stronger "CelebA" image prior in the reconstructions by the proposed model compared to the reconstructions of the baselines. This is particularly evident form the inpainting task, where natural noses and eyes are painted into the unnatural Anime faces. In this case the U-Net and the ResNet-based reconstructions are more smooth and blurry, with a less clear "CelebA fingerprint".

VII. DISCUSSION

We demonstrated the potential for invertible generative models as proximal operators in unrolled architectures. In our experiments, GLOW-based proximal operators outperform several non-generative baselines, and since they learn the actual underlying structure and Probability Density Function (PDF) explicitly, they allow the user to "probe" (i.e., sample from) this PDF. Normalizing flows are particularly suited, as they allow for exact likelihood calculation, contrary to for example GANs. Note that in this work we have chosen to use the GLOW model, however, in principle any invertible flow model can be used with our method.

Asim *et al.* [30] demonstrate that GLOW models enable solving inverse problems by gradient-based optimization in its latent (normal) space. In contrast, we here (1) use GLOW priors in a proximal gradient descent algorithm, performing iterative

TABLE III
EXPERIMENTAL RESULTS OF DEEP UNFOLDING WITH NORMALIZING FLOWS PRIORS COMPARED TO TWO STRONG BASELINES ON THE CELEBA-HQ IMAGES WITH OUT-OF-DISTRIBUTION MEASUREMENTS. VALUES REPORTED ARE MEAN PEAK SIGNAL TO NOISE RATIO (PSNR) OF THE RECONSTRUCTED IMAGES ACROSS THE TEST SET

Experiment (settings)	**ResNet Prox** [35]	**U-Net Prox** [47]	**GLOW Prox** (Proposed)
Denoising ($\eta \sim \mathcal{N}(\mu_n = 0,\, \sigma_n = 0.25)$)	25.489 dB	25.770 dB	**26.633** dB
Inpainting ($W = 16 \times 16$)	34.644 dB	35.917 dB	**36.634** dB
Deblurring ($\sigma_b = 8$)	35.358 dB	38.034dB	**39.461** dB

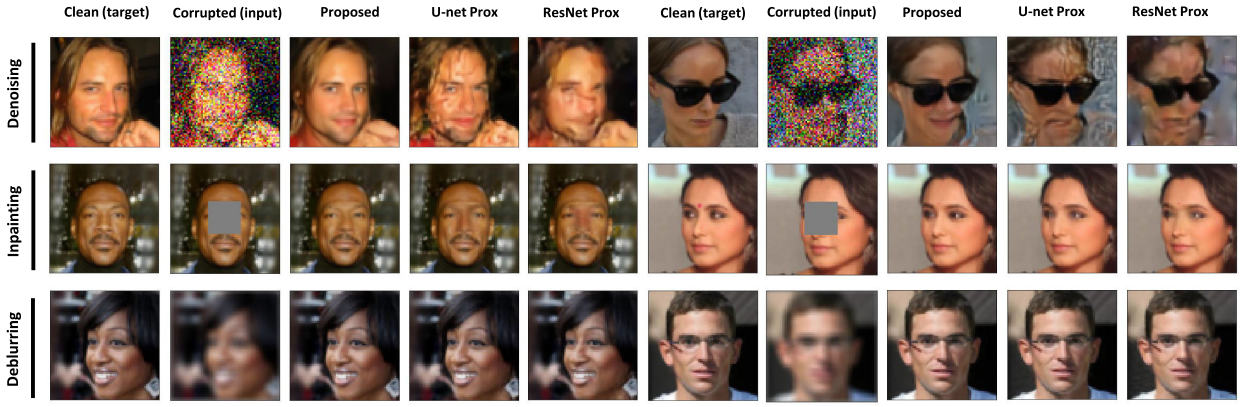

Fig. 3. Comparison of the proposed normalizing flows proximal mapping with baselines based on standard U-Net or ResNet proximal mappings on the CelebA-HQ images with out-of-distribution measurements. Results are shown for the denoising, inpainting and deblurring tasks applied to 6 typical CelebA-HQ example images.

TABLE IV
EXPERIMENTAL RESULTS OF DEEP UNFOLDING WITH NORMALIZING FLOWS PRIORS COMPARED TO TWO BASELINES. VALUES REPORTED ARE MEAN PEAK SIGNAL TO NOISE RATIO (PSNR) OF THE RECONSTRUCTED IMAGES ACROSS AN OUT-OF-DISTRIBUTION TEST SET

Experiment (settings)	**ResNet Prox** [35]	**U-Net Prox** [47]	**GLOW Prox** (Proposed)
Denoising ($\eta \sim \mathcal{N}(\mu_n = 0,\, \sigma_n = 0.2)$)	21.746 dB	**22.587** dB	22.102 dB
Inpainting ($W = 19 \times 19$)	23.705 dB	**24.015** dB	23.472 dB
Deblurring ($\sigma_b = 5$)	25.007 dB	26.224 dB	**27.419** dB

shrinkage in its normal space, and taking gradient steps towards the data-consistency term, and then (2) unfold several iterations of this algorithm to fit a fixed-complexity model to the target data distribution. Consequently, our proposed method yields improvements in the number of iterations (or folds) to acquire a feasible and plausible solution. For example we only need four folds for the inpainting task, while the method of [30] needs 50 iterations.

Compared to disjointly trained generative models used in e.g. the plug-and-play framework, unrolling the network and performing end-to-end training allows the learned PDF's to adapt to a given task. In our experiments on inpainting, we observed that directly using the pre-trained GLOW model (without any retraining to adapt to the task), yielded significantly worse results. This indicates that task-based adaptation of the PDF in an unrolled scheme dramatically improves performance.

However, once we perform unrolled training, thereby adapting the PDF, we no longer promote normality for each of the latent spaces of the different flow models. While this theoretically prohibits sampling from the PDFs to probe their behaviour, we in practice noticed that naive sampling of latents (using a Gaussian) yielded structured outputs nonetheless. In future work we consider promoting such normality after each shrinkage step by including a maximum likelihood loss (i.e. (14)) in addition to the final reconstruction loss during unrolled training.

A critical component of unrolling is the untying of the weights over the iterations. Our choice to untie the weights follows from literature, such as, Hershey *et al.* [52], who claim that untying the model parameters across layers in an unfolded framework helps create a more robust network. Because through untying the model parameters, we increase the network flexibility to learn a specific task while at the same time keeping the advantages of the original unfolded algorithm as being interpretable and stable. Many following works have also chosen to do so, see e.g. [53] and [54]. Our own initial experiments back this up, and we observed worse performance when not untying the weights of the model.

A downside of normalizing flows is the amount of flows that need to be stacked in order to model sufficiently rich distributions, and the associated memory footprint. In order to train the unrolled architecture with multiple GLOW models

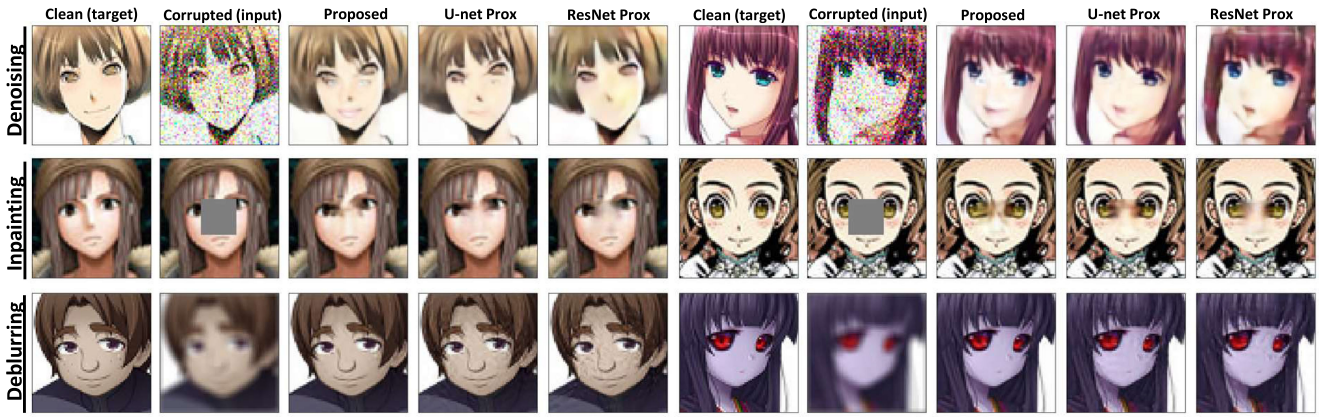

Fig. 4. Comparison of the proposed normalizing flows proximal mapping with baselines based on standard U-Net or ResNet proximal mappings. Results are shown for the denoising, inpainting and deblurring tasks applied to 6 typical out-of-distribution example images.

end-to-end under such memory footprints, we leverage their invertible nature and reduce the memory complexity to $O(1)$ following the work of Putzky and Welling [51]. Nevertheless, the computational complexity is untouched by this approach, and normalizing flows still require more computations than the convolutional baselines we compared to in this work.

In this work, we only experimented on fairly uni-modal datasets (Celeb-A and animefaces), that is, all the examples in the dataset are centered and cropped faces. A rich multi-modal dataset of e.g. natural images might present additional challenges for our current approach: the flows would need to map all the modes of the data distribution into a single Gaussian distribution. However, because the mapping needs to be invertible, different modes will inevitably be 'connected' (i.e. there will be mass between them). It is an open question how and if the shrinkage step we perform would be able to deal with this. For example, if the shrinkage step is too large, we might move from the region of one mode in latent space to that of another mode.

We here focused on a particular type of optimizer. One may also explore unfolding of other iterative optimizers, such as primal dual proximal methods, in combination with generative GLOW models. We may also consider expanding the application space to different types of data and problems including medical imaging (e.g. MRI, CT, ultrasound), time-series (ECG, audio, wireless communication), graphs, point clouds, and compressed sensing.

VIII. Conclusion

We propose a framework for deep algorithm unfolding based on task-adapted normalizing flows priors. Our method first learns generic priors on a given training dataset, and then adapts these to the specific image restoration task at hand. We evaluate the performance of our approach for a variety of such tasks, i.e., image denoising, inpainting and deblurring and for different scenarios, i.e., out-of-distribution measurements. We demonstrate performance gains compared to strong baselines. While unfolding and end-to-end training enables fitting to (and exploiting) a specific data distribution, it also makes it more sensitive to out of distribution measurements. We show that generative flow proximal operators suffer less from this problem than standard discriminative U-Net or ResNet ones, and thus have advantages in real world applications of unfolding. Beyond image restoration, we expect our method to find applications in compressed sensing and medical image reconstruction, which is part of future work.

References

[1] N. Pezzotti *et al.*, "An adaptive intelligence algorithm for undersampled knee mri reconstruction," *IEEE Access*, vol. 8, pp. 204 825–204 838, 2020.
[2] F. D. M. Neto and A. J. da Silva Neto, *An Introduction to Inverse Problems With Applications*. Berlin, Heidelberg, Germany: Springer, 2012.
[3] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[4] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[6] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D ?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2392–2399.
[7] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Netw.*, vol. 131, pp. 251–275, 2020.
[8] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.
[9] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, vol. 1, pp. 341–349.
[10] L. Xu, J. S. J. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, vol. 1, pp. 1790–1798.
[11] U. Demir and G. B. Ünal, "Patch-based image inpainting with generative adversarial networks," 2018, *arXiv:1803.07422*.
[12] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edgeconnect: Generative image inpainting with adversarial edge learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 3265–3274.
[13] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "ReconNet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 449–458.
[14] D. M. Pelt and K. J. Batenburg, "Fast tomographic reconstruction from limited data using artificial neural networks," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5238–5251, Dec. 2013.
[15] D. Boublil, M. Elad, J. Shtok, and M. Zibulevsky, "Spatially-adaptive reconstruction in computed tomography using neural networks," *IEEE Trans. Med. Imag.*, vol. 34, no. 7, pp. 1474–1485, Jul. 2015.
[16] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017.

[17] Y. Fang, L. Liu, P. Haipeng, J. Kurths, and Y. Yang, "Overview of compressed sensing: Sensing model, reconstruction algorithm, and its applications," *Appl. Sci.*, vol. 10, 2020, Art. no. 5909.

[18] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 537–546.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[20] R. Deriche, P. Kornprobst, and M. Nikolova, "Half-quadratic regularization for MRI image restoration," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2003, pp. VI–585.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv.*, 2015, vol. 9351, pp. 234–241.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[23] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *CoRR*, 2020, *arXiv:2008.13751*.

[24] I. Goodfellow *et al.*, "Generative adversarial nets," in *Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., Red Hook, NY, USA: Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[25] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. 2nd Int. Conf. Learn. Representations*, Banff, AB, Canada, Apr. 14–16, 2014.

[26] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in *Adv. in Neural Inf. Process. Syst.*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and N. Garnett, Eds., Red Hook, NY, USA: Curran Associates, Inc., 2018, vol. 31. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf

[27] P. Hand, O. Leong, and V. Voroninski, "Phase retrieval under a generative prior," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, pp. 9154–9164, 2018.

[28] P. Hand and V. Voroninski, "Global guarantees for enforcing deep generative priors by empirical risk," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 401–418, Jan. 2020.

[29] M. Asim, F. Shamshad, and A. Ahmed, "Solving bilinear inverse problems using deep generative priors," 2018, *arXiv:1802.04073*.

[30] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand, "Invertible generative models for inverse problems: Mitigating representation error and dataset bias," in *Proc. 37th Int. Conf. Mach. Learn.*, pp. 4577–4587, 2020.

[31] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[32] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[33] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and interpretable deep blind image deblurring via algorithm unrolling," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 666–681, 2020.

[34] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein, "Unrolled optimization with deep priors," 2017, *arXiv:1705.08041*.

[35] M. Mardani *et al.*, "Neural proximal gradient descent for compressive imaging," in *Proc. NeurIPS*, 2018.

[36] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, vol. 29, pp. 10–18.

[37] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1828–1837.

[38] O. Solomon *et al.*, "Deep unfolded robust PCA with application to clutter suppression in ultrasound," *IEEE Trans. Med. Imag.*, vol. 39, no. 4, pp. 1051–1063, Apr. 2020.

[39] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.

[40] C. A. Metzler, A. Mousavi, and R. G. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 1770–1781.

[41] S. Wang, S. Fidler, and R. Urtasun, "Proximal deep structured models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29, pp. 865–873.

[42] J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1322–1332, Jun. 2018.

[43] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, pp. 2305–2318, 2019.

[44] K. Zhang, L. Van Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3214–3223.

[45] I. Kobyzev, S. J. D. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3964–3979, Nov. 2021.

[46] J. Whang, Q. Lei, and A. Dimakis, "Solving inverse problems with a flow-based noise model," in *Proc. of the 38th Int. Conf. Mach. Learn., Ser. Proc. Mach. Learn. Res.*, M. Meila and T. Zhang, Eds., Jul. 18–24, 2021, vol. 139, pp. 11 146–11 157. [Online]. Available: https://proceedings.mlr.press/v139/whang21a.html

[47] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.

[48] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, Apr. 24–26, 2017. [Online]. Available: https://openreview.net/forum?id=HkpbnH9lx

[49] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Representations (Virtual)*, Feb. 2018.

[50] S. Rakshit, "Anime faces," 2020, Accessed: Jul. 29, 2021. [Online]. Available: http://www.kaggle.com/soumikrakshit/anime-faces

[51] P. Putzky and M. Welling, "Invert to learn to invert," in *Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Red Hook, NY, USA: Curran Associates, Inc., 2019, vol. 32, pp. 444–454.

[52] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," 2014, *arXiv:1409.2574*.

[53] J.-T. Chien and C.-H. Lee, "Deep unfolding for topic models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 318–331, Feb. 2018.

[54] C. Bertocchi, E. Chouzenoux, M.-C. Corbineau, J.-C. Pesquet, and M. Prato, "Deep unfolding of a proximal interior point method for image restoration," *Inverse Problems*, vol. 36, no. 3, 2020, Art. no. 034005.

Xinyi Wei received the B.S. degree from Shanghai Maritime University, Shanghai, China, and the M.S. degree in electrical engineering from the Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands, where she is currently working toward the Ph.D. degree with TU/e. Her research interests include automotive radar interference mitigation, and deep learning.

Hans van Gorp received the B.Sc. and M.Sc. degrees in electrical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2018, 2020, respectively, where he is currently working towards the Ph.D. degree. His research interests include generative modeling, deep unfolding, and uncertainty analysis for sleep research.

Lizeth Gonzalez-Carabarin received the Ph.D. degree from Hokkaido University, Sapporo, Japan in 2015. She was a postdoctoral with the Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, in 2018, followed by 2-years Marie-Curie fellowship with the Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands. Her research interests include Deep Learning models for hardware, and Deep Learning for biomedical applications.

Daniel Freedman received the A.B. degree in physics from Princeton University, Princeton, NJ, USA, in 1993, and the Ph.D. degree in engineering sciences from Harvard University, Cambridge, MA, USA, in 2000. From 2000 to 2009, he was a Professor of computer science with the Rensselaer Polytechnic Institute, Troy, NY, USA. In 2007, he became a Fulbright Fellow and a Visiting Professor with the Weizmann Institute of Science, Rehovot, Israel. Since 2009, he has been a number of corporate research positions with HP Labs, IBM Research, Microsoft Research, and Google Research. He is currently with Verily (formerly Google Life Sciences), where he leads Verily Research. He works in the fields of computer vision, novel imaging modalities, and AI for medical applications. In addition to the Fulbright Fellowship, he was the recipient of the National Science Foundation CAREER Award.

Yonina C. Eldar (Fellow, IEEE) received the B.Sc. degree in physics and electrical engineering from Tel-Aviv University (TAU), Tel-Aviv, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2002. She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. She was a Professor with the Department of Electrical Engineering, aTechnion. She is also a Visiting Professor with MIT, a Visiting Scientist with Broad Institute, and an Adjunct Professor with Duke University, Durham, NC, USA, and was a Visiting Professor at Stanford. She is author of the book *Sampling Theory: Beyond Bandlimited Systems* and coauthor of five other books published by Cambridge University Press. Her research interests include statistical signal processing, sampling theory and compressed sensing, learning and optimization methods, and their applications to biology, medical imaging, and optics.

Dr. Eldar was the recipient of many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award (2013), IEEE/AESS Fred Nathanson Memorial Radar Award (2014), and IEEE Kiyo Tomiyasu Award (2016). She was a Horev Fellow of the Leaders in Science and Technology program at the Technion and an Alon Fellow. She is a Member of the Israel Academy of Sciences and Humanities (elected 2017), and a EURASIP Fellow. She was also the recipient of the Michael Bruno Memorial Award from the Rothschild Foundation, Weizmann Prize for Exact Sciences, Wolf Foundation Krill Prize for Excellence in Scientific Research, Henry Taub Prize for Excellence in Research (twice), Hershel Rich Innovation Award (three times), Award for Women with Distinguished Contributions, Andre and Bella Meyer Lectureship, Career Development Chair at the Technion, Muriel & David Jacknow Award for Excellence in Teaching, and Technion's Award for Excellence in Teaching (two times). She was the recipient of several best paper awards and best demo awards together with her research students and colleagues including the SIAM outstanding Paper Prize, UFFC Outstanding Paper Award, Signal Processing Society Best Paper Award and IET Circuits, Devices and Systems Premium Award, was selected as one of the 50 most influential women in Israel and in Asia, and is a highly cited researcher.

She was a Member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education. She is the Editor in Chief of *Foundations and Trends in Signal Processing*, a Member of the IEEE Sensor Array and Multichannel Technical Committee and serves on several other IEEE committees. In the past, she was a Signal Processing Society Distinguished Lecturer, Member of the IEEE SIGNAL PROCESSING THEORY AND METHODS and Bio Imaging Signal Processing technical committees, and was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, *EURASIP Journal of Signal Processing*, *SIAM Journal on Matrix Analysis and Applications*, and *SIAM Journal on Imaging Sciences*. She was the co-chair and technical co-chair of several international conferences and workshops.

Ruud J. G. van Sloun (Member, IEEE) received the B.Sc. and M.Sc. degrees (cum laude) in electrical engineering and the Ph.D. degree (cum laude) from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2012, 2014, and 2018, respectively. Since then, he has been an Assistant Professor with the Department of Electrical Engineering, Eindhoven University of Technology and since January 2020 a Kickstart-AI Fellow with Philips Research, Eindhoven. From 2019 to 2020, he was also a Visiting Professor with the Department of Mathematics and Computer Science with the Weizmann Institute of Science, Rehovot, Israel. His research interests include artificial intelligence and deep learning for signal processing and imaging, model-based deep learning, compressed sensing, ultrasound imaging, and probabilistic signal, and image analysis. He is an NWO Rubicon laureate and was the recipient of the Google Faculty Research Award in 2020.