PatchNR: Learning from Small Data by Patch Normalizing Flow Regularization

Abstract

Learning neural networks using only a small amount of data is an important ongoing research topic with tremendous potential for applications. In this paper, we introduce a regularizer for the variational modeling of inverse problems in imaging based on normalizing flows. Our regularizer, called patchNR, involves a normalizing flow learned on patches of very few images. In particular, the training is independent from the considered inverse problem such that the same regularizer can be used for different forward operators acting on the same class of images. By investigating the distribution of patches versus those of the whole image class, we prove that our variational model is indeed a MAP approach. Our model can be generalized to conditional patchNRs, if additional supervised information is available. Numerical examples for superresolution of material images and low-dose or limited-angle computed tomography (CT) demonstrate that our method provides high quality results among methods with similar assumptions, but requires only few data.

1 Introduction

Solving inverse problems with limited access to training data is an active field of research. In inverse problems, we aim to reconstruct an unknown ground truth image \( x \) from some observation

\[
y = \text{noisy}(f(x)),
\]

where \( f \) is an ill-posed forward operator and “noisy” describes some noise model. In the recent years learning-based reconstruction methods as supervisely trained neural networks on large datasets \([37, 70]\) and conditional generative models \([6, 27, 59, 76]\) attained a lot of attention. However, for certain applications like medical or material imaging, the acquisition of a large database of paired ground truth images and observations is very costly or even impossible \([48, 88]\). As a remedy, many recent methods assume that the forward model is known and use it to reduce the required amount of data or to avoid the use of paired ground truth images and corresponding observations completely. In literature, such methods are often called “unsupervised” \([4, 57, 60, 78]\) even though the knowledge of the forward operator and a ground truth image allows the generation of a corresponding measurement. However, as the term “unsupervised” is used in many different variants in the literature, we will refer to such methods as “model-based”. Many of them are based on minimizing a variational functional...
of the form
\[
J(x; y) = D(f(x), y) + \lambda R(x), \quad \lambda > 0, \tag{2}
\]
where \(D\) is a data-fidelity term which depends on the noise model and measures how well the reconstruction fits to the observation and \(R\) is a regularizer which copes with the ill-posedness and incorporates prior information. Over the last years, learned regularizers like the total deep variation \([46, 47]\) or adversarial regularizers \([57, 65]\) as well as extensions of plug-and-play methods \([24, 79, 83]\) with learned denoisers \([31, 34, 69, 89]\) showed promising results, see \([8, 62]\) for an overview.

Furthermore, many papers leveraged the tractability of the likelihood of normalizing flows to learn a prior \([9, 29, 86, 87]\). They utilize the invertibility to optimize over the range of the flow together with the Gaussian assumption on the latent space. Also, diffusion models \([39, 40, 77, 78]\) have shown great generative modelling capabilities and have been used as a prior for inverse problems. Moreover, other generative models, such as GANs \([26, 63]\) or VAEs \([44]\), have been used as a regularizer, see the recent review \([20]\) and references therein. However, even if these methods allow an unsupervised reconstruction, their training is often computationally costly and a huge amount of training images is required.

One possibility to reduce the training effort consists in using only small image patches. Denoising methods based on the comparison of similar patches provided state-of-the-art methods \([12, 15, 51, 52]\) for a long time. Recently, the approximation of patch distributions of images was successfully exploited in certain papers \([3, 17, 30, 32, 33, 73]\). In particular, the authors of \([91]\) proposed the negative log likelihood of all patches of an image as a regularizer, where the underlying patch distribution was assumed to follow a Gaussian mixture model (GMM) which parameters were learned from few clean images. This method is still competitive to many approaches based on deep learning and several extensions were suggested recently \([22, 64, 74]\). However, even though GMMs can approximate any probability density function if the number of components is large enough, they suffer from limited flexibility in case of a fixed number of components, see \([23]\) and the references therein. Moreover the subsequent reconstruction procedure detailed in \([91]\) is computationally expensive.

In this paper, we propose to use a regularizer which incorporates a normalizing flow (NF) learned on image patches. NFs were introduced in \([19, 68]\), see also \([72]\) for its continuous counterpart. NFs build upon invertible neural networks and allow for explicit density evaluation.

Our PatchNR consists of a NF which is trained to approximate the distribution of patches. As the structure of small patches is usually much simpler than those of whole images, it appears that their approximation is more accurate. Moreover, as a large data base of patches can be extracted from few images, we require only a very small amount of training images. Once the patchNR is learned, we use the negative log likelihood of all patches as regularizer in \((2)\) for its reconstruction. Indeed we will prove that \((2)\) can be obtained by a MAP approach. Since the regularizer is only specific to the considered image class, but not to the inverse problem, it can be applied for different forward operators as, e.g., for low-dose CT and limited-angle CT without additional training. This is in contrast to supervised methods as FBP+UNet, where the network has to be trained for each new forward operator separately. Further, we generalize our approach to conditional patchNRs such that additional information or supervised training data can be incorporated. Note that the training with very few supervised images was also considered in few-shot learning, see \([84]\) for an overview.

We demonstrate by numerical examples that our patchNR admits high quality results for superresolution, deconvolution, low-dose and limited-angle CT and it is even close to results in supervised learning. The paper is organized as follows: We start by explaining the training of our patchNR and the reconstruction in Section 2. In Section 3 the conditional model is introduced. Our approach is integrated into the MAP framework in Section 4, i.e., the patchNR defines a probability density on the and deconvolution. Finally, conclusions are drawn in Section 6.

## 2 Patch Normalizing Flows in Variational Modeling

In the following, we assume that we are given a small number of high quality example images \(x_1, \ldots, x_M \in \mathbb{R}^{d_1 \times d_2}\). Indeed \(M = 1\) or \(M = 6\) in our numerical examples - from a certain image class as CT, material or texture images. Our method consists of two steps, namely i) learning a NF for approximating the patch distribution of the example images, and ii) establishing a variational model
which incorporates the learned patchNR in the regularizer and to compute the unknown image as its minimizer.

Learning patchNRs Let \( p_1, \ldots, p_N \in \mathbb{R}^{s_1 \times s_2}, s_i \ll d_i, i = 1, 2 \) denote all possibly overlapping patches of the example images, where we assume that the patches \( p_1, \ldots, p_N \) are realizations of an absolute continuous probability distribution \( Q \) with density \( q \). We aim to approximate \( q \) by a NF. For simplicity, we rearrange the images and the patches into vectors of size \( d = d_1 d_2 \) and \( s := s_1 s_2 \), respectively. Then we learn a diffeomorphism \( T = T_\theta : \mathbb{R}^s \to \mathbb{R}^s \) such that \( Q \approx T_\# P_Z := P_Z \circ T^{-1} \), where \( P_Z \) is a \( s \)-dimensional standard normal distribution. To this end, we set our normalizing flow \( T \) to be an invertible neural network with parameters collected in \( \theta \). There were several approaches proposed in literature to construct such invertible neural networks \([5, 13, 19, 43, 56]\). In this paper, we adapt the architecture from \([5]\) as detailed in Appendix I. In order to train our NF, we aim to minimize the (forward) Kullback-Leibler divergence

\[
\text{KL}(Q, T_\# P_Z) := \int_{\mathbb{R}^s} \log \left( \frac{dQ}{dT_\# P_Z} \right) dQ = \mathbb{E}_{p\sim Q} \left[ \log \left( \frac{q(p)}{p_{T_\#}p_Z(p)} \right) \right],
\]

where we set the expression to \(+\infty\) if \( Q \) is not absolutely continuous with respect to \( T_\# P_Z \); see Appendix C for a short discussion. Noting that the first term is a constant independent of the parameters \( \theta \) and using the change-of-variables formula for probability density functions of push-forward measures \( p_{T_\#}p_Z = p_Z(T^{-1})|\det(\nabla T^{-1})| \), we obtain that the above formula is up to a constant equal to

\[
-\mathbb{E}_{p\sim Q} \left[ \log \left( p_Z(T^{-1}(p)) + \log \left| \det(\nabla T^{-1}(p)) \right| \right) \right].
\]

By replacing the expectation by the empirical mean of our training set, inserting the standard normal density \( p_Z \) and neglecting some constants, we finally obtain the loss function

\[
\mathcal{L}(\theta) := \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \left\| T^{-1}(p_i) \right\|^2 - \log \left( \left| \det(\nabla T^{-1}(p_i)) \right| \right). \tag{3}
\]

Now, we minimize this loss function using the Adam optimizer \([42]\).

Reconstruction with PatchNRs Once the patchNR \( T \) is learned, we aim to use it within a regularizer of the variational model \([2]\) to solve the inverse problem \([1]\). To this end, denote by \( E_i : \mathbb{R}^d \to \mathbb{R}^s, i = 1, \ldots, N_p \) the operator, which extracts the \( i \)-th (vectorized) patch from the unknown (vectorized) image \( x \in \mathbb{R}^d \). Then, we define our regularizer by the negative log likelihood of all patches under the probability distribution learned by the patchNR. More precisely, we define the patchNR based prior

\[
\frac{1}{s} \sum_{i=1}^{N_p} -\log \left( p_{T_\#}p_Z(E_i(x)) \right),
\]

where \( N_p \) is the number of patches in the image \( x \) and \( s = s_1 s_2 \) the number of pixels in a patch. Similar to the previous paragraph, this can be reformulated by the change-of-variables formula and by ignoring some constants as

\[
\text{patchNR}(x) := \frac{1}{s} \sum_{i=1}^{N_p} \frac{1}{2} \left\| T^{-1}(E_i(x)) \right\|^2 - \log \left( \left| \det(\nabla T^{-1}(E_i(x))) \right| \right).
\]

Note that if we ignore the boundary of the image, the patchNR is translation invariant. That is, a translation of the image does not change the value of the regularizer. Now, we reconstruct our ground truth by finding a minimizer of the variational problem

\[
\mathcal{J}(x; y) = D(f(x), y) + \lambda \text{patchNR}(x), \quad \lambda > 0. \tag{4}
\]

For the minimization, we use the Adam optimizer \([42]\). To speed up the numerical computations, we do not consider all overlapping patches in \( x \), but choose randomly a subset of \( N_p \) possibly overlapping patches in each iteration. Note that the resulting optimization problem is non-convex and therefore the choice of initialization is important. In our experiments we initialize this optimization with a coarse reconstruction, i.e., a bicubic interpolation for superresolution or the filtered backprojection (FBP) for CT.
Remark 1 (Relation to EPLL). Our patchNR is closely related to the expected patch log likelihood (EPLL) prior proposed by Zoran and Weiss in [91]. Here, the authors use the prior defined as

$$\text{EPLL}(x) = \frac{1}{N_p} \sum_{i=1}^{N_p} - \log \left(p(E_i(x))\right),$$

where $p$ is the probability density function of a GMM approximating the patch distribution of the image class of interest. However, GMMs have a limited expressiveness and can only hardly approximate complicated probability distributions induced by patches [23]. Further, the reconstruction process proposed in [91] is computationally very costly even though a reduction of the computational effort was considered in several papers [64, 74]. Indeed, we will show in our numerical examples that the patchNR clearly outperforms the reconstructions from EPLL.

3 Conditional Patch Normalizing Flows

If supervised training images are available, its use can improve the reconstruction quality. We show how to incorporate such information. Let few training images $x_1, ..., x_M$ and corresponding observations $y_1, ..., y_M$ from the inverse problem (1) be given. Then the additional information could be for example a corresponding patch from the corrupted image or, as in our numerical example, images $\tilde{x}_i$ reconstructed from $y_i$, $i = 1, \ldots, M$ by a straightforward method. For CT, this could be the FBP of the sinogram and in superresolution, the bicubic interpolation of the low-resolution image. Similar as for the unconditional patchNR, we proceed in two steps. First, we learn a conditional NF, which models the conditional patch distribution of the training images given the patches of the additional images. Afterwards, we use this conditional patchNR for the reconstruction.

Learning the Conditional patchNR. Assuming that we have given $N$ example patches $p_1, ..., p_N$ and corresponding additional information $c_1, ..., c_N$, we now aim to approximate the underlying conditional distribution of the patches. To this end, we train a conditional NF $f$ [18, 27]. For two random variables $X_p$ and $Y_c$, a conditional NF aims to approximate the conditional distributions $P_{X_p|Y_c=c}$ for all possible observations of $c$. Formally, a conditional NF is a learned mapping $T = T_\theta : \mathbb{R}^c \times \mathbb{R}^s \rightarrow \mathbb{R}^s$ such that $T(c; \cdot)$ is invertible for any $c \in \mathbb{R}^c$. Then, $T$ is trained such that it holds $T(c; \cdot) P_Z \approx P_{X_p|Y_c=c}$ for all $c$. For this purpose, we minimize the expected KL divergence of $P_{X_p|Y_c=c}$ and $T(c; \cdot) P_Z$, i.e.,

$$\mathbb{E}_{c \sim P_{Y_c}} \left[ \text{KL}(P_{X_p|Y_c=c}, T(c; \cdot) P_Z) \right] = \mathbb{E}_{c \sim P_{Y_c}} \left[ \mathbb{E}_{p \sim P_{X_p|Y_c=c}} \left[ \log \left( \frac{P_{X_p|Y_c=c}(p)}{P_T(c; \cdot) P_Z(p)} \right) \right] \right].$$

Since the numerator is constant and using the law of total probability, this simplifies to

$$- \mathbb{E}_{(p,c) \sim P_{(X_p,Y_c)}} \left[ \log \left( P_T(c; p) \right) \right].$$

Now, the change-of-variables formula and a discretization of the expectation yield the loss function

$$L(\theta) := \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \left\| T^{-1}(c_i; p_i) \right\|^2 - \log \left( \frac{1}{\det(\nabla T^{-1}(c_i; p_i))} \right),$$

where $T^{-1}$ denotes the inverse of $T$ with respect to the second argument.

Reconstruction with Conditional PatchNRs. Once, the conditional patchNR is learned, we generalize the regularizer to a conditional patchNR by considering the negative log-likelihood of all patches in the image as for the unconditional patchNR. Applying the change-of-variables formula and ignoring the constants we get

$$\text{cPatchNR}(x, y) := \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \left\| T^{-1}(C_i(y); E_i(x)) \right\|^2 - \log \left( \frac{1}{\det(\nabla T^{-1}(C_i(y); E_i(x)))} \right),$$

where $C_i$ extracts the $i$-th information vector from $y$. 
4 Analysis of Patch Normalizing Flows

In this section, we investigate the patch distribution which is approximated by the patchNR. More precisely, we show that any probability density on the class of images induces a probability density on the space of patches and vice versa. The proofs from this section are given in Appendix [B].

Let $X : \Omega \rightarrow \mathbb{R}^d$ with $X \sim P_X$ be a $d$-dimensional random variable on the space of images. By $E_i : \mathbb{R}^d \rightarrow \mathbb{R}^{d-s}$ we denote the operator which extracts all pixels from a $d$-dimensional image, which do not belong to the $i$-th patch. Let $I : \Omega \rightarrow \{1, ..., N_p\}$ be a random variable which follows the uniform distribution on $\{1, ..., N_p\}$. Then, the random variable $\omega \rightarrow E_I(\omega)(X(\omega))$ describes the selection of a random patch from a random image. We call the distribution $Q$ of $E_I(X)$ the patch distribution corresponding to $P_X$. The following lemma provides an explicit formula for the density of $Q$.

**Lemma 2.** Let $P_X$ be a probability distribution on $\mathbb{R}^d$ with density $p_X$. Then, also the corresponding patch distribution $Q$ is absolute continuous with density

$$q(p) = \frac{1}{N_p} \sum_{i=1}^{N_p} \int_{\mathbb{R}^{d-s}} p_X(E_i^T(p) + E_i^T(\tilde{x})) d\tilde{x}$$

For the proof of the reverse direction, namely that given a probability measure $Q$ on the space of patches, the patchNR defines a probability density function on the space of all images, we need the following lemma. It states that the density induced by a NF with a Gaussian latent distribution is up to a constant bounded from below and above by the density of certain normal distributions. In particular, it has exponential asymptotic decay. Note that similar questions about bi-Lipschitz continuous diffeomorphisms were investigated more detailed in [28, 36]. However, as we did not find the particular lemma in the literature, we give a proof within the appendix.

**Lemma 3.** Let $T : \mathbb{R}^s \rightarrow \mathbb{R}^s$ be a diffeomorphism with Lipschitz constants $\text{Lip}(T) \leq K$ and $\text{Lip}(T^{-1}) \leq L$ and let $P_Z = \mathcal{N}(0, I)$. Then, it holds

$$\frac{1}{L^s K^s} \mathcal{N}(0, I) \leq p_{\mathcal{N}(\mathbb{R}^s)}(p) \leq \frac{1}{L^s K^s} \mathcal{N}(0, I)$$

for any $p \in \mathbb{R}^s$.

Now, we can show that the patchNR defines a probability distribution on the space of all images. This allows a MAP interpretation of the variational problem [2].

**Proposition 4.** Let $P_Z = \mathcal{N}(0, I)$ and let $T : \mathbb{R}^s \rightarrow \mathbb{R}^s$ be a bi-Lipschitz diffeomorphism, i.e., $\text{Lip}(T) < \infty$ and $\text{Lip}(T^{-1}) < \infty$. Then, for any $\rho > 0$ the function $\varphi(x) = \exp(-\rho \text{patchNR}(x))$ belongs to $L^1(\mathbb{R}^d)$, where

$$\text{patchNR}(x) = \frac{1}{s} \sum_{i=1}^{N_p} -\log(p_{\mathcal{N}(\mathbb{R}^s)}(E_i(x))).$$

5 Numerical Examples

In this section, we demonstrate the performance of our method. We focus on linear inverse problems, but the approach can also be extended to non-linear forward operators. First, in Section [5.1], we consider super-resolution on material data. This is a typical setting where only little data is available or super-resolution is needed to obtain sufficient detail for material research [16, 38, 67]. Then we apply the patchNR to low-dose CT in a full angle and a limited angle setting. Lastly, we consider deblurring on texture images [4]. We investigate the sensitivity of our model on the hyperparameters by a detailed ablation study for the full angle CT in Appendix [J]. Further, we present an outlook on a possible extension of patchNRs for zero-shot super-resolution in Appendix [A]. Details towards the data fidelity terms are given in Appendix [F]. As the patchNR reconstruction process is non-convex we provide an empirical convergence study for full angle CT in Appendix [K].

4the code for all experiments is available at [https://github.com/FabianAltekrueger/patchNR](https://github.com/FabianAltekrueger/patchNR)
Comparison methods. We compare our method with established methods from the literature. In particular, we compare with Wasserstein Patch Prior (WPP) [3, 30], Expected Patch Log Likelihood (EPLL) [91] and Local Adversarial Regularizer (localAR) [65], as they work on patches and are model-based as well. Note that we optimize the EPLL GMM prior using a gradient descent optimizer, as half quadratic splitting proposed by the authors of [91] is much more expensive for the superresolution and CT forward operator. Moreover, we include comparisons with Plug-and-Play Forward Backward Splitting with DRUNet (DPIR) [89], the Deep Image Prior in combination with a TV prior (DIP+TV) [81, 10], which does not need any training data, and with supervised methods as the post-processing UNet (FBP+UNet) [37, 70] for CT and an asymmetric CNN (ACNN) [80] for superresolution. Note that post-processing methods are no longer the state-of-the-art for CT reconstruction, but are still widely used and serve as a comparison to an exemplary supervised method. Currently learned iterative methods are providing better results [2]. We will evaluate our methods using different quality measures, such as PSNR, SSIM [85], LPIPS [90] and Blur effect [14]. Details towards the comparison methods and quality measures are given in Appendix E.

5.1 Superresolution

We choose the forward operator $f$ as a convolution with a $16 \times 16$ Gaussian blur kernel with standard deviation 2 and stride 4. To keep the dimensions consistent, we use zero-padding. For the experiments we consider material data which was also used in [3, 30, 32]. A series of multi-scale 3D images has been acquired by synchrotron micro-computed tomography at the SLS beamline TOMCAT. We consider a composite ("SiC Diamonds") obtained by microwave sintering of silicon and diamonds, see [82]. In our experiments we consider a voxel spacing of $1.625 \mu m$. From this 3D image we extract 2D slices of size $600 \times 600$ and use them as ground truth and example images for our experiments. We generate low resolution images by using the predefined forward operator and adding additive Gaussian noise with standard deviation $\sigma = 0.01$. The patchNR is trained on patches of only one example image of size $600 \times 600$. Since we do not want to consider boundary effects, we cut off a boundary of 40 pixels before evaluating the quality measures.

In Figure 1 we compare different methods for reconstructing the high resolution image $x$ from the given low-resolution observation $y$. The use of the patchNR yields very clear and better images than the other model-based methods, visually and in terms of the quality metrics; see Table 1 for an average over 100 test images. In particular, the reconstruction of patchNR is less blurry than the DIP+TV and WPP reconstruction, specifically in the regions between the edges. If we have a

---

Table 1: Superresolution. Averaged quality measures and standard deviations of the high resolution reconstructions. The best values (model-based) are marked in bold.

| Method       | PSNR       | Blurred Effect | LPIPS      | SSIM        | Runtime |
|--------------|------------|----------------|------------|-------------|---------|
| bicubic      | 25.63 ± 0.56 | 0.533 ± 0.008 | 0.406 ± 0.013 | 0.699 ± 0.0012 | 0.0002s |
| DIP         | 27.18 ± 0.53 | 0.444 ± 0.011 | 0.322 ± 0.015 | 0.770 ± 0.011 | 56.62s  |
| DIP+TV      | 27.99 ± 0.54 | 0.372 ± 0.006 | 0.191 ± 0.009 | 0.764 ± 0.007 | 2.34308s|
| EPLL        | 28.11 ± 0.55 | 0.407 ± 0.009 | 0.244 ± 0.012 | 0.779 ± 0.010 | 0.012s  |
| WPP         | 28.80 ± 0.37 | 0.360 ± 0.008 | 0.167 ± 0.014 | 0.749 ± 0.0011 | 0.012s  |
| patchNR      | 28.53 ± 0.49 | 0.360 ± 0.008 | 0.159 ± 0.008 | 0.760 ± 0.008 | 0.028s  |
| ACNN (supervised) | 28.59 ± 0.52 | 0.392 ± 0.006 | 0.159 ± 0.008 | 0.789 ± 0.009 | 153.79s |
| cPatchNR (one-shot) | 28.59 ± 0.52 | 0.367 ± 0.005 | 0.203 ± 0.011 | 0.789 ± 0.009 | 153.79s |

---

Figure 1: Comparison of different methods for superresolution. Top: full image. Bottom: zoomed-in part.
low-resolution counterpart of our example image, the results of the patchNR can be improved by the cPatchNR.

5.2 Computerized Tomography

For computerized tomography (CT) we use the LoDoPaB dataset \cite{lo dopab} for low-dose CT imaging. It is based on scans of the Lung Image Database Consortium and Image Database Resource Initiative \cite{lidc} which serve as ground truth images, while the measurements are simulated. The dataset contains 35820 training images, 3522 validation images and 3553 test images. Here the ground truth images have a resolution of 362 × 362px. The LoDoPab dataset uses a two-dimensional parallel beam geometry with equidistant detector bins. The forward operator is the discretized linear Radon transformation and the noise can be modelled using a Poisson distribution. We trained the PatchNR using patches of a small set of 6 handpicked CT ground truth images illustrated in Figure 8. Once trained, the PatchNR can be used both for the full angle CT and the limited angle CT setting.

**Full angle CT** For full angle CT we consider 1000 equidistant angles between 0 and $\pi$. In Figure 2 we compare different methods for full angle low-dose CT imaging. Here the patchNR yields better results than DIP+TV and localAR, in particular the edges are sharper and more realistic in the reconstruction of patchNR. Note that the small blur effect in FBP and DIP+TV come from a large number of high frequency artifacts. Visually, there are only small differences between patchNR and FBP+UNet observable, although FBP+UNet is a fully supervised method trained on 35820 image pairs, while we only used 6 ground truth images for training the patchNR. The quality measures averaged over the first 100 test images of the LoDoPaB dataset in Table 3 confirm these observations.

**Limited angle CT** Now we consider the limited angle CT reconstruction problem, i.e., instead of considering equidistant angles between 0 and $\pi$ we only have a subset of angles. In our experiment we cut off the first and last 100 angles, i.e., we cut off 36° out of 180°. This leads to a much worse FBP reconstruction. In Figure 3 we compare the different reconstruction methods for the limited angle problem. Although the FBP shows a very bad reconstruction in the 36° part where the angles

---

Table 2: Full angle CT. Averaged quality measures and standard deviations of the reconstructions. The best values (model-based) are marked in bold. PSNR and SSIM were evaluated on an adaptive data range. Note that the diversity of the test set causes relatively high standard deviations.

| Method       | PSNR (dB) | Blur Effect | LPIPS | SSIM  | Runtime |
|--------------|-----------|-------------|-------|-------|---------|
| FBP          | 30.37 ± 2.95 | 0.412 ± 0.067 | 0.261 ± 0.130 | 0.739 ± 0.141 | 0.03s   |
| DIP + TV     | 34.45 ± 4.20 | 0.407 ± 0.049 | 0.181 ± 0.168 | 0.821 ± 0.147 | 1514.33s |
| EPLL         | 34.89 ± 4.41 | 0.437 ± 0.046 | 0.204 ± 0.177 | 0.821 ± 0.154 | 36.03s  |
| localAR      | 33.64 ± 3.74 | 0.419 ± 0.054 | 0.202 ± 0.164 | 0.807 ± 0.145 | 30.03s  |
| patchNR      | **35.19 ± 4.52** | **0.428 ± 0.048** | **0.191 ± 0.176** | **0.829 ± 0.152** | **48.39s** |
| FBP+UNet (supervised) | 35.48 ± 4.52 | 0.417 ± 0.049 | 0.162 ± 0.143 | 0.837 ± 0.143 |

---

Figure 2: Full angle reconstruction of the ground truth CT image using different methods. **Top:** full image. **Bottom:** zoomed-in part.
Table 3: Limited angle CT. Averaged quality measures and standard deviations of the reconstructions. The best values (model-based) are marked in bold. PSNR and SSIM were evaluated on an adaptive data range. Note that the diversity of the test set causes relatively high standard deviations.

| Method       | PSNR  | Blur Effect | LPIPS  | SSIM  | Runtime |
|--------------|-------|-------------|--------|-------|---------|
| FBP          | 21.96 ± 2.25 | 0.467 ± 0.082 | 0.305 ± 0.117 | 0.531 ± 0.097 | 0.02s   |
| FBP+DIP+TV   | 32.57 ± 3.25  | 0.450 ± 0.050  | 0.191 ± 0.165  | 0.803 ± 0.146  | 1770.89s |
| EPLL         | 32.78 ± 3.46  | 0.466 ± 0.045  | 0.216 ± 0.175  | 0.801 ± 0.151  | 127.21s  |
| localAR      | 31.06 ± 2.95  | 0.476 ± 0.048  | 0.222 ± 0.166  | 0.779 ± 0.142  | 53.47s   |
| patchNR      | 33.20 ± 3.55  | 0.453 ± 0.050  | 0.201 ± 0.176  | 0.811 ± 0.151  | 485.93s  |
| FBP+UNet (supervised) | 33.75 ± 3.58  | 0.439 ± 0.052  | 0.171 ± 0.134  | 0.820 ± 0.140  | 0.53s    |

Figure 3: Limited angle reconstruction of the ground truth CT image using different methods. Top: full image. Middle and bottom: zoomed-in parts.

are cut off, the patchNR can reconstruct these details well and in a realistic manner. In particular, the edges of patchNR reconstruction are preserved, while for the other methods these have a pronounced blur, see Table 3 for an average of the quality measures over the first 100 test images.

5.3 Deblurring

We consider the Kylberg texture dataset available at [link](https://kylberg.org/kylberg-texture-dataset-v-1-0) which consists of 28 different texture classes with 190 unique images for each class. For our experiments we use images from the class “Lentils” and train a patchNR using only one example image. To simulate the blurred image, we use the first blurring kernel of [55] and add additional Gaussian noise with standard deviation $\sigma = 5/255$. In Figure 4 we compare different reconstructions for the deblurring task. We can observe that the patchNR can reconstruct the ground truth image well, in particular the structure of the surface of the lentils is preserved. For a quantitative evaluation on the 20 test images see Table 4. Note that using the blurred counterpart of the example image for training the cPatchNR leads to an improvement of the patchNR.

6 Discussion and Conclusions

In this paper we introduced patchNRs, which are patch-based NFs used for regularizing the variational modeling of inverse problems. The subsequent reconstruction method is model-based and does not require pairs of corrupted data and the ground-truths. But if such data pairs are available, the model can be generalized to conditional patchNRs. We demonstrated the performance of our method by numerical examples and showed that it leads to better results than comparable, established methods, visually and in terms of the quality measures. Note that it is not clear how patch based learning influences biases in datasets. Using a small number of images can be very dangerous as these datasets
Table 4: Deblurring. Averaged quality measures and standard deviations of the reconstructions. The best values (model-based) are marked in bold.

|       | DPIR  | DIP+TV | EPLL  | WPP  | patchNR | cPatchNR (one-shot) |
|-------|-------|--------|-------|------|---------|---------------------|
| PSNR  | 30.53 ± 0.56 | 28.86 ± 0.54 | 31.05 ± 0.45 | 29.63 ± 0.42 | **31.09 ± 0.48** | 31.23 ± 0.50 |
| Blur Effect | 0.542 ± 0.021 | 0.550 ± 0.009 | 0.458 ± 0.012 | **0.446 ± 0.015** | 0.457 ± 0.013 | 0.464 ± 0.013 |
| LPIPS | 0.317 ± 0.023 | 0.246 ± 0.017 | 0.109 ± 0.018 | 0.113 ± 0.010 | **0.104 ± 0.015** | 0.103 ± 0.016 |
| SSIM  | 0.805 ± 0.018 | 0.767 ± 0.019 | **0.833 ± 0.010** | 0.789 ± 0.009 | 0.833 ± 0.010 | 0.836 ± 0.011 |
| Runtime | 14.19s | 747.53s | 62.85s | 245.38s | 96.94s | 99.69s |

Figure 4: Image deblurring on the texture "Lentils". Top: full image. Bottom: zoomed-in part.

are easily imbalanced. Further research needs to be invested into understanding how many images are sufficient for an adequate patch representation of a dataset and when the patch representation can be used as a prior for inverse problems. Moreover, quality measures for images are not sufficient for judging the quality of an image, in particular in medical applications. Further evaluation with medical expertise needs to be done before drawing any conclusions. Moreover, all patch-based methods are essentially limited in the sense that they can not capture global image correlations. Furthermore, normalizing flows are often not able to capture out of distribution data [45], so if a patch is far away from the patch "manifold", the likelihoods might be meaningless. However, it is an open question whether patch-based learning mitigates this effect.

The patchNR can be extended into several directions. First, we want to use the regularizer for training NNs in an model-based way for a fast reconstruction of several observations. Then the patchNR can be applied for uncertainty quantification by using, e.g., invertible architectures [5, 19, 27] or Langevin sampling methods [78]. Furthermore, the use of deep features instead of patches is another fruitful research direction. Finally, an application of patchNRs to zero-shot superresolution is described in detail in Appendix A.

Acknowledgments and Disclosure of Funding

F.A. acknowledge support from the German Research Foundation (DFG) under Germany’s Excellence Strategy – The Berlin Mathematics Research Center MATH+ within the project TP: EF3-7, A.D. from (DFG; GRK 2224/1) and the Klaus Tschira Stiftung via the project MALDISTAR (project number 00.010.2019), P.H. from the DFG within the project SPP 2298 "Theoretical Foundations of Deep Learning" and J.H. by the DFG within the project STE 571/16-1. The data from Section [5, 1] has been acquired in the frame of the EU Horizon 2020 Marie Sklodowska-Curie Actions Innovative Training Network MUMMERING (Multiscale, Multimodal and Multidimensional imaging for EngineRiNG, Grant Number 765604) at the beamline TOMCAT of the SLS by A. Saadaldin, D. Bernard, and F. Marone Welford. We acknowledge the Paul Scherrer Institut, Villigen, Switzerland for provision of synchrotron radiation beamtime at the TOMCAT beamline X02DA of the SLS. P.H. thanks Cosmas Heiß and Shayan Hundrieser for fruitful discussions.
References

[1] J. Adler, H. Kohr, A. Ringh, J. Moosmann, S. Banert, M. J. Ehrhardt, G. R. Lee, K. Niinimaki, B. Gris, O. Verdier, J. Karlsson, W. J. Palenstijn, O. Öktem, C. Chen, H. A. Loarca, and M. Lohmann. Operator discretization library (ODL), 2018.

[2] J. Adler and O. Öktem. Learned primal-dual reconstruction. IEEE Transactions on Medical Imaging, 37(6):1322–1332, Jun 2018.

[3] F. Altekrüger and J. Hertrich. WPPNets and WPPFlows: The power of Wasserstein patch priors for superresolution. arXiv preprint arXiv:2201.08157, 2022.

[4] R. Anirudh, J. J. Thiagarajan, B. Kailkhura, and T. Bremer. An unsupervised approach to solving inverse problems using generative adversarial networks. arXiv preprint arXiv:1805.07281, 2018.

[5] L. Ardizzone, J. Kruse, C. Rother, and U. Köthe. Analyzing inverse problems with invertible neural networks. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.

[6] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe. Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02392, 2019.

[7] S. G. Armato et al. The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): a completed reference database of lung nodules on CT scans. Medical physics, 38(2):915–931, 2011.

[8] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. Acta Numerica, 28:1–174, 2019.

[9] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand. Invertible generative models for inverse problems: mitigating representation error and dataset bias. In H. D. III and A. Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 399–409. PMLR, 13–18 Jul 2020.

[10] D. O. Baguer, J. Leuschner, and M. Schmidt. Computed tomography reconstruction using deep image prior and learned reconstruction methods. Inverse Problems, 36(9):094004, 2020.

[11] R. Barbano, J. Leuschner, M. Schmidt, A. Denker, A. Hauptmann, P. Maass, and B. Jin. Is deep image prior in need of a good education? arXiv preprint arXiv:2111.11926, 2021.

[12] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2, pages 60–65. IEEE, 2005.

[13] R. Chen, J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen. Residual flows for invertible generative modeling. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

[14] F. Crete, T. Dolmiere, P. Ladret, and M. Nicolas. The blur effect: perception and estimation with a new no-reference perceptual blur metric. In Human vision and electronic imaging XII, volume 6492, page 64920I. International Society for Optics and Photonics, 2007.

[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiaziarian. BM3D image denoising with shape-adaptive principal component analysis. In SPARS’09-Signal Processing with Adaptive Sparse Structured Representations, 2009.

[16] A. Dahari, S. Kench, I. Squires, and S. J. Cooper. Super-resolution of multiphase materials by combining complementary 2d and 3d image data using generative adversarial networks. arXiv preprint arXiv:2110.11281, 2021.

[17] J. Delon and A. Houdard. Gaussian priors for image denoising. In Denoising of Photographic Images and Video, pages 125–149. Springer, 2018.

[18] A. Denker, M. Schmidt, J. Leuschner, and P. Maass. Conditional invertible neural networks for medical imaging. Journal of Imaging, 7(11), 2021.

[19] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017.

[20] M. Duff, N. D. F. Campbell, and M. J. Ehrhardt. Regularising inverse problems with generative machine learning models. arXiv preprint arXiv:2107.11191, 2021.
[21] M. Emad, M. Peemen, and H. Corporaal. DualSR: Zero-shot dual learning for real-world super-resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1629–1638, 2021.

[22] R. Friedman and Y. Weiss. Posterior sampling for image restoration using explicit patch priors. *arXiv preprint arXiv:2104.09895*, 2021.

[23] C. R. Genovese and L. Wasserman. Rates of convergence for the Gaussian mixture sieve. *The Annals of Statistics*, 28(4):1105–1127, 2000.

[24] D. Gilton, G. Ongie, and R. Willett. Learned patch-based regularization for inverse problems in imaging. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 211–215. IEEE, 2019.

[25] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *2009 IEEE 12th international conference on computer vision*, pages 349–356. IEEE, 2009.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[27] P. Hagemann, J. Hertrich, and G. Steidl. Stochastic normalizing flows for inverse problems: a Markov Chains viewpoint. *arXiv preprint arXiv:2109.11375*, 2021.

[28] P. Hagemann and S. Neumayer. Stabilizing invertible neural networks using mixture models. *Inverse Problems*, 37(8):085002, 2021.

[29] L. Helminger, M. Bernasconi, A. Djelouah, M. H. Gross, and C. Schroers. Generic image restoration with flow based priors. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 334–343, 2021.

[30] J. Hertrich, A. Houdard, and C. Redenbach. Wasserstein patch prior for image superresolution. *arXiv preprint arXiv:2109.12880*, 2021.

[31] J. Hertrich, S. Neumayer, and G. Steidl. Convolutional proximal neural networks and plug-and-play algorithms. *Linear Algebra and its Applications*, 631:203–234, 2021.

[32] J. Hertrich, D. P. L. Nguyen, J.-F. Aujol, D. Bernard, Y. Berthoumieu, A. Saadaldin, and G. Steidl. PCA reduced Gaussian mixture models with applications in superresolution. *Inverse Problems & Imaging*, 2021.

[33] A. Houdard, C. Bouveyron, and J. Delon. High-dimensional mixture models for unsupervised image denoising (HDMI). *SIAM Journal on Imaging Sciences*, 11(4):2815–2846, 2018.

[34] S. Hurault, A. Leclaire, and N. Papadakis. Gradient step denoiser for convergent plug-and-play. In *International Conference on Learning Representations*, 2022.

[35] J. Jacod and P. Protter. *Probability Essentials*. Springer-Verlag Berlin Heidelberg, 2004.

[36] P. Jaini, I. Kobyzev, Y. Yu, and M. Brubaker. Tails of lipschitz triangular flows. In *International Conference on Machine Learning*, pages 4673–4681. PMLR, 2020.

[37] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[38] J. Jung, J. Na, H. K. Park, J. M. Park, G. Kim, S. Lee, and H. S. Kim. Super-resolving material microstructure image via deep learning for microstructure characterization and mechanical behavior analysis. *npj Computational Materials*, 7(1):96, 2021.

[39] B. Kawar, M. Elad, S. Ermon, and J. Song. Denoising diffusion restoration models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[40] B. Kawar, G. Vaksman, and M. Elad. SNIPS: Solving noisy inverse problems stochastically. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[41] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

[42] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
[43] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[44] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[45] P. Kirichenko, P. Izmailov, and A. G. Wilson. Why normalizing flows fail to detect out-of-distribution data. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, page 20578–20589. Curran Associates, Inc., 2020.

[46] E. Kobler, A. Effland, K. Kunisch, and T. Pock. Total deep variation for linear inverse problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7549–7558, 2020.

[47] E. Kobler, A. Effland, K. Kunisch, and T. Pock. Total deep variation: A stable regularization method for inverse problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[48] M. D. Kohli, R. M. Summers, and J. R. Geis. Medical image data and datasets in the era of machine learning—whitepaper from the 2016 C-MIMI meeting dataset session. *Journal of Digital Imaging*, 30(4):392–399, 2017.

[49] J. Kruse, G. Detommaso, R. Scheichl, and U. Köthe. Hint: Hierarchical invertible neural transport for density estimation and bayesian inference. *arXiv preprint arXiv:1905.10687v1*, 2019.

[50] G. Kyllberg. *The Kylberg texture dataset v. 1.0*. Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, 2011.

[51] F. Laus, M. Nikolova, J. Persch, and G. Steidl. A nonlocal denoising algorithm for manifold-valued images using second order statistics. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

[52] M. Lebrun, A. Buades, and J.-M. Morel. A nonlocal Bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

[53] J. Leuschner, M. Schmidt, D. O. Baguer, and P. Maass. LoDoPaB-CT, a benchmark dataset for low-dose computed tomography reconstruction. *Scientific Data*, 8(109), 2021.

[54] J. Leuschner, M. Schmidt, P. S. Ganguly, V. Andriashen, S. B. Coban, A. Denker, D. Bauer, A. Hadjiaradjji, K. J. Batenburg, P. Maass, and M. van Eijnatten. Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications. *Journal of Imaging*, 7(3), 2021.

[55] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971. IEEE, 2009.

[56] A. Lugmayr, M. Danelljan, L. Van Gool, and R. Timofte. SRFlow: Learning the super-resolution space with normalizing flow. In *ECCV*, 2020.

[57] S. Lunz, O. Öktem, and C.-B. Schönlieb. Adversarial regularizers in inverse problems. *Advances in neural information processing systems*, 31, 2018.

[58] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[59] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[60] S. Mukherjee, M. Carioni, O. Öktem, and C.-B. Schönlieb. End-to-end reconstruction meets data-driven regularization for inverse problems. *Advances in Neural Information Processing Systems*, 34:21413–21425, 2021.

[61] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don’t know? *arXiv preprint arXiv:1810.09136*, 2018.
[62] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willet. Deep learning techniques for inverse problems in imaging. *arXiv preprint arXiv:2005.06001*, 2020.

[63] X. Pan, X. Zhan, B. Dai, D. Lin, C. C. Loy, and P. Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *European Conference on Computer Vision (ECCV)*, 2020.

[64] S. Parameswaran, C. Deledalle, L. Denis, and T. Q. Nguyen. Accelerating GMM-based patch priors for image restoration: Three ingredients for a 100x speed-up. *IEEE Transactions on Image Processing*, 28(2):687–698, 2019.

[65] J. Prost, A. Houard, A. Almansa, and N. Papadakis. Learning local regularization for variational image restoration. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 358–370. Springer, 2021.

[66] J. Radon. On the determination of functions from their integral values along certain manifolds. *IEEE Transactions on Medical Imaging*, 5(4):170–176, 1986.

[67] E. J. Reid, L. F. Drummy, C. A. Bouman, and G. T. Buzzard. Multi-resolution data fusion for super resolution imaging. *IEEE Transactions on Computational Imaging*, 8:81–95, 2022.

[68] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[69] Y. Romano, M. Elad, and P. Milanfar. The little engine that could: Regularization by denoising (RED). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

[70] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015.

[71] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[72] L. Ruthotto and E. Haber. An introduction to deep generative modeling. *DMV Mitteilungen*, 44(3):1–24, 2021.

[73] P. Sandeep and T. Jacob. Single image super-resolution using a joint GMM method. *IEEE Transactions on Image Processing*, 25(9):4233–4244, 2016.

[74] H. Shi, Y. Traonmilin, and J.-F. Aujol. Compressive learning for patch-based image denoising. *HAL preprint hal-03429102*, 2021.

[75] A. Shocher, N. Cohen, and M. Irani. “Zero-shot” super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3118–3126, 2018.

[76] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, 28, 2015.

[77] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[78] Y. Song, L. Shen, L. Xing, and S. Ermon. Solving inverse problems in medical imaging with score-based generative models. In *The 10th International Conference on Learning Representations*, 2022.

[79] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman. Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, 2(4):408–423, 2016.

[80] C. Tian, Y. Xu, W. Zuo, C.-W. Lin, and D. Zhang. Asymmetric CNN for image superresolution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[81] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[82] S. Vaucher, P. Unifantowicz, C. Ricard, L. Dubois, M. Kuball, J.-M. Catala-Civera, D. Bernard, M. Stampanoni, and R. Nicula. On-line tools for microscopic and macroscopic monitoring of microwave processing. *Physica B: Condensed Matter*, 398(2):191–195, 2007.
[83] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In 2013 IEEE Global Conference on Signal and Information Processing, pages 945–948. IEEE, 2013.

[84] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys, 53(3), 2020.

[85] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.

[86] X. Wei, H. V. Gorp, L. G. Carabarín, D. Freedman, Y. C. Eldar, and R. van Sloun. Deep unfolding with normalizing flow priors for inverse problems. IEEE Transactions on Signal Processing, 70:2962–2971, 2022.

[87] J. Whang, Q. Lei, and A. Dimakis. Solving inverse problems with a flow-based noise model. In M. Meila and T. Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 11146–11157. PMLR, 18–24 Jul 2021.

[88] C. S. Xu, K. J. Hayworth, Z. Lu, P. Grob, A. M. Hassan, J. G. García-Cerdán, K. K. Niyogi, E. Nogales, R. J. Weinberg, and H. F. Hess. Enhanced FIB-SEM systems for large-volume 3D imaging. eLife, 6, 2017.

[89] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte. Plug-and-play image restoration with deep denoiser prior. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[90] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 586–595, 2018.

[91] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In IEEE International Conference on Computer Vision, pages 479–486. IEEE, 2011.
A Application: Zero-shot Superresolution with PatchNRs

Here we describe one possible extension of patchNRs in detail. We aim to combine some concepts of zero-shot superresolution by internal learning [25, 75] with patchNRs. In these approaches, the main assumption is that the patch distribution of natural images is self-similar across the scales. Consequently, the patch distributions of the same image at different resolutions should be similar. Motivated by this observation, we train the patchNR on the low-resolution observations such that we do not longer require any sample from the high-resolution ground truth.

In the following we consider the case, where we have given one single low-resolution observation at training time and additionally the forward operator at test time. In particular, we do not require access to any high-resolution ground truth image and therefore the method is fully unsupervised. We train the patchNR on the patches from the low-resolution observation using (3) and reconstruct the high-resolution prediction by minimizing (4). The training data is enriched by rotating and mirror reflecting of the patches such that we get 8 times more training patches. Note that in this setting the patchNR needs to be retrained for every new observation.

Numerical Simulations We use a convolution with a $16 \times 16$ Gaussian blur kernel with standard deviation 1 and stride 2 as forward operator and add Gaussian noise with standard deviation 0.01 on the low-resolution observation. The patchNR is trained for 10000 optimizer steps with a learning rate of 0.0001, a batch size of 128 and a patch size of $6 \times 6$. Then, for reconstruction we use a random subset of $N_p = 80000$ patches per iteration, a regularization parameter $\lambda = 0.25 \frac{1}{N_p}$ and optimize over 60 iterations using Adam with a learning rate of 0.01. As comparison baselines we use $L^2$-TV [71], DIP+TV, ZSSR [75] and DualSR [21] and we test the methods on the BSD68 dataset [58]. The resulting quality measures are given in Table 5. In Figure 5 we present three reconstruction examples of the test set. Reconstructions of the patchNR lead to less blurry images and in particular, structures and edges are preserved, while in $L^2$-TV and DIP+TV some parts of the images are smoothed out.

![Superresolution on BSD68 dataset. Top: full image. Bottom: zoomed-in part.](image)
Table 5: Superresolution. Averaged quality measures and standard deviations of the reconstructions of BSD68 dataset. The best values are marked in bold.

|       | $L^2$-TV   | DIP-TV    | ZSSR  | DualSR | patchNR |
|-------|------------|-----------|-------|--------|---------|
| PSNR  | 28.35 ± 3.55 | 28.44 ± 3.69 | 28.83 ± 3.57 | 28.64 ± 3.47 | 29.08 ± 3.58 |
| Blur Effect | 0.340 ± 0.045 | 0.348 ± 0.061 | 0.334 ± 0.054 | **0.326** ± 0.045 | 0.327 ± 0.049 |
| LPIPS  | **0.184** ± 0.073 | 0.215 ± 0.086 | 0.224 ± 0.085 | 0.216 ± 0.074 | 0.202 ± 0.076 |
| SSIM   | 0.820 ± 0.072 | 0.821 ± 0.087 | 0.834 ± 0.066 | 0.829 ± 0.061 | **0.846** ± 0.061 |
| Runtime | 13.12s | 171.31s | 36.64s | 53.47s | 132.36s |

B Proofs

**Proof of Lemma 2.** Let $A \in B(\mathbb{R}^s)$ be an arbitrary Borel set. Then, we have by Bayes’ theorem that

$$Q(A) = \sum_{i=1}^{N_p} P(I = i)E_i \# P_X(A) = \sum_{i=1}^{N_p} \frac{1}{N_p} \int_{\mathbb{R}^s} 1_A(E_i^{-1}(x)) p_X(x) dx.$$  

Now, we use the decomposition

$$\mathbb{R}^d = \text{Ker}(E_i) \oplus \text{Im}(E_i^T) = \text{Im}(E_i^T) \oplus \text{Im}(E_i^T).$$  

With Fubini’s theorem, $E_i E_i^T = I$ and $E_i \tilde{E}_i^T = 0$, this is equal to

$$Q(A) = \frac{1}{N_p} \sum_{i=1}^{N_p} \int_{\mathbb{R}^s} \int_{\mathbb{R}^d} 1_A(E_i E_i^T(p) + \tilde{E}_i^T(\tilde{x}))p_X(E_i^T(p) + \tilde{E}_i^T(\tilde{x})) d\tilde{x} dp$$

$$= \frac{1}{N_p} \sum_{i=1}^{N_p} \int_{\mathbb{R}^s} \int_{\mathbb{R}^d} 1_A(p)p_X(E_i^T(p) + \tilde{E}_i^T(\tilde{x})) d\tilde{x} dp$$

$$= \int_A \frac{1}{N_p} \sum_{i=1}^{N_p} \int_{\mathbb{R}^d} p_X(E_i^T(p) + \tilde{E}_i^T(\tilde{x})) d\tilde{x} dp.$$  

This proves the claim. 

**Proof of Lemma 3.** Using the Lipschitz continuity of $T$, we obtain

$$\frac{1}{K^2}||p - T(0)||^2 = \frac{1}{K^2}||T^{-1}(p) - T(0)||^2 \leq ||T^{-1}(p) - 0||^2 = ||T^{-1}(p)||^2$$

Now, applying the change-of-variables formula and that $|\text{det}(\nabla T^{-1}(p))| \leq L^*$, we conclude

$$p_{T^{-1}}(p) = p_Z(T^{-1}(p))|\text{det}(\nabla T^{-1}(p))| \leq \frac{L^*}{(2\pi)^{s/2}} \exp(-\frac{1}{2}||T^{-1}(p)||^2) \leq \frac{L^*}{(2\pi)^{s/2}} \exp(-\frac{1}{2}||p - T(0)||^2) = L^* K^* N(p|T(0), K^2 I).$$

This shows the second inequality. For the first inequality, note that by the inverse function theorem

$$\nabla T^{-1}(p) = \nabla T^{-1}(T(\nabla T^{-1}(p))) = (\nabla T^{-1}(p))^{-1}.$$  

Using the Lipschitz continuity of $T$, this implies

$$|\text{det}(\nabla T^{-1}(p))| = |\text{det}(\nabla T^{-1}(p))|^{-1} \geq 1/K^*.$$  

Further, by the Lipschitz continuity of $T^{-1}$ it holds that

$$||T^{-1}(p)||^2 = ||T^{-1}(p) - T^{-1}(T(0))||^2 \leq L^2||p - T(0)||^2.$$  

Putting the things together yields

$$p_{T^{-1}}(p) = p_Z(T^{-1}(p))|\text{det}(\nabla T^{-1}(p))| \geq \frac{1}{K^* (2\pi)^{s/2}} \exp(-\frac{1}{2}||T^{-1}(p)||^2) \geq \frac{1}{K^* (2\pi)^{s/2}} \exp(-\frac{L^2}{2}||p - T(0)||^2) = \frac{1}{L^* K^*} N(p|T(0), \frac{1}{L^*} I).$$

This completes the proof.
Proof of Proposition 4. Using Lemma 3, there exists some $C > 0$ such that it holds

$$
\int_{\mathbb{R}^d} \varphi(x) dx = \int_{\mathbb{R}^d} \left( \prod_{i=1}^{N_p} p(T = E_i(x)) \right)^{\rho/s} dx \\
\leq C \int_{\mathbb{R}^d} \left( \prod_{i=1}^{N_p} \mathcal{N}(E_i(x)|T(0), K^2 I) \right)^{\rho/s} dx \\
= C \int_{\mathbb{R}^d} \prod_{i=1}^{N_p} \prod_{j=1}^{s} \mathcal{N}(E_i(x)_j|T(0)_j, K^2)^{\rho/s} dx,
$$

where $(E_i(x))_j$ is the $j$-th element from $E_i(x)$. Since $(E_i(x))_j = x_{\sigma(i,j)}$ for some mapping $\sigma: \{1, \ldots, N_p\} \times \{1, \ldots, s\} \to \{1, \ldots, d\}$ and using Fubini’s theorem, this simplifies to

$$
C \int_{\mathbb{R}^d} \prod_{i=1}^{N_p} \prod_{j=1}^{s} \mathcal{N}(x_{\sigma(i,j)}|T(0)_j, K^2)^{\rho/s} dx \\
= C \int_{\mathbb{R}^d} \prod_{k=1}^{d} \prod_{(i,j) \in \sigma^{-1}\{k\}} \mathcal{N}(x_k|T(0)_j, K^2)^{\rho/s} dx \\
= C \prod_{k=1}^{d} \int_{\mathbb{R}^d} \prod_{(i,j) \in \sigma^{-1}\{k\}} \mathcal{N}(x|T(0)_j, K^2)^{\rho/s} dx.
$$

Here $\sigma^{-1}\{k\}$ denotes the preimage of $\sigma$ which denotes the set of all index pairs $(i, j)$ such that $(E_i(x))_j = x_k$ for $x \in \mathbb{R}^d$. As each pixel in the images is covered by at least one patch, this set is non-empty. Using the fact that products and powers of normal densities are integrable, we obtain that this expression is finite and the proof is complete.

\[\square\]

C  Section on the forward KL divergence

Note that the KL divergence is not symmetric. Consequently, one could also derive a loss function for normalizing flows by interchanging the arguments of the KL divergence [3, 27, 49]. The loss function arising in this way is often called backward KL, while our approach is called forward KL. However, the two loss functions differ in their assumptions and approximation properties, see e.g. [27]. In particular, the backward KL requires the knowledge of the target density $q$ which is not given in our case. Furthermore, the forward KL is known to be mode-covering, which is a desirable property for learning a possibly multimodal patch distribution. Therefore, we will stick to the forward KL.

D  Motivation for conditional patchNR

For many inverse problems the measurements can again be interpreted as an image. In this case we can use the conditional patchNR incorporating the information of paired data consisting of training images $x$ and corresponding observations $y$. This section aims to motivate the use of conditional patchNRs in such cases based on the example of superresolution from Section 5.1.

For superresolution, we can assign a low-resolution patch to a high resolution patch. Now, the conditional patchNR does not only learn the patch distribution, but it learns simultaneously the distribution of a high-resolution patch given a certain low-resolution patch. For instance, if the low-resolution patch shows an edge, it is very unlikely that the high-resolution patch is approximately constant. In contrast to the unconditional patchNR, the conditional patchNR will penalize an approximately constant patch at this point as it has access to this low-resolution information.

E  Quality measures and comparisons

Quality measures  For evaluation of the results we use different quality measures:
• **PSNR.** The peak-signal to noise ratio (PSNR) between two images \(x\) and \(y\) on \([0, 1]^{m \times n}\) is defined by

\[
\text{PSNR}(x, y) := 10 \log \left( \frac{\|x - y\|^2}{mn \cdot \max_x^2} \right),
\]

consequently larger PSNR values correspond to a better reconstruction. Here \(\max_x\) is the maximal possible pixel value of the image and in general we have \(\max_x = 1\). Nevertheless, in \([53]\) it is proposed to choose \(\max_x = \max(x) - \min(x)\) for CT since here the pixel values are far from most common values. Herewith we avoid too optimistic results. Note that the PSNR prefers smooth reconstructions and a high PSNR does not always coincides with a good visual impression, see e.g. \([85]\). Therefore, we consider further quality measures.

• **SSIM.** The structural similarity index measure (SSIM) \([85]\) compares the overall image structure of two images \(x\) and \(y\) on \([0, 1]^{m \times n}\). It is computed by moving a local window at \(M\) locations

\[
\text{SSIM}(x, y) := \frac{1}{M} \sum_{i=1}^{M} \frac{(2\mu_x^{(i)} \mu_y^{(i)} + C_1)(2\sigma_{xy}^{(i)} + C_2)}{((\mu_x^{(i)})^2 + (\mu_y^{(i)})^2 + C_1)((\sigma_x^{(i)})^2 + (\sigma_y^{(i)})^2 + C_2)},
\]

where \(\mu_x^{(i)}\) and \(\mu_y^{(i)}\) are the mean intensity, \(\sigma_x^{(i)}\) and \(\sigma_y^{(i)}\) are the standard deviation and \(\sigma_{xy}^{(i)}\) is the covariance of \(x\) and \(y\) at the local window \(i\). The local window is chosen to be \(7 \times 7\) and the constants \(C_1 = (K_1L)^2\), \(C_2 = (K_2L)^2\) ensure stability, where \(K_1 = 0.01\), \(K_2 = 0.03\) and \(L = \max(x) - \max(y)\) is the data range as in \([85]\).

• **Blur effect.** The blur effect \([14]\) is based on a comparison of the input image \(x\) with a blurred version \(x_{\text{blur}}\). A high variation between the input image and the blurred version implies that the input image is sharp, while for blurred \(x\) the variation is small. The blur effect is normalized to \([0, 1]\), where a small blur effect indicates that \(x\) is very sharp while a large blur effect means that \(x\) is very blurry.

• **LPIPS.** Finally, we use the learned perceptual image patch similarity (LPIPS) \([90]\) for measuring the perceptual similarity of our results and the ground truth. The basic idea of LPIPS is to compare the feature maps extracted from some deep neural network that is trained for some classical imaging task which is not necessarily related to our original problem. A small value of LPIPS indicates a high perceptual similarity.

**Description of compared methods**  Here we give a short overview about the comparison methods we used.

• **Bicubic interpolation.** For superresolution, the simplest comparison is the bicubic interpolation \([41]\), which is based on the local approximation of the image by polynomials of degree 3.

• **Filtered Backprojection and UNet.** For CT a classical method is the Filtered Backprojection (FBP) described by the adjoint Radon transform \([66]\). We use the ODL implementation \([1]\) for our experiments. There we choose the filter type Hann and a frequency scaling of 0.641. In order to improve the image quality of the FBP, a post-processing network can be learned. Here we consider the popular choice of a UNet (FBP+UNet) \([70]\), which was used in \([37]\) for CT imaging. We use the implementation from \([54]\) which is trained on the 35820 training images of LoDoPaB dataset.

• **Wasserstein Patch Prior.** The idea of the Wasserstein Patch Prior (WPP) \([3, 30]\) is to use the Wasserstein-2 distance between the patch distribution of the reconstruction and the patch distribution of a given reference image. Here a high resolution reference image \(\tilde{x}\) (or a high-resolution cutout) with a similar patch distribution as the unknown high resolution image \(x\) is needed. For a representation of structures of different sizes, \(x\) and \(\tilde{x}\) are considered at different scales \(x_1 = x, \tilde{x}_1 = \tilde{x}, x_2 = A\tilde{x}_1, \tilde{x}_2 = A\tilde{x}_{2-1}\) for a downsampling operator \(A\).

---

\[\text{We use the implementation }\text{https://github.com/johertrich/Wasserstein_Patch_Prior}\text{ version 0.1.}\]

\[\text{We use the implementation }\text{https://github.com/richzhang/PerceptualSimilarity} \text{ version 0.1.}\]

\[\text{We use the original implementation from }\text{https://github.com/johertrich/Wasserstein_Patch_Prior}\text{ available at }\text{https://github.com/johertrich/Wasserstein_Patch_Prior}\text{.}\]
Then the aim is to minimize the functional
\[
J(x) = D(f(x), y) + \lambda \sum_{l=1}^{L} W_{2}^{2}(\mu_{x_{l}}, \mu_{\tilde{x}_{l}}),
\]
where the patch distributions of \( x \) and \( \tilde{x} \) are defined by
\[
\mu_{x_{l}} = \frac{1}{N_{l}} \sum_{i=1}^{N_{l}} \delta_{p_{x_{l}}}, \quad \mu_{\tilde{x}_{l}} = \frac{1}{N_{l}} \sum_{i=1}^{N_{l}} \delta_{p_{\tilde{x}_{l}}},
\]

**Deep Image Prior with TV regularization.** The idea of the Deep Image Prior (DIP) [81] is to solve the optimization problem
\[
\hat{\theta} \in \arg \min_{\theta} D(f(G_{\theta}(z)), y),
\]
where \( G_{\theta} \) is a convolutional neural network with parameters \( \theta \) and \( z \) is a randomly chosen input. Then, the reconstruction \( \hat{x} \) is given by \( \hat{x} = G_{\theta}(z) \). It was shown in [81] that DIP admits competitive results for many inverse problems. A combination of DIP with the TV (DIP+TV) regularizer was successfully used in [10] for CT reconstruction. Here the optimization problem is extended to
\[
\hat{\theta} \in \arg \min_{\theta} D(f(G_{\theta}(z)), y) + TV((G_{\theta}(z)),
\]

Note that each reconstruction with the DIP+TV requires the training of a neural network. In contrast to WPP and patchNR, the DIP+TV is a data-free method, i.e., it does not require any clean image for training. We tested a pre-trained variant of the DIP+TV on the same training images. However, this did not improve the results significantly. Note that warm-start initialization techniques were proposed in [11]. Here, the authors observed faster reconstruction times for a pre-trained DIP but not significantly better results. Therefore we stick to the random initialization.

**Plug-and-Play Forward Backward Splitting with DRUNet.** In Plug-and-Play methods, first introduced by [83], the main idea is to consider an optimization algorithm from convex analysis for solving (2) and to replace the proximal operator with respect to the regularizer by a more general denoiser. Here, modify the forward backward splitting algorithm
\[
x_{n+1} = \text{prox}_{\eta R}(x_{n} - \eta \nabla x D(f(x_{n}), y))
\]
for minimizing the functional (2) by the iteration
\[
x_{n+1} = G(x_{n} - \eta \nabla x D(f(x_{n}), y)),
\]
where \( G \) is a neural network trained for denoising natural images. We use the DRUNet (DPIR) from [89] as denoiser and run it for 100 iterations. Note that the denoiser is trained on natural images and not on images from the specific image domain. However, as we have given only very few clean images from the considered image domain it is impossible to train a denoiser with comparable quality on them.

**Local Adversarial Regularizer.** The adversarial regularizer was introduced in [57] and this framework was recently applied for learning patch-based regularizers (localAR) [65]. The idea is to train a network \( r_{\theta} \) as a critic between patch distributions in order to distinguish between clean and degraded patches. The network is trained by minimizing
\[
D(\theta) = E_{z \sim P_{c}}[r_{\theta}(z)] - E_{z \sim P_{n}}[r_{\theta}(z)] + \mu E_{z \sim P_{c}}[\| \nabla z r_{\theta}(z) \|_{2} - 1]^{2},
\]
where \( P_{c} \) and \( P_{n} \) are the distributions of clean and noisy patches, respectively, and \( P_{i} \) is the distribution of all lines connecting samples in \( P_{c} \) and \( P_{n} \). Then the aim is to minimize the functional
\[
J(x) = D(f(x), y) + \frac{1}{|I|} \sum_{i \in I} r_{\theta}(P_{i}(x))
\]

For superresolution, we use the original implementation from [81] available at
\[https://github.com/DmitryUlyanov/deep-image-prior\] in combination with a TV regulariser; for CT, we use the original implementation from [10] available at
\[https://github.com/jleuschn/dival/blob/master/dival/reconstructors\]
For our experiments we used the code of [65], but instead of patch size 15 we used the patch size 6 and replaced the fully convolutional discriminator by a discriminator with 2 convolutional layers, followed by 4 fully connected layers.

- **Expected Patch Log Likelihood.** The Expected Patch Log Likelihood (EPLL) prior [91] assumes that the patch distribution of the ground truth can be approximated by a GMM $p$ fitted to the patch distribution of the reference images. Reconstruction is done by minimizing the functional

$$J(x) = D(f(x), y) - \lambda \sum_{i=1}^{N} p(P_i(x)).$$

In [91] the authors used half quadratic splitting to optimize this objective function. For our experiments we implemented the GMM in PyTorch and used the Adam [42] optimizer. This is because we are not aware how to efficiently implement the half quadratic splitting for the CT forward operator.

- **Asymmetric CNN.** The asymmetric CNN (ACNN) [80] is a 23-layer CNN trained in a supervised way on 28 paired images pairs of the composite "SiC Diamonds" using the $L^2$ loss function.

- **Zero Shot Super-Resolution.** For Zero Shot Super-Resolution (ZSSR) [75] the main assumption is that the patch distribution of natural images is self-similar across the scales. Exploiting this fact, a lightweight CNN is trained in a supervised fashion on a paired dataset generated by the low-resolution image itself. This dataset is created by downsampling the low-resolution image to obtain a lower-resolution image and is enlarged by data augmentation like random rotations, random crops or mirror reflections. A high-resolution prediction is then created by applying the trained model to the low-resolution observation. For our experiments we reimplemented the ZSSR.

- **DualSR.** The idea of DualSR [21] is a dual-path pipeline, where an upsampling GAN learns the upsampling process and a downsampling GAN learns the degradation model, trained on cropped parts of the low-resolution image. This method can be used for blind superresolution, but since we know the forward operator in our case, we replace the downsampling GAN by the given degradation process.

**F Details on the data fidelity terms**

Here we want to discuss the concrete form of the data fidelity term in (2), which is often derived by a Bayesian approach. In particular, the data fidelity term corresponds to the negative log likelihood $D(f(x), y) = -\log(p_{y|X=x}(y))$.

In Section 5.1 and Section 5.3 we consider the Bayesian inverse problem

$$Y = f(X) + \eta$$

for $\eta \sim \mathcal{N}(0, \sigma^2 I)$ independent from $X$. Consequently, from a Bayesian viewpoint the negative log likelihood $-\log(p_{y|X=x}(y))$ can be, up to a constant, rewritten by

$$-\log(p_{y|X=x}(y)) = -\log \left( \exp(-\|f(x) - y\|^2/(2\sigma^2)) \right) = \frac{1}{2\sigma^2} \|f(x) - y\|^2.$$

Thus the concrete form of (2) is given by

$$J(x) = \frac{1}{2\sigma^2} \|f(x) - y\|^2 + \rho R(x) = \|f(x) - y\|^2 + \lambda R(x),$$

with $\lambda = 2\rho\sigma^2$.

In Section 5.2 we consider the inverse problem (6), with Poisson noise given by

$$\eta = -f(x) - \log \left( \frac{\tilde{N}_1}{N_0} \right), \quad \tilde{N}_1 \sim \text{Pois}(N_0 \exp(-f(x))),$$

\footnote{We use the original implementation available at \url{https://github.com/memad73/DualSR}}
where \( N_0 = 4096 \) is the mean photon count per detector bin without attenuation. Since the observation \( y \) can be rewritten by

\[
y = -\log \left( \frac{\hat{N}_1}{N_0} \right)
\]

and using Jacobi’s transformation formula (see e.g., [35, Theorem 12.7]), the likelihood \( p_{Y|X=x}(y) \) is given by

\[
p_{Y|X=x}(y) = p_{\hat{N}_1}(e^{-y}N_0)N_0 \exp \left( -\sum_{i=1}^{d} y_i \right).
\]

Up to a constant, this results in the negative log likelihood

\[
-\log(p_{Y|X=x}(y)) = \sum_{i=1}^{d} e^{-(f(x)_i)N_0} - e^{-y_i N_0} \left( -(f(x)_i) + \log(N_0) \right)
\]

and finally, the concrete form of (2) is given by

\[
J(x) = \sum_{i=1}^{d} e^{-(f(x)_i)N_0} - e^{-y_i N_0} \left( -(f(x)_i) + \log(N_0) \right) + \lambda R(x).
\]

G Negative log likelihood

There is some evidence that normalizing flows, when trained on full images, fail at out-of-distribution (OOD) detection [45]. Nalisnick et al. [61] present normalizing flow models which are assign way higher likelihoods to unknown, OOD datasets. In this context we test if this behaviour also occurs for our patchNR. In Figure 6 (left) we calculated the likelihood of clean patches and blurred patches from the test set for the patchNR used in Section 5.3. Obviously, the clean patches from the original image have a lower negative log likelihood than patches from blurred images. The further motivates the use of negative log likelihood as a regularization term.

In Figure 6 (right) we compare the negative log likelihood of the GMM with the patchNR as they perform quite similar on the deblurring task for lentils in Section 5.3. Both models assign similar likelihood scores to patches from the test image. This suggests that the patch distribution of the textures is simple enough to be efficiently approximated with both a GMM and a NF.

H Exemplary ground truth images

In Figure 7 two exemplary test images from the respective image class are shown.

Figure 6: Left: Comparison of the negative log likelihood of for clean and blurred patches of “Lentils” from the Kylberg texture dataset [50]. The blurred images were created using the same kernel as in Section 5.3. Right: Negative log likelihood of patches from the “Lentils” test image for the patchNR and the GMM with 200 components and fully learned covariance matrices.
I Details on the architecture of the patchNR

We use 5 GlowCoupling blocks and permutations in an alternating manner, where the coupling blocks are from the freely available FrEIA package\(^{13}\). The 3-layer subnetworks are fully connected with ReLU activation functions and 512 nodes. The patchNR is trained on 6 × 6 patches, i.e., s = 36. For each image class, we trained the patchNR and the cPatchNR using Adam optimizer\(^{42}\) with a learning rate of 0.0001, a batch size of 32 and for 750000 optimizer steps. Training took about 2.5 hours on a single NVIDIA GeForce RTX 2080 super GPU with 8 GB GPU memory.

Superresolution For training we used one high resolution image of size 600 × 600. For reconstruction, we use the regularization parameter \(\lambda = 0.15 \frac{1}{N_p}\), the random subset of overlapping patches is of size \(N_p = 130000\) in each iteration and we optimize over 300 iterations using Adam optimizer with a learning rate of 0.03.

Computerized Tomography In contrast to superresolution, here we trained the patchNR on 6 ground truth images of size 362 × 362, see Figure 8. For both full angle and limited angle CT we used a regularization parameter \(\lambda = 700 \frac{1}{N_p}\), a random subset of \(N_p = 40000\) overlapping patches in each iteration and Adam optimizer with a learning rate of 0.005. While for full angle CT we optimized over 300 iterations, for limited angle CT 3000 iterations are used.

Deblurring The patchNR is trained on one example image of size 500 × 500. For reconstruction, the regularization parameter is set to \(\lambda = 0.87 \frac{1}{N_p}\) and for every iteration step we used a random subset of \(N_p = 40000\) overlapping patches. We optimized for 600 iterations using Adam optimizer with a learning rate of 0.005.

J Ablation study

Here we give a short ablation study comparing the patch size, the regularization parameter and the number of patches extracted in each iteration in order to present the sensitivity of the patchNR regarding the hyperparameter choice. Moreover, we compare the performance of the patchNR for a different training set size and test it on the stability under the choice of training images. This is done for the example of CT.

For the ablation study we focus on the full angle CT setting from Section 5.2. We trained the patchNR for patch sizes 4 × 4, 6 × 6, 8 × 8 and 10 × 10. In Table 6 we tested the sensitivity w.r.t. the patch size. For all patch sizes we extracted 40000 patches per iteration and used the optimal regularization parameter \(\lambda\) (which is set to 1600, 700, 400 and 250 for the patch sizes 4 × 4, 6 × 6, 8 × 8 and

\(^{13}\)available at https://github.com/VLL-HD/FrEIA
10 × 10, respectively). We can observe that a larger patch size lead to slightly more blurry images. However the PSNR seems to change very little within different patch sizes, therefore we observe that our method is quite robust against the choice of the patch size.

In Table 7 we test the sensitivity w.r.t. number of patches used per iteration. Here we consider the patch size 6 × 6 and use the regularization parameter \( \lambda = 700 \). Here we see that PSNR generally increases with number of patches but SSIM and blur effect seem to get worse at some point.

In Table 8 we consider the influence of the regularization parameter for different patch sizes. For all patch sizes under consideration the blurring level increases with a higher regularization parameter.

In addition to this hyperparameters we also explored the choice of the training set. In Table 9 we evaluate the patchNR with patch size 6 × 6 when trained on 1, 6 or 50 images. Obviously, for the CT dataset 1 training image is not enough to learn the patch distribution. This can be explained by the diversity of the CT dataset, see e.g. Figure 8.

Further in Table 10 we varied the training set of 6 images and evaluated the model on 3 different choices. In total we trained the patchNR 15 times on 6 randomly chosen training images of the LoDoPaB dataset and then evaluated on the test set. Again we consider the patch size 6 × 6 and a regularization parameter \( \lambda = 700 \). Note that the bad case in Table 10 comes from the bad training set of the patchNR, see Figure 9. Please zoom in to see the noise in the ground truth images.

Note that the only randomness at test time comes from the stochastic optimization procedure and the random extraction of patches per iteration. Therefore, the reconstruction quality varies only very slightly between two reconstructions of the same test image. For instance, we created 20 reconstructions for the first test image of the full-angle CT experiment and observed a standard-deviation of 0.004 for PSNR, 0.0003 for Blur effect, 0.0005 for LPIPS and 0.0001 for SSIM.

Overall, we see that the method is quite robust towards certain hyperparameter changes, and it can even be a matter of taste which ones to prefer as the image metrics do not always agree.

Lastly, in Figure 10 we compare the PSNR values of patchNR (+) and EPLL (|) reconstructions in the full angle CT case in order to see the dependence of the quality of the reconstruction with respect to the test image. Here we can observe that not only the mean of the patchNR reconstructions is better than the one of EPLL, but also the reconstructions are consistently better.

Table 6: Ablation study. Full angle CT. Patch size with optimal regularization parameter \( \lambda \). Averaged quality measures and standard deviations of the reconstructions. PSNR and SSIM were evaluated on an adaptive data range.

| Patch size | \( s = 4 \times 4 \) | \( s = 6 \times 6 \) | \( s = 8 \times 8 \) | \( s = 10 \times 10 \) |
|------------|-----------------|-----------------|-----------------|-----------------|
| PSNR       | 35.00 ± 4.45    | 35.19 ± 4.52    | 35.20 ± 4.58    | 35.17 ± 4.56    |
| Blur Effect| 0.430 ± 0.045   | 0.428 ± 0.048   | 0.447 ± 0.056   | 0.448 ± 0.057   |
| LPIPS      | 0.194 ± 0.177   | 0.191 ± 0.176   | 0.201 ± 0.178   | 0.201 ± 0.177   |
| SSIM       | 0.825 ± 0.153   | 0.829 ± 0.152   | 0.827 ± 0.154   | 0.828 ± 0.154   |

Table 7: Ablation study. Full angle CT. Patch size \( s = 6 \times 6 \). Averaged quality measures and standard deviations of the reconstructions. PSNR and SSIM were evaluated on an adaptive data range.

| Extracted patches per iteration | 10000 | 20000 | 30000 | 40000 | 50000 | 60000 |
|--------------------------------|-------|-------|-------|-------|-------|-------|
| PSNR                           | 35.04 ± 4.39 | 35.16 ± 4.39 | 35.19 ± 4.52 | 35.21 ± 4.55 | 35.21 ± 4.56 |
| Blur Effect                    | 0.410 ± 0.041 | 0.423 ± 0.047 | 0.428 ± 0.048 | 0.433 ± 0.051 | 0.436 ± 0.051 |
| LPIPS                          | 0.180 ± 0.173 | 0.188 ± 0.176 | 0.191 ± 0.176 | 0.195 ± 0.177 | 0.196 ± 0.178 |
| SSIM                           | 0.829 ± 0.148 | 0.829 ± 0.151 | 0.829 ± 0.152 | 0.828 ± 0.153 | 0.828 ± 0.154 |
Table 8: Ablation study. Full angle CT. 40000 extracted patches per iteration. Averaged quality measures and standard deviations of the reconstructions. PSNR and SSIM were evaluated on an adaptive data range.

| Patch Size = 4 | 1400 | 1500 | 1600 | 1700 | 1800 |
|----------------|------|------|------|------|------|
| PSNR           | 34.95 ± 4.39 | 34.99 ± 4.42 | 35.00 ± 4.45 | 35.01 ± 4.46 | 35.01 ± 4.48 |
| Blur Effect    | 0.422 ± 0.043 | 0.426 ± 0.044 | 0.430 ± 0.045 | 0.433 ± 0.046 | 0.436 ± 0.047 |
| LPIPS          | 0.191 ± 0.176 | 0.193 ± 0.176 | 0.194 ± 0.177 | 0.196 ± 0.177 | 0.197 ± 0.177 |
| SSIM           | 0.826 ± 0.151 | 0.826 ± 0.152 | 0.825 ± 0.153 | 0.825 ± 0.154 | 0.824 ± 0.155 |

| Patch Size = 6 | 500 | 600 | 700 | 800 | 900 |
|----------------|-----|-----|-----|-----|-----|
| PSNR           | 34.97 ± 4.33 | 35.13 ± 4.46 | 35.19 ± 4.52 | 35.21 ± 4.57 | 35.19 ± 4.58 |
| Blur Effect    | 0.408 ± 0.044 | 0.420 ± 0.047 | 0.428 ± 0.048 | 0.436 ± 0.050 | 0.442 ± 0.052 |
| LPIPS          | 0.184 ± 0.174 | 0.188 ± 0.176 | 0.191 ± 0.176 | 0.196 ± 0.177 | 0.199 ± 0.178 |
| SSIM           | 0.829 ± 0.146 | 0.829 ± 0.150 | 0.829 ± 0.152 | 0.827 ± 0.154 | 0.826 ± 0.155 |

| Patch Size = 8 | 300 | 350 | 400 | 450 | 500 |
|----------------|-----|-----|-----|-----|-----|
| PSNR           | 35.12 ± 4.47 | 35.19 ± 4.54 | 35.20 ± 4.58 | 35.18 ± 4.59 | 35.15 ± 4.60 |
| Blur Effect    | 0.428 ± 0.053 | 0.439 ± 0.055 | 0.447 ± 0.056 | 0.454 ± 0.057 | 0.459 ± 0.058 |
| LPIPS          | 0.192 ± 0.176 | 0.197 ± 0.178 | 0.201 ± 0.178 | 0.204 ± 0.178 | 0.207 ± 0.178 |
| SSIM           | 0.829 ± 0.150 | 0.829 ± 0.153 | 0.827 ± 0.154 | 0.826 ± 0.156 | 0.825 ± 0.157 |

| Patch Size = 10 | 150 | 200 | 250 | 300 | 350 |
|-----------------|-----|-----|-----|-----|-----|
| PSNR            | 34.76 ± 4.23 | 35.10 ± 4.47 | 35.17 ± 4.56 | 35.13 ± 4.58 | 35.06 ± 4.58 |
| Blur Effect     | 0.406 ± 0.046 | 0.432 ± 0.053 | 0.448 ± 0.057 | 0.459 ± 0.059 | 0.468 ± 0.060 |
| LPIPS           | 0.187 ± 0.172 | 0.194 ± 0.176 | 0.201 ± 0.177 | 0.207 ± 0.178 | 0.212 ± 0.178 |
| SSIM            | 0.827 ± 0.144 | 0.829 ± 0.150 | 0.828 ± 0.154 | 0.826 ± 0.156 | 0.823 ± 0.157 |

Table 9: Ablation study. Full angle CT. 40000 extracted patches per iteration. Patch size $s = 6 \times 6$. Regularization parameter $\lambda = 700$. Averaged quality measures and standard deviations of the reconstructions. PSNR and SSIM were evaluated on an adaptive data range.

| Number of training images | 1   | 6   | 50  |
|---------------------------|-----|-----|-----|
| PSNR                      | 33.68 ± 3.57 | 35.19 ± 4.32 | 35.24 ± 4.60 |
| Blur Effect               | 0.354 ± 0.024 | 0.428 ± 0.048 | 0.449 ± 0.059 |
| LPIPS                     | 0.161 ± 0.140 | 0.191 ± 0.176 | 0.202 ± 0.180 |
| SSIM                      | 0.802 ± 0.127 | 0.829 ± 0.152 | 0.827 ± 0.156 |

Figure 10: Full angle CT. Comparison of PSNR value of EPLL and patchNR.

K Empirical convergence analysis

To reconstruct the ground truth image from given measurements $y$, we minimize the functional $J(x; y)$ in equation (4) w.r.t. $x$ using the Adam optimizer. The resulting optimization problem is non-convex and the final minimizer could depend on the initialization. In our experiments, it has proven useful to start the optimization with a rough reconstruction. For superresolution we use a bicubic interpolation, for both full angle CT and limited angle CT we choose a FBP reconstruction.
Table 10: Ablation study. Full angle CT. 40000 extracted patches per iteration. Patch size $s = 6 \times 6$. Regularization parameter $\lambda = 700$. 6 random training images. Averaged quality measures and standard deviations of the reconstructions. PSNR and SSIM were evaluated on an adaptive data range.

|          | worst run | our run   | best run  | mean ± standard deviation |
|----------|-----------|-----------|-----------|---------------------------|
| PSNR     | 34.90 ± 4.39 | 35.19 ± 4.52 | 35.26 ± 4.60 | 35.13 ± 0.09            |
| Blur Effect | 0.443 ± 0.058 | 0.428 ± 0.048 | 0.440 ± 0.054 | 0.438 ± 0.012          |
| LPIPS    | 0.201 ± 0.179 | 0.191 ± 0.176 | 0.197 ± 0.180 | 0.195 ± 0.008          |
| SSIM     | 0.825 ± 0.153 | 0.829 ± 0.152 | 0.828 ± 0.154 | 0.827 ± 0.001          |

Figure 11: The PSNR for the first two test images per iteration of the optimization process.

and for deblurring the blurred image as the initialization. In order to empirically test the convergence of $J(x; y)$ we evaluated the PSNR during the optimization process. For this experiment we used the patchNR from Section 5.2 and study the full angle CT case. In Figure 11 we visualize the PSNR per iteration for the first two images of the test dataset and show reconstructions at iteration 0, 150 and 300. It can be seen that the PSNR is steadily rising during the optimization. Arguably for the left image in Figure 11 we could have chosen even more iterations.

L Further examples

Here we give some more examples of our experiments from Section 5; see Figure 12, Figure 13 and Figure 14.
Figure 12: Comparison of different methods for superresolution. Top: full image. Bottom: zoomed-in part.
Figure 13: Full angle reconstruction of the ground truth CT image using different methods. Top: full image. Bottom: zoomed-in part.
Figure 14: Limited angle reconstruction of the ground truth CT image using different methods. Top: full image. Bottom: zoomed-in part.