What's in a Prior? Learned Proximal Networks for Inverse Problems

Zhenghan Fang[*1] **Sam Buchanan**[*2] **Jeremias Sulam**[1]
[1]Mathematical Institute for Data Science, Johns Hopkins University [2]TTIC

Abstract

Proximal operators are ubiquitous in inverse problems, commonly appearing as part of algorithmic strategies to regularize problems that are otherwise ill-posed. Modern deep learning models have been brought to bear for these tasks too, as in the framework of plug-and-play or deep unrolling, where they loosely resemble proximal operators. Yet, something essential is lost in employing these purely data-driven approaches: there is no guarantee that a general deep network represents the proximal operator of any function, nor is there any characterization of the function for which the network might provide some approximate proximal. This not only makes guaranteeing convergence of iterative schemes challenging but, more fundamentally, complicates the analysis of what has been learned by these networks about their training data. Herein we provide a framework to develop *learned proximal networks* (LPN), prove that they provide exact proximal operators for a data-driven nonconvex regularizer, and show how a new training strategy, dubbed *proximal matching*, provably promotes the recovery of the log-prior of the true data distribution. Such LPN provide general, unsupervised, expressive proximal operators that can be used for general inverse problems with convergence guarantees. We illustrate our results in a series of cases of increasing complexity, demonstrating that these models not only result in state-of-the-art performance, but provide a window into the resulting priors learned from data.

1 Introduction

Inverse problems involve estimating some underlying variables that have undergone a degradation process, such as in denoising, deblurring, inpainting, or compressed sensing [14, 75]. While these problems are naturally ill-posed, solutions to any of these problems involve, either implicitly or explicitly, the utilization of *priors*, or models, about what type of solutions are preferable [34, 13, 6]. Traditional methods model this prior distribution directly, by constructing functions (or regularization terms) that promote specific properties in the resulting estimate, such as for it to be smooth [101], piece-wise smooth [81, 18], or for it to have a sparse decomposition under a given basis or even a potentially overcomplete dictionary [19, 87]. On the other hand, from a machine learning perspective, the complete restoration mapping has also been modeled by a regression function, typically by providing a large collection of input-output (or clean-corrupted) pairs of samples [68, 75, 112].

An interesting third alternative combines these two approaches by making the insightful observation that for almost any inverse problem, a proximal step for the regularization function is needed. Such a sub-problem can be loosely interpreted as a denoising step and, as a result, off-the-shelf and very strong-performing denoising algorithms (such as those given by modern deep learning methods) can be employed as a subroutine. The Plug-and-Play (PnP) framework is one such example of this idea [103, 108, 70, 110, 52, 93], but others exist as well [80, 79]. While these alternatives work very well in practice, little is known about the approximation properties of these methods. For instance, *do these denoising networks actually (i.e., provably) provide a proximal operator for some regularization function?* Moreover, and from a variational perspective, *would this regularization*

[*]Equal contribution.
[†]Emails: {zfang23,jsulam1}@jhu.edu, sam@ttic.edu.

Deep Learning and Inverse Problems Workshop at NeurIPS 2023.

function recover the correct regularizer, such as the (log) prior of the data distribution? While some answers exist [48, 65, 26, 113, 41], they rely on generally restrictive settings (see a thorough discussion of related works in Appendix A). More broadly, the ability to characterize a data-driven (potentially nonconvex) regularizer that enables good restoration is paramount in applications that demand notions of robustness and interpretability, and this remains an open challenge.

In this work, we address these questions by proposing a new class of deep neural networks, termed *learned proximal networks* (LPN), that *exactly implement the proximal operator* of a general learned function. Such a LPN implicitly, but exactly, learns a regularization function that can be characterized and evaluated, shedding light onto what has been learned from data. In turn, we present a new training problem, which we dub *proximal matching*, that provably promotes the recovery of the correct regularization term (i.e., the log of the data distribution), which need not be convex. Moreover, the ability of LPNs to implement exact proximal operators allows for guaranteed convergence to critical points of the variational problem, which we derive for a representative PnP reconstruction algorithm under no additional assumptions on the trained LPN. We demonstrate through experiments on synthetic data and hand-written digits that our LPNs can recover the correct underlying data distribution, and further show that LPNs lead to state-of-the-art reconstruction performance on tasks such as image deblurring, CT reconstruction and compressed sensing, while enabling precise characterization of the data-dependent prior learned by the model.

2 Background

Consider an unknown signal in an Euclidean space, $\mathbf{x} \in \mathbb{R}^n$, and a known measurement operator that maps to an output space, $A : \mathbb{R}^n \to \mathbb{R}^m$. The goal of inverse problems is to recover $\mathbf{x}$ from its noisy observation $\mathbf{y} = A(\mathbf{x}) + \mathbf{v} \in \mathbb{R}^m$, where $\mathbf{v}$ is a noise or nuisance term. A prior is typically needed to regularize the problem, which can generally take the form $\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - A(\mathbf{x})\|_2^2 + \phi(\mathbf{x})$, for a function $\phi(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ promoting a solution that is likely under the prior distribution of $\mathbf{x}$.

Proximal operators For a given functional ϕ as above, its proximal operator prox_ϕ is defined by [72, 11] $\mathrm{prox}_\phi(\mathbf{y}) := \mathrm{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|^2 + \phi(\mathbf{x})$. The continuous proximal of a (potentially nonconvex) function can be fully characterized as the gradient of a convex function (illustrated in Figure 1).

Proposition 1. *[Characterization of continuous proximal operators, [44, Corollary 1]] Let $\mathcal{Y} \subset \mathbb{R}^n$ be non-empty and open and $f : \mathcal{Y} \to \mathbb{R}^n$ be a continuous function. Then, f is a proximal operator of a function $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ if and only if there exists a convex differentiable function ψ such that $f(\mathbf{y}) = \nabla\psi(\mathbf{y})$ for each $\mathbf{y} \in \mathcal{Y}$.*

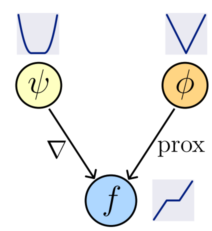


Figure 1: Illustration of Prop. 1 with $\phi(\cdot) = \|\cdot\|_1$.

Plug-and-Play The Plug-and-Play (PnP) framework employs off-the-shelf denoising algorithms to solve general inverse problems within an ADMM approach [16]. PnP replaces the explicit solution of prox_ϕ with generic denoising algorithms, such as BM3D [29, 103] or CNN-based denoisers [70, 108, 110, 52, 107, 106, 100], bringing the benefits of advanced denoisers to general inverse problems. While useful in practice, such denoisers are *not* in general proximal operators. Although PnP has achieved impressive results with deep learning based denoisers, little is known about the implicit prior–if any–encoded in these denoisers, thus diminishing the interpretability of the reconstruction results. Furthermore, although certain convergence guarantees have been derived for PnP with MMSE denoisers [105], it chiefly relies on the assumption that the denoiser is non-expansive (which can be hard to verify or enforce in practice). See Appendix A for a more comprehensive review of related works.

3 Learned Proximal Networks

First, we present a way to parameterize a neural network such that its mapping is guaranteed to be the proximal operator of some (potentially nonconvex) scalar-valued functional. Motivated by Proposition 1, we parameterize *gradients of convex functions* by differentiating a neural network that implements a convex function, which can be implemented by an input convex neural network (ICNN) [4]. Consider a single-layer neural network characterized by the weights $\mathbf{W} \in \mathbb{R}^{m\times n}$, bias term $\mathbf{b} \in \mathbb{R}^m$ and a scalar non-linearity $g : \mathbb{R} \to \mathbb{R}$. Such a network, at a sample $\mathbf{x}$, is given by $\mathbf{z} = g(\mathbf{W}\mathbf{x} + \mathbf{b})$. With this notation, we now define our **Learned Proximal Networks (LPN)**.

Proposition 2 (Learned Proximal Networks)**.** *Consider a scalar-valued $(K + 1)$-layered neural network $\psi_\theta : \mathbb{R}^n \to \mathbb{R}$ defined by $\psi_\theta(\mathbf{x}) = \mathbf{w}^T\mathbf{z}_K + b$ and the recursion*

$$\mathbf{z}_1 = g(\mathbf{H}_1\mathbf{x} + \mathbf{b}_1), \qquad \mathbf{z}_k = g(\mathbf{W}_k\mathbf{z}_{k-1} + \mathbf{H}_k\mathbf{x} + \mathbf{b}_k),\ k \in [2, K]$$

where $\theta = \{\mathbf{w}, b, (\mathbf{W}_k)_{k=2}^K, (\mathbf{H}_k, \mathbf{b}_k)_{k=1}^K\}$ *are learnable parameters, and* g *is a convex, non-decreasing and* C^2 *scalar function. Assume that all entries of* $\mathbf{W}_k$ *and* $\mathbf{w}$ *are non-negative, and let* f_θ *be the gradient map of* ψ_θ *w.r.t. its input, i.e.* $f_\theta = \nabla_{\mathbf{x}}\psi$*. Then, there exists a function* $\phi_\theta : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *such that* $f_\theta(\mathbf{x}) = \mathrm{prox}_{\phi_\theta}(\mathbf{x})$*,* $\forall\, \mathbf{x} \in \mathbb{R}^n$*.*

Recovering the prior from its proximal Once an LPN f_θ is obtained, we would like to recover its "primitive" function, ϕ_θ, since this function is precisely the regularizer used in the variational objective, $\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - A(\mathbf{x})\|_2^2 + \phi_\theta(\mathbf{x})$. To start with, Gribonval and Nikolova [44] characterize $\phi_\theta(f_\theta(\mathbf{y}))$ as a function of f_θ and ψ_θ. However, to evaluate the prior $\phi_\theta(\mathbf{x})$ at an arbitrary point $\mathbf{x}$, we must invert f_θ, i.e. find $\mathbf{y}$ such that $f_\theta(\mathbf{y}) = \mathbf{x}$. To achieve this, inspired by [46], we add a quadratic term to ψ_θ, $\psi_\theta(\mathbf{x}; \alpha) = \psi_\theta(\mathbf{x}) + \frac{\alpha}{2}\|\mathbf{x}\|_2^2$, with $\alpha \in \mathbb{R}^+$, turning ψ_θ strongly convex – and its gradient map, $f_\theta = \nabla\psi_\theta$, invertible and bijective. We then compute this inverse by minimizing the *convex* objective $\min_{\mathbf{y}} \psi_\theta(\mathbf{y}) - \langle \mathbf{x}, \mathbf{y} \rangle$.

Training LPNs via proximal matching To solve inverse problems efficiently, it is crucial that LPN learns the prox of the correct data prior, i.e., the prox of negative log likelihood $\mathrm{prox}_{-\log p_{\mathbf{x}}}$. Unfortunately, the prior distributions of real-world data are typically unknown, making supervised training infeasible. Thus, we propose a way to learn this from *only i.i.d. samples from the unknown data distribution*. We train LPN to perform *denoising* by minimizing $\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[d(f_\theta(\mathbf{y}), \mathbf{x})\right]$, where d is a distance function, and $\mathbf{y} = \mathbf{x} + \sigma\mathbf{v}$ with $\mathbf{v} \sim \mathcal{N}(0, \mathbf{I})$. Unfortunately, popular choices for d, e.g. the squared ℓ^2, ℓ^1 or LPIPS [111] loss, do not lead to the desired proximal operator of the log prior, since they do not yield the MAP estimate (see an example in Figure 2). We thus propose a new loss function that promotes the recovery of the true proximal operator, termed **proximal matching loss**:

$$\mathcal{L}_{PM}(\theta; \gamma) = \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}}\left[\ell_\gamma(\|f_\theta(\mathbf{y}) - \mathbf{x}\|_2)\right], \quad \ell_\gamma(x) = 1 - \frac{1}{(\pi\gamma^2)^{n/2}}\exp\left(-\frac{x^2}{\gamma^2}\right), \gamma > 0, \tag{3.1}$$

where n is the dimension of $\mathbf{x}$. We show that, given sufficient samples and network capacity, minimizing $\mathcal{L}_{PM}$ yields the true proximal operator almost surely as $\gamma \searrow 0$.

Theorem 3.1 (Learning via Proximal Matching)**.** *Consider a signal* $\mathbf{x} \sim p_{\mathbf{x}}$*, where* $\mathbf{x}$ *is bounded and* $p_{\mathbf{x}}$ *is a continuous density, and a noisy observation* $\mathbf{y} = \mathbf{x} + \sigma\mathbf{v}$*, where* $\mathbf{v} \sim \mathcal{N}(0, \mathbf{I})$ *and* $\sigma > 0$*. Let* $\ell_\gamma(x) : \mathbb{R} \to \mathbb{R}$ *be defined as in* (3.1)*. Consider the optimization problem* $f^* = \mathrm{argmin}_{f \text{ measurable}} \lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\ell_\gamma\left(\|f(\mathbf{y}) - \mathbf{x}\|_2\right)\right]$*. Then, almost surely (i.e., for almost all* $\mathbf{y}$*),* $f^*(\mathbf{y}) = \mathrm{argmax}_{\mathbf{c}}\, p_{\mathbf{x}|\mathbf{y}}(\mathbf{c}) \triangleq \mathrm{prox}_{-\sigma^2 \log p_{\mathbf{x}}}(\mathbf{y})$*.*

Solving Inverse Problems with LPN Once trained, LPN can be used to solve inverse problems with the PnP framework (see Algorithm 3, Appendix D.3 for using LPN with PnP-PGD). We show that, guaranteeing the employed denoiser is indeed a proximal operator enables convergence guarantees without stringent conditions, such as nonexpansivity or enforcing the denoiser to take a restrictive form, in contrast to previous PnP schemes [83, 91, 92, 26, 27, 96, 47, 48, 105, 45, 85, 98].

Theorem 3.2. *Consider the sequence of iterates* $\mathbf{x}_k$*,* $k \in \{0, 1, \dots\}$*, defined by Algorithm 3 run with a linear measurement operator* $\mathbf{A}$ *and a LPN* f_θ *with softplus activations, trained with* $0 < \alpha < 1$*. Assume that the step size satisfies* $0 < \eta < 1/\|\mathbf{A}^T\mathbf{A}\|$*. Then, the iterates* $\mathbf{x}_k$ *converge to a fixed point* $\mathbf{x}^*$ *of Algorithm 3: that is, there exists* $\mathbf{x}^* \in \mathbb{R}^n$ *such that* $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}^*$*, and* $f_\theta\left(\mathbf{x}^* - \eta\nabla h(\mathbf{x}^*)\right) = \mathbf{x}^*$*.*

4 Experiments

Learning soft-thresholding from Laplacian distribution We first train LPN on i.i.d. samples from the 1-D Laplacian distribution $p(x) = \frac{1}{2}\exp\left(-|x|\right)$. The negative log likelihood (NLL) is the ℓ_1 norm, $-\log p(x) = |x| - \log(\frac{1}{2})$, and its proximal operator is the soft-thresholding function $\mathrm{prox}_{-\log p}(x) = \mathrm{sign}(x)\max(|x| - 1, 0)$. As visualized in Figure 2, when using either the ℓ^2 or ℓ^1 loss, the learned prox differs from the correct

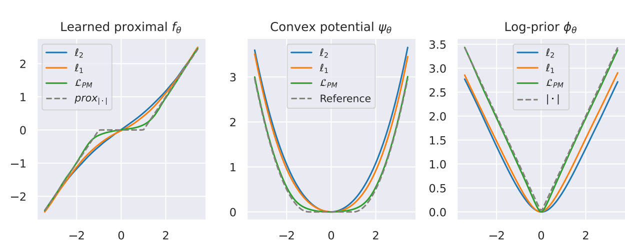
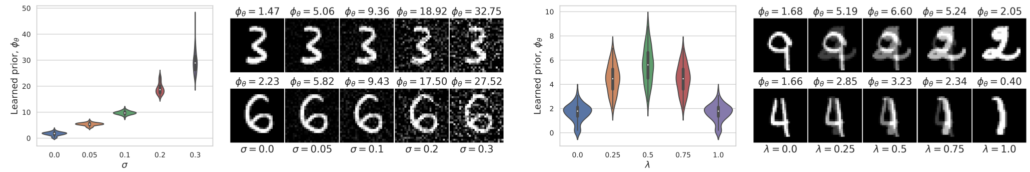


Figure 2: The proximal operator f_θ, convex potential ψ_θ, and log-prior ϕ_θ learned by LPN via different losses: the squared ℓ_2 loss, ℓ_1 loss, and the proposed proximal matching loss $\mathcal{L}_{PM}$. The ground-truth data distribution is the Laplacian $p(x) = \frac{1}{2}\exp(-|x|)$, with log-prior $-\log p(x) = |x| - \log(\frac{1}{2})$. The gray dashed line shows the ground-truth for each case.



(a) Adding Gaussian noise with standard deviation σ. **(b)** Convex combination of two images $(1-\lambda)\mathbf{x}+\lambda\mathbf{y}$.

Figure 3: The log-prior ϕ_θ learned by LPN on MNIST, evaluated at images with various perturbation types and to different degrees: (a) additive Gaussian noise and (b) convex combination between two images. Violin plots show the learned priors evaluated over 100 test images.

soft-thresholding function, verifying our analysis in Section 3. When we switch to the proximal matching loss $\mathcal{L}_{PM}$, the learned proximal matches the soft-thresholding function, corroborating our theoretical analysis in Theorem 3.1 and showcasing the importance of proximal matching loss.

Learning a prior for MNIST Next, we train an LPN on MNIST [59] and evaluate the obtained prior on a series of inputs with different types and degrees of perturbations in order to gauge how such modifications to the data are reflected by the learned log-prior. Figure 3a visualizes the change of prior ϕ_θ after adding increasing levels of Gaussian noise. As expected, as the noise level increases, the values reported by the log prior also increases, reflecting that they are less likely according to the true distribution of the real images. We also present a study that depicts the non-convexity of the learned log prior in Figure 3b. This is natural, since the convex combination of two images no longer resembles a natural image, hence the true prior should indeed be nonconvex. LPN can correctly learn such *nonconvexity* in the prior, while existing approaches using convex priors, either hand-crafted [101, 81, 66, 12, 33, 20] or data-driven [74, 26], are suboptimal by not faithfully capturing the true prior.

Table 1: Results for inverse problems.

METHOD	PSNR (↑)	SSIM (↑)
Tomographic reconstruction		
FBP	21.29	.203
Operator-agnostic		
AR [65]	33.48	.890
Ours	**34.14**	**.891**
Operator-specific		
UAR [74]	**34.76**	**.897**
Compressed sensing (compression rate $= 1/16$**)**		
Sparsity (Wavelet)	26.54	.666
AR [65]	29.71	.712
Ours	**38.03**	**.919**
Compressed sensing (compression rate $= 1/4$**)**		
Sparsity (Wavelet)	36.80	.921
AR [65]	37.94	.920
Ours	**44.05**	**.973**

Solving inverse problems with LPN We showcase the capability of LPN for two realistic inverse problems: sparse-view CT reconstruction and compressed sensing, on the public Mayo-CT dataset [69] of Computed Tomography images. For sparse-view CT, as shown in Table 1 and Figure 6a (Appendix G.3), our method significantly improves over the baseline FBP [104], outperforms the task-agnostic counterpart AR [65], and performs just slightly worse than the task-specific approach UAR [74] – without even having had access to the used forward operator. Figure 6b (Appendix G.3) and Table 1 depict the compressed sensing results, where LPN significantly outperforms the Wavelet baseline and AR, demonstrating much better generalizability to different forward operators. Additionally, we experiment with deblurring on CelebA [63] and include the results in Appendix G.4.

5 Conclusion

The learned proximal networks presented in this paper form a class of neural networks that guarantees to parameterize proximal operators. We showed how the "primitive" function of the proximal operator parameterized by an LPN can be recovered, allowing explicit characterization of the prior learned from data. Furthermore, via proximal matching, LPN can learn the true prox of the log-prior of an unknown distribution from only i.i.d. samples. When used to solve general inverse problems, LPN achieves state-of-the-art results while providing more interpretability by explicit characterization of the (nonconvex) prior, with convergence guarantees. The ability to not only provide unsupervised models for general inverse problems but, chiefly, to characterize the priors learned from data open exciting new research questions of uncertainty quantification [5, 97, 90], sampling [36, 25, 54, 53, 51, 36], equivariant learning [23, 21, 22], learning without ground-truth [95, 94, 38], and robustness [49, 30], all of which constitute matter of ongoing work.

References

[1] Amir Adler, Yacov Hel-Or, and Michael Elad. A shrinkage learning approach for single image super-resolution with overcomplete representations. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*, pages 622–635. Springer, 2010.

[2] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[3] Hemant K Aggarwal, Merry P Mani, and Mathews Jacob. Modl: Model-based deep learning architecture for inverse problems. *IEEE transactions on medical imaging*, 38(2):394–405, 2018.

[4] Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 146–155. PMLR, 2017.

[5] Anastasios N Angelopoulos, Amit Pal Kohli, Stephen Bates, Michael Jordan, Jitendra Malik, Thayer Alshaabi, Srigokul Upadhyayula, and Yaniv Romano. Image-to-image regression with distribution-free uncertainty quantification and applications in imaging. In *International Conference on Machine Learning*, pages 717–730. PMLR, 2022.

[6] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[7] Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-ŁOjasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, May 2010. ISSN 0364-765X. doi: 10.1287/moor.1100.0449.

[8] Hédy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming. A Publication of the Mathematical Programming Society*, 137(1):91–129, February 2013. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-011-0484-9.

[9] Thilo Balke, Fernando Davis Rivera, Cristina Garcia-Cardona, Soumendu Majee, Michael Thompson McCann, Luke Pfister, and Brendt Egon Wohlberg. Scientific computational imaging code (scico). *Journal of Open Source Software*, 7(LA-UR-22-28555), 2022.

[10] Heinz H Bauschke, Sarah M Moffat, and Xianfu Wang. Firmly nonexpansive mappings and maximally monotone operators: Correspondence and duality. *Set-Valued and Variational Analysis*, 20(1):131–153, March 2012. ISSN 1877-0533, 1877-0541. doi: 10.1007/s11228-011-0187-7.

[11] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[12] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[13] Martin Benning and Martin Burger. Modern regularization methods for inverse problems. *Acta numerica*, 27:1–111, 2018.

[14] Mario Bertero, Patrizia Boccacci, and Christine De Mol. *Introduction to inverse problems in imaging*. CRC press, 2021.

[15] Radu Ioan Boţ, Ernö Robert Csetnek, and Szilárd Csaba László. An inertial forward–backward algorithm for the minimization of the sum of two nonconvex functions. *EURO Journal on Computational Optimization*, 4(1):3–25, February 2016. ISSN 2192-4414. doi: 10.1007/s13675-015-0045-8.

[16] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[17] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[18] Kristian Bredies, Karl Kunisch, and Thomas Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

[19] Alfred M Bruckstein, David L Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.

[20] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40:120–145, 2011.

[21] Dongdong Chen, Julián Tachella, and Mike E Davies. Equivariant imaging: Learning beyond the range space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4379–4388, 2021.

[22] Dongdong Chen, Julián Tachella, and Mike E Davies. Robust equivariant imaging: a fully unsupervised framework for learning to image from noisy and partial measurements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5647–5656, 2022.

[23] Dongdong Chen, Mike Davies, Matthias J Ehrhardt, Carola-Bibiane Schönlieb, Ferdia Sherry, and Julián Tachella. Imaging with equivariant deep learning: From unrolled network design to fully unsupervised learning. *IEEE Signal Processing Magazine*, 40(1):134–147, 2023.

[24] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Zhangyang Wang, Howard Heaton, Jialin Liu, and Wotao Yin. Learning to optimize: A primer and a benchmark. *The Journal of Machine Learning Research*, 23(1):8562–8620, 2022.

[25] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[26] Regev Cohen, Yochai Blau, Daniel Freedman, and Ehud Rivlin. It has potential: Gradient-driven denoisers for convergent solutions to inverse problems. *Advances in Neural Information Processing Systems*, 34:18152–18164, 2021.

[27] Regev Cohen, Michael Elad, and Peyman Milanfar. Regularization by denoising via Fixed-Point projection (RED-PRO). *SIAM journal on imaging sciences*, 14(3):1374–1406, January 2021. doi: 10.1137/20M1337168.

[28] Michel Coste. *An Introduction to Semialgebraic Geometry*. Istituti editoriali e poligrafici internazionali, 2000. ISBN 9788881472253.

[29] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[30] Mohammad Zalbagi Darestani, Akshay S Chaudhari, and Reinhard Heckel. Measuring robustness in deep learning based compressive sensing. In *International Conference on Machine Learning*, pages 2433–2444. PMLR, 2021.

[31] Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. March 2023.

[32] Albrecht Dold. *Lectures on Algebraic Topology*. Springer Science & Business Media, December 2012. ISBN 9783642678219.

[33] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[34] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[35] Zhenghan Fang, Kuo-Wei Lai, Peter van Zijl, Xu Li, and Jeremias Sulam. Deepsti: Towards tensor reconstruction using fewer orientations in susceptibility tensor imaging. *Medical image analysis*, 87:102829, 2023.

[36] Berthy T Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L Bouman, and William T Freeman. Score-based diffusion models as principled priors for inverse imaging. *arXiv preprint arXiv:2304.11751*, 2023.

[37] Pierre Frankel, Guillaume Garrigos, and Juan Peypouquet. Splitting methods with variable metric for Kurdyka–Łojasiewicz functions and general convergence rates. *Journal of optimization theory and applications*, 165(3):874–900, 2015. ISSN 0022-3239. doi: 10.1007/s10957-014-0642-3.

[38] Angela F Gao, Oscar Leong, He Sun, and Katherine L Bouman. Image reconstruction without explicit priors. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[39] Davis Gilton, Greg Ongie, and Rebecca Willett. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 6:328–343, 2019.

[40] Davis Gilton, Gregory Ongie, and Rebecca Willett. Deep equilibrium architectures for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:1123–1133, 2021.

[41] Alexis Goujon, Sebastian Neumayer, and Michael Unser. Learning weakly convex regularizers for convergent Image-Reconstruction algorithms. August 2023.

[42] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

[43] Rémi Gribonval. Should penalized least squares regression be interpreted as maximum a posteriori estimation? *IEEE transactions on signal processing: a publication of the IEEE Signal Processing Society*, 59(5):2405–2410, May 2011. ISSN 1053-587X, 1941-0476. doi: 10.1109/TSP.2011.2107908.

[44] Rémi Gribonval and Mila Nikolova. A characterization of proximity operators. *Journal of Mathematical Imaging and Vision*, 62(6-7):773–789, 2020.

[45] Andreas Hauptmann, Subhadip Mukherjee, Carola-Bibiane Schönlieb, and Ferdia Sherry. Convergent regularization in inverse problems and linear plug-and-play denoisers. July 2023.

[46] Chin-Wei Huang, Ricky T Q Chen, Christos Tsirigotis, and Aaron Courville. Convex potential flows: Universal probability distributions with optimal transport and convex optimization. In *International Conference on Learning Representations*, 2021.

[47] Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Gradient step denoiser for convergent plug-and-play. In *International Conference on Learning Representations*, 2022.

[48] Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. In *International Conference on Machine Learning*, pages 9483–9505. PMLR, 2022.

[49] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. Robust compressed sensing mri with deep generative priors. *Advances in Neural Information Processing Systems*, 34:14938–14954, 2021.

[50] Ajil Jalal, Sushrut Karmalkar, Alexandros G Dimakis, and Eric Price. Instance-optimal compressed sensing via posterior sampling. *arXiv preprint arXiv:2106.11438*, 2021.

[51] Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Adv. Neural Inf. Process. Syst.*, 34:13242–13254, 2021.

[52] Ulugbek S Kamilov, Charles A Bouman, Gregery T Buzzard, and Brendt Wohlberg. Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 40(1):85–97, 2023.

[53] Bahjat Kawar, Gregory Vaksman, and Michael Elad. SNIPS: Solving noisy inverse problems stochastically. May 2021.

[54] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. January 2022.

[55] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[56] Erich Kobler, Teresa Klatzer, Kerstin Hammernik, and Thomas Pock. Variational networks: Connecting variational methods and deep learning. In *Pattern Recognition*, Lecture Notes in Computer Science, pages 281–293. Springer, Cham, September 2017. ISBN 9783319667089, 9783319667096. doi: 10.1007/978-3-319-66709-6_23.

[57] Erich Kobler, Alexander Effland, Karl Kunisch, and Thomas 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.

[58] Kuo-Wei Lai, Manisha Aggarwal, Peter van Zijl, Xu Li, and Jeremias Sulam. Learned proximal networks for quantitative susceptibility mapping. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*, pages 125–135. Springer, 2020.

[59] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[60] Housen Li, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. Nett: Solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005, 2020.

[61] Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. ALISTA: Analytic weights are as good as learned weights in LISTA. In *International Conference on Learning Representations*, 2019.

[62] Jiaming Liu, Xiaojian Xu, Weijie Gan, Ulugbek Kamilov, et al. Online deep equilibrium learning for regularization by denoising. *Advances in Neural Information Processing Systems*, 35:25363–25376, 2022.

[63] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

[64] Stanislaw Lojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. *Les équations aux dérivées partielles*, 117:87–89, 1963.

[65] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Adversarial regularizers in inverse problems. *Advances in neural information processing systems*, 31, 2018.

[66] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[67] Morteza Mardani, Qingyun Sun, David Donoho, Vardan Papyan, Hatef Monajemi, Shreyas Vasanawala, and John Pauly. Neural proximal gradient descent for compressive imaging. *Advances in Neural Information Processing Systems*, 31, 2018.

[68] Michael T McCann, Kyong Hwan Jin, and Michael Unser. Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, 34(6):85–95, 2017.

[69] C McCollough. Tu-fg-207a-04: overview of the low dose ct grand challenge. *Medical physics*, 43(6Part35):3759–3760, 2016.

[70] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1781–1790, 2017.

[71] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2): 18–44, March 2021. ISSN 1558-0792. doi: 10.1109/MSP.2020.3016905.

[72] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

[73] Subhadip Mukherjee, Sören Dittmer, Zakhar Shumaylov, Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Learned convex regularizers for inverse problems. *arXiv preprint arXiv:2008.02839*, 2020.

[74] Subhadip Mukherjee, Marcello Carioni, Ozan Öktem, and Carola-Bibiane Schönlieb. End-to-end reconstruction meets data-driven regularization for inverse problems. *Advances in Neural Information Processing Systems*, 34:21413–21425, 2021.

[75] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

[76] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014. ISSN 2167-3888. doi: 10.1561/2400000003.

[77] Jack Richter-Powell, Jonathan Lorraine, and Brandon Amos. Input convex gradient networks. *arXiv preprint arXiv:2111.12187*, 2021.

[78] R Tyrell Rockafellar and Roger J-B Wets. *Variational Analysis*. Grundlehren der mathematischen Wissenschaften. Springer-Verlag Berlin Heidelberg, 1 edition, 1998. ISBN 9783642024313, 9783540627722. doi: 10.1007/978-3-642-02431-3.

[79] Yaniv Romano and Michael Elad. Boosting of image denoising algorithms. *SIAM Journal on Imaging Sciences*, 8(2):1187–1219, 2015.

[80] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

[81] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[82] Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill, New York, 3 edition, 1976. ISBN 9780070542358.

[83] Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-Play methods provably converge with properly trained denoisers. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5546–5557. PMLR, 2019.

[84] Vineet R Shenoy, Tim K Marks, Hassan Mansour, and Suhas Lohit. Unrolled ippg: Video heart rate estimation via unrolling proximal gradient descent. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2715–2719. IEEE, 2023.

[85] Suhas Sreehari, S V Venkatakrishnan, Brendt Wohlberg, Gregery T Buzzard, Lawrence F Drummy, Jeffrey P Simmons, and Charles A Bouman. Plug-and-Play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, 2(4):408–423, December 2016. ISSN 2333-9403. doi: 10.1109/TCI.2016.2599778.

[86] Elias M Stein and Rami Shakarchi. *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*. Princeton University Press, April 2005. ISBN 9780691113869.

[87] Jeremias Sulam, Boaz Ophir, and Michael Elad. Image denoising through multi-scale learnt dictionaries. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 808–812. IEEE, 2014.

[88] Jeremias Sulam, Aviad Aberdam, Amir Beck, and Michael Elad. On multi-layer basis pursuit, efficient algorithms and convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(8):1968–1980, 2019.

[89] Jeremias Sulam, Ramchandran Muthukumar, and Raman Arora. Adversarial robustness of supervised sparse coding. *Advances in neural information processing systems*, 33:2110–2121, 2020.

[90] He Sun and Katherine L Bouman. Deep probabilistic imaging: Uncertainty quantification and multi-modal solution characterization for computational imaging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2628–2637, 2021.

[91] Yu Sun, Brendt Wohlberg, and Ulugbek S Kamilov. An online Plug-and-Play algorithm for regularized image reconstruction. *IEEE Transactions on Computational Imaging*, 5(3):395–408, September 2019. ISSN 2333-9403. doi: 10.1109/TCI.2019.2893568.

[92] Yu Sun, Zihui Wu, Xiaojian Xu, Brendt Wohlberg, and Ulugbek S Kamilov. Scalable Plug-and-Play ADMM with convergence guarantees. *IEEE Transactions on Computational Imaging*, 7:849–863, 2021. ISSN 2333-9403. doi: 10.1109/TCI.2021.3094062.

[93] Julián Tachella, Yoann Altmann, Nicolas Mellado, Aongus McCarthy, Rachael Tobin, Gerald S Buller, Jean-Yves Tourneret, and Stephen McLaughlin. Real-time 3d reconstruction from single-photon lidar data using plug-and-play point cloud denoisers. *Nature communications*, 10(1):4984, 2019.

[94] Julián Tachella, Dongdong Chen, and Mike Davies. Unsupervised learning from incomplete measurements for inverse problems. *Advances in Neural Information Processing Systems*, 35:4983–4995, 2022.

[95] Julián Tachella, Dongdong Chen, and Mike Davies. Sensing theorems for unsupervised learning in linear inverse problems. *Journal of Machine Learning Research*, 24(39):1–45, 2023.

[96] Hong Ye Tan, Subhadip Mukherjee, Junqi Tang, and Carola-Bibiane Schönlieb. Provably convergent Plug-and-Play Quasi-Newton methods. March 2023.

[97] Jacopo Teneggi, Matthew Tivnan, Web Stayman, and Jeremias Sulam. How to trust your diffusion model: A convex optimization approach to conformal risk control. In *International Conference on Machine Learning*, pages 33940–33960. PMLR, 2023.

[98] Afonso M Teodoro, Jose M Bioucas-Dias, and Mario A T Figueiredo. A convergent image fusion algorithm using Scene-Adapted Gaussian-Mixture-Based denoising. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, September 2018. ISSN 1057-7149, 1941-0042. doi: 10.1109/TIP.2018.2869727.

[99] Andreas Themelis and Panagiotis Patrinos. Douglas–Rachford splitting and ADMM for nonconvex optimization: Tight convergence results. *SIAM journal on optimization: a publication of the Society for Industrial and Applied Mathematics*, 30(1):149–181, January 2020. ISSN 1052-6234. doi: 10.1137/18M1163993.

[100] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 131:251–275, 2020.

[101] Andrey N Tikhonov and Vasiliy Y Arsenin. Solutions of ill-posed problems. vh winston & sons, 1977.

[102] Bahareh Tolooshams, Satish Mulleti, Demba Ba, and Yonina C Eldar. Unrolled compressed blind-deconvolution. *IEEE Transactions on Signal Processing*, 2023.

[103] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

[104] Martin J Willemink and Peter B Noël. The evolution of image reconstruction for ct—from filtered back projection to artificial intelligence. *European radiology*, 29:2185–2195, 2019.

[105] Xiaojian Xu, Yu Sun, Jiaming Liu, Brendt Wohlberg, and Ulugbek S Kamilov. Provable convergence of Plug-and-Play priors with MMSE denoisers. *IEEE Signal Processing Letters*, 27:1280–1284, 2020. ISSN 1558-2361. doi: 10.1109/LSP.2020.3006390.

[106] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[107] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[108] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

[109] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3217–3226, 2020.

[110] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2021.

[111] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver 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.

[112] Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.

[113] Zihao Zou, Jiaming Liu, Brendt Wohlberg, and Ulugbek S Kamilov. Deep equilibrium learning of explicit regularizers for imaging inverse problems. *arXiv preprint arXiv:2303.05386*, 2023.

A Related Works

Deep Unrolling In addition to Plug-and-Play, deep unrolling is another approach using deep neural networks to replace proximal operators for solving inverse problems. Similar to PnP, the deep unrolling model is parameterized by an unrolled iterative algorithm, with certain (proximal) steps replaced by deep neural nets. In contrast to PnP, the unrolling model is trained in an end-to-end fashion by paired data of ground truth and corresponding measurements from specific forward operators. Truncated deep unrolling methods unfold the algorithm for a fixed number of steps [42, 1, 61, 3, 2, 109, 71, 39, 102, 56, 24, 67, 88], while infinite-step models have been recently developed based on deep equilibrium learning [40, 62, 113]. In future work, LPN can improve the performance and interpretability of deep unrolling methods in e.g., medical applications [58, 35, 84] or in cases that demand the analysis of robustness [89]. The end-to-end supervision in unrolling can also help increase the performance of LPN-based methods for inverse problems in general.

Explicit Regularizer A series of works have been dedicated to designing explicit data-driven regularizer for inverse problems, such as RED [80], AR [65], ACR [73], UAR [74] and others [60, 57, 26, 113, 41]. Our work contributes a new angle to this field, by learning a proximal operator for the log-prior and recovering the prior from the learned proximal.

Gradient Denoiser Gradient step (GS) denoisers [26, 47, 48] are a cluster of recent approaches that parameterize a denoiser via the gradient map of a neural network. Although these works share similarities to our LPN, there are a few key differences.

1. Parameterization. In GS denoisers, the denoiser is parameterized by a gradient descent step: $f = \mathrm{Id} - \nabla g$, where Id represents the identity operator, and g is a scalar-valued function that is parameterized directly by a neural network [26], or defined implicitly by a network $N : \mathbb{R}^n \to \mathbb{R}^n$: $g(\mathbf{y}) = \frac{1}{2}\|\mathbf{y} - N(\mathbf{y})\|_2^2$ [47, 48]. Cohen et al. [26] also experiment with a denoiser architecture analogous to our LPN architecture, but find its denoising performance to be inferior to the GS denoiser (we will discuss this further in the final bullet below). In order to have accompanying convergence guarantees when used in PnP schemes, these GS parameterizations demand special structures on the learned denoiser—in particular, Lipschitz constraints on ∇g—which can be challenging to enforce in practice.
2. Proximal operator guarantee. The GS denoisers in Cohen et al. [26], Hurault et al. [47] are not a priori guaranteed to be proximal operators. Hurault et al. [48] proposed a way to guarantee the GS denoiser to be a proximal operator by limiting the Lipschitz constant of ∇g, also exploiting the characterization of Gribonval and Nikolova [44]. However, as a result, their denoiser necessarily has a bounded Lipschitz constant, even within the support of the data distribution, limiting the generality and universality of the proximals that can be approximated. On the other hand, LPNs could parameterize any continuous proximal operator on a compact domain given universality of ICNN [46].
3. Training. All GS denoiser methods used the conventional ℓ_2 loss for training. We propose the proximal matching loss and show that it is essential for the network to learn the correct proximal operator of the log-prior of data distribution. Indeed, we attribute the inferior performance of the ICNN-based architecture that Cohen et al. [26] experiment with, which is analogous to our LPN, to the fact that their experiments train this architecture on MMSE-based denoising, where "regression to the mean" on multimodal and nonlinear natural image data hinders performance (see, e.g., Delbracio and Milanfar [31] in this connection). The key insight that powers our successful application of LPNs in experiments is the proximal matching training framework, which allows us to make full use of the constrained capacity of the LPN in representing highly expressive proximal operators (corresponding to (nearly) maximum a-posteriori estimators for data distributions).

B Additional Theorems

B.1 Learning via proximal matching (discrete case)

Theorem B.1 (Learning via Proximal Matching (Discrete Case))**.** *Consider a signal* $\mathbf{x} \sim P(\mathbf{x})$*, with* $P(\mathbf{x})$ *a discrete distribution, and a noisy observation* $\mathbf{y} = \mathbf{x} + \sigma\boldsymbol{\varepsilon}$*, where* $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I})$ *and*

$\sigma > 0$. *Let* $m_\gamma(x) : \mathbb{R} \to \mathbb{R}$ *be defined by* $m_\gamma(x) = 1 - \exp\left(-\frac{x^2}{\gamma^2}\right)$ [2]. *Consider the optimization problem*

$$f^* = \underset{f \text{ measurable}}{\operatorname{argmin}} \lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[m_\gamma \left(\|f(\mathbf{y}) - \mathbf{x}\|_2 \right) \right].$$

Then, almost surely (i.e., for almost all $\mathbf{y}$*),* $f^*(\mathbf{y}) = \operatorname{argmax}_{\mathbf{c}} P(\mathbf{x} = \mathbf{c} \mid \mathbf{y})$.

The proof is deferred to Appendix C.3.

C Proofs

In this section, we include the proofs for the results presented in this paper.

C.1 Proof of Proposition 2

Proof. By Amos et al. [4, Proposition 1], ψ_θ is convex. Since the activation g is differentiable, ψ_θ is also differentiable. Hence, $f_\theta = \nabla \psi_\theta$ is the gradient of a convex function. Thus, by Proposition 1, f_θ is a proximal operator of a function. □

C.2 Proof of Theorem 3.1

Proof. First, note by linearity of the expectation that for any measurable f, one has

$$\lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[m_\gamma \left(\|f(\mathbf{y}) - \mathbf{x}\|_2 \right) \right] = 1 - \lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\varphi_{\gamma^2/2}(f(\mathbf{y}) - \mathbf{x}) \right], \tag{C.1}$$

where $\varphi_{\gamma^2/2}$ denotes the density of an isotropic Gaussian random variable with mean zero and variance $\gamma^2/2$. Because $p(\mathbf{x})$ is a continuous density with respect to the Lebesgue measure $d\mathbf{x}$, by Gaussian conditioning, we have that the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ admits a density $p_{\mathbf{x}|\mathbf{y}}$ with respect to $d\mathbf{x}$ as well. Taking conditional expectations, we have

$$\lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\varphi_{\gamma^2/2}(f(\mathbf{y}) - \mathbf{x}) \right] = \lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{y}} \mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\varphi_{\gamma^2/2}(f(\mathbf{y}) - \mathbf{x}) \right]. \tag{C.2}$$

From here, we can state the intuition for the remaining portion of the proof. Intuitively, because the Gaussian density $\varphi_{\sigma^2/2}$ concentrates more and more at zero as $\gamma \searrow 0$, and meanwhile is nevertheless a probability density for every $\gamma > 0$,[3] the inner expectation over $\mathbf{x} \mid \mathbf{y}$ leads to simply replacing the integrand with its value at $\mathbf{x} = f(\mathbf{y})$; the integrand is of course the conditional density of $\mathbf{x}$ given $\mathbf{y}$, and from here it is straightforward to argue that this leads the optimal f to be (almost surely) the conditional maximum a posteriori (MAP) estimate, under our regularity assumptions on $p(\mathbf{x})$.

To make this intuitive argument rigorous, we need to translate our regularity assumptions on $p(\mathbf{x})$ into regularity of $p_{\mathbf{x}|\mathbf{y}}$, interchange the γ limit in (C.2) with the expectation over $\mathbf{y}$, and instantiate a rigorous analogue of the heuristic "concentration" argument. First, we have by Bayes' rule and Gaussian conditioning

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}) = \frac{\varphi_{\sigma^2}(\mathbf{y} - \mathbf{x}) p(\mathbf{x})}{(\varphi_{\sigma^2} * p)(\mathbf{y}),}$$

where $*$ denotes convolution of densities; the denominator is the density of $\mathbf{y}$, and it satisfies $\varphi_{\sigma^2} * p > 0$ since $\varphi_{\sigma^2} > 0$. In particular, this implies that $p_{\mathbf{x}|\mathbf{y}}$ is a continuous function of $(\mathbf{x}, \mathbf{y})$, because $p(\mathbf{x})$ is continuous by assumption. We can then write, by the definition of convolution,

$$\mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\varphi_{\gamma^2/2}(f(\mathbf{y}) - \mathbf{x}) \right] = \varphi_{\gamma^2/2} * p_{\mathbf{x}|\mathbf{y}}(f(\mathbf{y})),$$

so following (C.2), we have

$$\lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\varphi_{\gamma^2/2}(f(\mathbf{y}) - \mathbf{x}) \right] = \lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{y}} \left[\varphi_{\gamma^2/2} * p_{\mathbf{x}|\mathbf{y}}(f(\mathbf{y})) \right]. \tag{C.3}$$

[2] This definition of m_γ differs slightly from the one in (3.1), but the two definitions are equivalent in terms of minimization objective as they only differ by a scaling constant.

[3] For readers familiar with signal processing or Schwartz's theory of distributions, this could be alternately stated as "the small-variance limit of the Gaussian density behaves like a Dirac delta distribution".

We are going to argue that the limit can be moved inside the expectation in (C.3) momentarily; for the moment, we consider the quantity that results after moving the limit inside the expectation. To treat this term, we apply a standard approximation to the identity argument to evaluate the limit of the preceding expression. [86, Ch. 3, Example 3] implies that the densities $\varphi_{\gamma^2/2}$ constitute an approximation to the identity as $\gamma \to 0$, and because $p_{\mathbf{x}|\mathbf{y}}$ is continuous, we can then apply [86, Ch. 3, Theorem 2.1] to obtain that

$$\lim_{\gamma \searrow 0} \varphi_{\gamma^2/2} * p_{\mathbf{x}|\mathbf{y}}(f(\mathbf{y})) = p_{\mathbf{x}|\mathbf{y}}(f(\mathbf{y})).$$

In particular, after justifying the interchange of limit and expectation in (C.3), we will have shown, by following our manipulations from (C.1), that

$$\lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[m_\gamma \left(\|f(\mathbf{y}) - \mathbf{x}\|_2 \right) \right] = 1 - \mathbb{E}_{\mathbf{y}} \left[p_{\mathbf{x}|\mathbf{y}}(f(\mathbf{y})) \right]. \tag{C.4}$$

We will proceed to conclude the proof from this expression, and justify the limit-expectation interchange at the end of the proof. The problem at hand is equivalent to the problem

$$\operatorname*{argmax}_{f \text{ measurable}} \; \mathbb{E}_{\mathbf{y}} \left[p_{\mathbf{x}|\mathbf{y}}(f(\mathbf{y})) \right].$$

Writing the expectation as an integral, we have by Bayes' rule as above

$$\mathbb{E}_{\mathbf{y}} \left[p_{\mathbf{x}|\mathbf{y}}(f(\mathbf{y})) \right] = \int_{\mathbb{R}^d} \varphi_{\sigma^2}(\mathbf{y} - f(\mathbf{y})) p(f(\mathbf{y})) d\mathbf{y}.$$

Let us define an auxiliary function $g : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ by $g(\mathbf{x}, \mathbf{y}) = \varphi_{\sigma^2}(\mathbf{y} - \mathbf{x}) p(\mathbf{x})$. Then

$$\mathbb{E}_{\mathbf{y}} \left[p_{\mathbf{x}|\mathbf{y}}(f(\mathbf{y})) \right] = \int_{\mathbb{R}^d} g(f(\mathbf{y}), \mathbf{y}) d\mathbf{y},$$

and moreover, for every $\mathbf{y}$, $g(\,\cdot\,, \mathbf{y})$ is continuous and compactly supported, by continuity and boundedness of the Gaussian density and the assumption that $p(\mathbf{x})$ is continuous and the random variable $\mathbf{x} \sim p(\mathbf{x})$ is bounded. We have for any measurable f

$$g(f(\mathbf{y}), \mathbf{y}) \leq \max_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}, \mathbf{y}). \tag{C.5}$$

Our aim is thus to argue that there is a choice of measurable f such that the preceding bound can be made tight; this will imply that any measurable f maximizing the objective $\mathbb{E}_{\mathbf{y}}[p_{\mathbf{x}|\mathbf{y}}(f(\mathbf{y}))]$ satisfies $g(f(\mathbf{y}), \mathbf{y}) = \max_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}, \mathbf{y})$ almost surely, or equivalently that $f(\mathbf{y}) \in \operatorname{argmax}_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}, \mathbf{y})$ almost surely. The claim will then follow, because $\operatorname{argmax}_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}, \mathbf{y}) = \operatorname{argmax}_{\mathbf{x} \in \mathbb{R}^d} p_{\mathbf{x}|\mathbf{y}}(\mathbf{x})$.

To this end, define $h(\mathbf{y}) = \max_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}, \mathbf{y})$. Then by the Weierstrass theorem, h is finite-valued, and for every $\mathbf{y}$ there exists some $\mathbf{c} \in \mathbb{R}^d$ such that $h(\mathbf{y}) = g(\mathbf{c}, \mathbf{y})$. Because g is continuous, it then follows from Rockafellar and Wets [78, Theorem 1.17(c)] that h is continuous. Moreover, because g is continuous and for every $\mathbf{y}$, $g(\,\cdot\,, \mathbf{y})$ is compactly supported, g is in particular level-bounded in $\mathbf{x}$ locally uniformly in $\mathbf{y}$ in the sense of Rockafellar and Wets [78, Definition 1.16], and it follows that the set-valued mapping $\mathbf{y} \mapsto \operatorname{argmax}_{\mathbf{x}} g(\mathbf{x}, \mathbf{y}) : \mathbb{R}^d \rightrightarrows \mathbb{R}^d$ is compact-valued, by the Weierstrass theorem, and outer semicontinuous relative to $\mathbb{R}^d$, by Rockafellar and Wets [78, Example 5.22]. Applying Rockafellar and Wets [78, Exercise 14.9, Corollary 14.6], we conclude that the set-valued mapping $\mathbf{y} \mapsto \operatorname{argmax}_{\mathbf{x}} g(\mathbf{x}, \mathbf{y})$ is measurable, and that in particular there exists a measurable function $f^* : \mathbb{R}^d \to \mathbb{R}^d$ such that $f^*(\mathbf{y}) \in \operatorname{argmax}_{\mathbf{x}} g(\mathbf{x}, \mathbf{y})$ for every $\mathbf{y} \in \mathbb{R}^d$. Thus, there is a measurable f attaining the bound in (C.5), and the claim follows after we can justify the preceding interchange of limit and expectation.

To justify the interchange of limit and expectation, we will apply the dominated convergence theorem, which requires us to show an integrable (with respect to the density of $\mathbf{y}$) upper bound for the function $\mathbf{y} \mapsto \mathbb{E}_{\mathbf{x}|\mathbf{y}}[\varphi_{\gamma^2/2}(f(\mathbf{y}) - \mathbf{x})]$. For this, we calculate

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\varphi_{\gamma^2/2}(f(\mathbf{y}) - \mathbf{x}) \right] &= \frac{1}{(\varphi_{\sigma^2} * p)(\mathbf{y})} \int_{\mathbb{R}^d} \varphi_{\sigma^2}(\mathbf{y} - \mathbf{x}) p(\mathbf{x}) \varphi_{\gamma^2/2}(f(\mathbf{y}) - \mathbf{x}) d\mathbf{x} \\
&\leq \frac{1}{(\varphi_{\sigma^2} * p)(\mathbf{y})} \left[\sup_{\mathbf{x}} \varphi_{\sigma^2}(\mathbf{y} - \mathbf{x}) p(\mathbf{x}) \right] \int_{\mathbb{R}^d} \varphi_{\gamma^2/2}(f(\mathbf{y}) - \mathbf{x}) d\mathbf{x}
\end{aligned}$$

$$= \frac{1}{(\varphi_{\sigma^2} * p)(\mathbf{y})} \left[\sup_{\mathbf{x}} \varphi_{\sigma^2}(\mathbf{y}-\mathbf{x}) p(\mathbf{x}) \right],$$

by Hölder's inequality and the fact that $\varphi_{\gamma^2/2}$ is a probability density. Because the random variable $\mathbf{x} \sim p(\mathbf{x})$ is assumed bounded, the density $p(\mathbf{x})$ has compact support, and the density $p(\mathbf{x})$ is assumed continuous, so there exists $R > 0$ such that if $\|\mathbf{x}\|_2 > R$ then $p(\mathbf{x}) = 0$, and $M > 0$ such that $p(\mathbf{x}) \le M$. We then have

$$\sup_{\mathbf{x}} \varphi_{\sigma^2}(\mathbf{y}-\mathbf{x}) p(\mathbf{x}) \le M \sup_{\mathbf{x}} \varphi_{\sigma^2}(\mathbf{y}-\mathbf{x}) \mathbb{1}_{\|\mathbf{x}\|_2 \le R}.$$

This means that the supremum can attain a nonzero value only on points where $\|\mathbf{x}\|_2 \le R$. On the other hand, for every $\mathbf{y}$ with $\|\mathbf{y}\|_2 \ge 2R$, whenever $\|\mathbf{x}\|_2 \le R$ the triangle inequality implies $\|\mathbf{y}-\mathbf{x}\|_2 \ge \|\mathbf{y}\|_2 - \|\mathbf{x}\|_2 \ge \frac{1}{2}\|\mathbf{y}\|_2$. Because the Gaussian density φ_{σ^2} is a radial function, we conclude that if $\|\mathbf{y}\|_2 \ge 2R$, one has

$$\sup_{\mathbf{x}} \varphi_{\sigma^2}(\mathbf{y}-\mathbf{x}) p(\mathbf{x}) \le M \varphi_{\sigma^2}(\mathbf{y}/2) = CM\varphi_{4\sigma^2}(\mathbf{y}),$$

where $C > 0$ depends only on d. At the same time, we always have

$$\sup_{\mathbf{x}} \varphi_{\sigma^2}(\mathbf{y}-\mathbf{x}) p(\mathbf{x}) \le \frac{M}{(2\pi\sigma^2)^{d/2}}.$$

Consequently, we have the composite upper bound

$$\sup_{\mathbf{x}} \varphi_{\sigma^2}(\mathbf{y}-\mathbf{x}) p(\mathbf{x}) \le \begin{cases} \frac{M}{(2\pi\sigma^2)^{d/2}} & \|\mathbf{y}\|_2 < 2R \\ 2M\varphi_{4\sigma^2}(\mathbf{y}) & \|\mathbf{y}\|_2 \ge 2R, \end{cases}$$

and by our work above

$$\mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\varphi_{\gamma^2/2}(f(\mathbf{y})-\mathbf{x}) \right] \le \frac{1}{(\varphi_{\sigma^2} * p)(\mathbf{y})} \times \begin{cases} \frac{M}{(2\pi\sigma^2)^{d/2}} & \|\mathbf{y}\|_2 < 2R \\ 2M\varphi_{4\sigma^2}(\mathbf{y}) & \|\mathbf{y}\|_2 \ge 2R. \end{cases}$$

Because $\varphi_{\sigma^2} * p$ is the density of $\mathbf{y}$, this upper bound is sufficient to apply the dominated convergence theorem to obtain

$$\lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\varphi_{\gamma^2/2}(f(\mathbf{y})-\mathbf{x}) \right] = \mathbb{E}_{\mathbf{y}} \lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\varphi_{\gamma^2/2}(f(\mathbf{y})-\mathbf{x}) \right].$$

Combining this assertion with the argument surrounding (C.4), we conclude the proof.

□

Remark (Other loss choices). Theorem 3.1 also holds for any m_γ such that m_γ is uniformly (in γ) bounded above, for each $\gamma > 0$ uniquely minimized at 0, and $\sup_{x \in \mathbb{R}} m_\gamma(x) - m_\gamma(\|\mathbf{x}\|_2)$ is an approximation to the identity as $\gamma \searrow 0$ (see [86, Ch. 3, §2]).

C.3 Proof of Theorem B.1

Proof. For brevity, we denote $\operatorname{argmax}_{\mathbf{c}} P(\mathbf{x} = \mathbf{c} \mid \mathbf{y})$ by $\mathrm{MAP}[\mathbf{x} \mid \mathbf{y}]$, i.e., the maximum a posteriori estimate of $\mathbf{x}$ given $\mathbf{y}$.

First, we show that $\mathrm{MAP}[\mathbf{x} \mid \mathbf{y}]$ is unique for almost all $\mathbf{y}$.

Consider $\mathbf{y}$ such that $\mathrm{MAP}[\mathbf{x} \mid \mathbf{y}]$ is not unique. There exists $i \neq j$, such that

$$\begin{aligned} & P(\mathbf{x}_i \mid \mathbf{y}) = P(\mathbf{x}_j \mid \mathbf{y}) \\ \iff & p(\mathbf{y} \mid \mathbf{x}_i) P(\mathbf{x}_i) = p(\mathbf{y} \mid \mathbf{x}_j) P(\mathbf{x}_j) \\ \iff & -\frac{1}{2}\|\mathbf{y}-\mathbf{x}_i\|^2 + \sigma^2 \log P(\mathbf{x}_i) = -\frac{1}{2}\|\mathbf{y}-\mathbf{x}_j\|^2 + \sigma^2 \log P(\mathbf{x}_j) \\ \iff & \langle \mathbf{y}, \frac{\mathbf{x}_i - \mathbf{x}_j}{2} \rangle = \frac{1}{2}\|\mathbf{x}_i\|^2 - \frac{1}{2}\|\mathbf{x}_j\|^2 - \sigma^2 \log P(\mathbf{x}_i) + \sigma^2 \log P(\mathbf{x}_j). \end{aligned}$$

i.e., $\mathbf{y}$ lies in a hyperplane defined by $\mathbf{x}_i, \mathbf{x}_j$ (note that $\mathbf{x}_i \neq \mathbf{x}_j$). Denote the hyperplane by

$$\mathcal{H}_{i,j} := \left\{ \mathbf{y} \mid \langle \mathbf{y}, \frac{\mathbf{x}_i - \mathbf{x}_j}{2} \rangle = \frac{1}{2}\|\mathbf{x}_i\|^2 - \frac{1}{2}\|\mathbf{x}_j\|^2 - \sigma^2 \log P(\mathbf{x}_i) + \sigma^2 \log P(\mathbf{x}_j) \right\}.$$

Consider

$$\mathcal{U} := \cup_{i \neq j} \mathcal{H}_{i,j}.$$

We have that $\forall \mathbf{y}$ with non-unique $\mathrm{MAP}[\mathbf{x} \mid \mathbf{y}]$,

$$\begin{aligned} &\exists i \neq j, \mathbf{y} \in \mathcal{H}_{i,j} \\ \iff &\mathbf{y} \in \mathcal{U}. \end{aligned}$$

Note that $\mathcal{U}$ has zero measure as a countable union of zero-measure sets, hence the measure of all $\mathbf{y}$ with non-unique $\mathrm{MAP}[\mathbf{x} \mid \mathbf{y}]$ is zero. Hence, for almost all $\mathbf{y}$, $\mathrm{MAP}[\mathbf{x} \mid \mathbf{y}]$ is unique.

Next, we show that for almost all $\mathbf{y}$,

$$f^*(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{c}} \mathbb{E}_{\mathbf{x}|\mathbf{y}}[\mathbb{1}_{\mathbf{c} \neq \mathbf{x}}].$$

Note that

$$\begin{aligned} &\lim_{\gamma \searrow 0} \mathbb{E}_{\mathbf{x},\mathbf{y}} \left[m_\gamma \left(\|f(\mathbf{y}) - \mathbf{x}\|_2 \right) \right] \\ =&\mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\lim_{\gamma \searrow 0} m_\gamma \left(\|f(\mathbf{y}) - \mathbf{x}\|_2 \right) \right] \\ =&\mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\mathbb{1}_{\|f(\mathbf{y}) - \mathbf{x}\|_2 \neq 0} \right] \\ =&\mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\mathbb{1}_{f(\mathbf{y}) \neq \mathbf{x}} \right]. \end{aligned}$$

Above, the first equality uses the monotone convergence theorem. Use the law of iterated expectations,

$$\mathbb{E}_{\mathbf{x},\mathbf{y}} \left[\mathbb{1}_{f(\mathbf{y}) \neq \mathbf{x}} \right] = \mathbb{E}_{\mathbf{y}} \mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\mathbb{1}_{f(\mathbf{y}) \neq \mathbf{x}} \right].$$

We will use this expression to study the global minimizers of the objective. By conditioning,

$$\mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\mathbb{1}_{f(\mathbf{y}) \neq \mathbf{x}} \right] \geq \min_{\mathbf{c}} \mathbb{E}_{\mathbf{x}|\mathbf{y}}[\mathbb{1}_{\mathbf{c} \neq \mathbf{x}}],$$

and so

$$\mathbb{E}_{\mathbf{y}} \left[\mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\mathbb{1}_{f(\mathbf{y}) \neq \mathbf{x}} \right] - \min_{\mathbf{c}} \mathbb{E}_{\mathbf{x}|\mathbf{y}}[\mathbb{1}_{\mathbf{c} \neq \mathbf{x}}] \right] \geq 0.$$

Because $p(\mathbf{y}) > 0$, it follows that every global minimizer of the objective f^* satisfies

$$\mathbb{E}_{\mathbf{x}|\mathbf{y}} \left[\mathbb{1}_{f^*(\mathbf{y}) \neq \mathbf{x}} \right] = \min_{\mathbf{c}} \mathbb{E}_{\mathbf{x}|\mathbf{y}}[\mathbb{1}_{\mathbf{c} \neq \mathbf{x}}] \text{ a.s.}$$

Hence, for almost all $\mathbf{y}$,

$$f^*(\mathbf{y}) \in \operatorname*{argmin}_{\mathbf{c}} \mathbb{E}_{\mathbf{x}|\mathbf{y}}[\mathbb{1}_{\mathbf{c} \neq \mathbf{x}}].$$

Finally, we show that $\operatorname{argmin}_{\mathbf{c}} \mathbb{E}_{\mathbf{x}|\mathbf{y}}[\mathbb{1}_{\mathbf{c} \neq \mathbf{x}}] = \mathrm{MAP}[\mathbf{x} \mid \mathbf{y}]$. The claim then follows from our preceding work showing that $\mathrm{MAP}[\mathbf{x} \mid \mathbf{y}]$ is almost surely unique. Consider

$$\begin{aligned} \mathbb{E}_{\mathbf{x}|\mathbf{y}}[\mathbb{1}_{\mathbf{c} \neq \mathbf{x}}] &= \sum_i P(\mathbf{x}_i \mid \mathbf{y}) \mathbb{1}_{\mathbf{c} \neq \mathbf{x}_i} \\ &= \sum_i P(\mathbf{x}_i \mid \mathbf{y})(1 - \mathbb{1}_{\mathbf{c} = \mathbf{x}_i}) \\ &= \sum_i P(\mathbf{x}_i \mid \mathbf{y}) - \sum_{\mathbf{x}_i = \mathbf{c}} P(\mathbf{x}_i \mid \mathbf{y}) \\ &= 1 - P(\mathbf{x} = \mathbf{c} \mid \mathbf{y}). \end{aligned}$$

Hence,

$$\begin{aligned} \operatorname*{argmin}_{\mathbf{c}} \mathbb{E}_{\mathbf{x}|\mathbf{y}}[\mathbb{1}_{\mathbf{c} \neq \mathbf{x}}] &= \operatorname*{argmax}_{\mathbf{c}} P(\mathbf{x} = \mathbf{c} \mid \mathbf{y}) \\ &= \mathrm{MAP}[\mathbf{x} \mid \mathbf{y}]. \end{aligned}$$

□

C.4 Proof of Theorem 3.2

We provide a proof of Theorem 3.2 under slightly more general assumptions in these appendices. The result is restated in this general setting below, as Theorem C.1.

Theorem C.1. *Consider the sequence of iterates $\mathbf{x}_k$, $k \in \{0, 1, \dots\}$, defined by Algorithm 3 run with a continuously differentiable measurement operator A and a LPN f_θ with softplus activations, trained with $0 < \alpha < 1$. Assume further that the data fidelity term $h(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - A(\mathbf{x})\|_2^2$ is definable in an o-minimal structure*[4] *and has L-Lipschitz gradient*[5]*, and that the step size satisfies $0 < \eta < 1/L$. Then, the iterates $\mathbf{x}_k$ converge to a fixed point $\mathbf{x}^*$ of Algorithm 3: that is, there exist $\mathbf{x}^* \in \mathbb{R}^n$ such that*

$$f_\theta\left(\mathbf{x}^* - \eta \nabla h(\mathbf{x}^*)\right) = \mathbf{x}^*, \tag{C.6}$$

and $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}^*$. Furthermore, $\mathbf{x}^*$ is a critical point[6] *of $h + \frac{1}{\eta}\phi_\theta$, where ϕ_θ is the prior associated to the LPN f_θ (i.e., $f_\theta = \mathrm{prox}_{\phi_\theta}$).*

Remark. Theorem 3.2 asserts fixed-point convergence of the iterates of Algorithm 3, and examining the proof of the more general version in Theorem C.1 shows moreover that $\mathbf{x}_k$ converges to a critical point of $h + \frac{1}{\eta}\phi_\theta$, where ϕ_θ is the implicitly-defined prior associated to f_θ, i.e. $f_\theta = \mathrm{prox}_{\phi_\theta}$. It is straightforward to adapt the proof of this result to using LPN in other PnP schemes such as PnP-ADMM (Algorithm 4), which is used in our experiments on inverse problems in Section 4, by appealing to different convergence analyses from the literature (see [99, Theorem 5.6], for example). We emphasize that Theorems 3.2 and C.1 require the bare minimum of assumptions on the learned LPN. This should be contrasted to PnP schemes which utilize a black-box denoiser for improved performance—convergence guarantees in this setting require restrictive a priori assumptions on the denoiser such as contractivity [83] or (firm) nonexpansivity [91, 92, 26, 27, 96, 47, 48],[7] which are difficult to verify or enforce in practice without sacrificing denoising performance—as well as PnP schemes that sacrifice expressivity for a principled approach by enforcing that the denoiser takes a restrictive form, such as being a (Gaussian) MMSE denoiser [105], a linear denoiser [45], or the proximal operator of an implicit convex function [85, 98]. Additionally, as shown in Gribonval [43], when interpreted as proximal operators, the prior in MMSE denoisers can be drastically different from the original (true data) prior, raising concerns about the correctness of the reconstruction result.

Because LPNs are by construction guaranteed to be proximal operators, as we have described in Section 3, we immediately obtain convergence guarantees for PnP schemes with LPN denoisers as a consequence of classical optimization analyses. Our proof appeals to a special case of a convergence result of [15] (see also [8, 37] for earlier results). Before proceeding to the proof, we state a few settings and results from Boţ et al. [15] that are useful for proving Theorem C.1, for better readability.

Problem 1 ([15, Problem 1])**.** *Let $f : \mathbb{R}^m \to (-\infty, +\infty]$ be a proper, lower semicontinuous function which is bounded below and let $h : \mathbb{R}^m \to \mathbb{R}$ be a Fréchet differentiable function with Lipschitz continuous gradient, i.e. there exists $L_{\nabla h} \geq 0$ such that $\|\nabla h(\mathbf{x}) - \nabla h(\mathbf{x}')\| \leq L_{\nabla h}\|\mathbf{x} - \mathbf{x}'\|$ for all $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^m$. Consider the optimization problem*

$$(P) \quad \inf_{\mathbf{x}\in\mathbb{R}^m} [f(\mathbf{x}) + h(\mathbf{x})].$$

Algorithm C.1 ([15, Algorithm 1])**.** *Choose $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{R}^m, \underline{\alpha}, \overline{\alpha} > 0, \beta \geq 0$ and the sequences $(\alpha_n)_{n\geq 1}, (\beta_n)_{n\geq 1}$ fulfilling*

$$0 < \underline{\alpha} \leq \alpha_n \leq \overline{\alpha} \;\; \forall n \geq 1$$

[4]This mild technical assumption is satisfied by an extremely broad array of nonlinear operators A: for example, any A which is a polynomial in the input $\mathbf{x}$ (in particular, linear A), or a rational function with nonvanishing denominator, is definable, and compositions and inverses of definable funtions are definable, so that definability of A implies definability of h [7]. We discuss these issues in more detail in the proof of the result.

[5]This is a very mild assumption. For example, when A is linear, the gradient of the data fidelity term ∇h has a Lipschitz constant no larger than $\|A^*A\|$, where $\|\cdot\|$ denotes the operator norm of a linear operator and A^* is the adjoint of A.

[6]In this work, the set of critical points of a function f is defined by $\mathrm{crit}(f) := \{\mathbf{x} : 0 \in \partial f(\mathbf{x})\}$, where ∂f is the limiting (Mordukhovich) Fréchet subdifferential of f (see definition in [15, Section 2]).

[7]Sun et al. [91] prove their results under an assumption that the denoiser is "θ-averaged" for $\theta \in (0, 1)$; see [91, §A]. When $\theta = \frac{1}{2}$, this coincides with the definition of firm nonexpansivity (c.f. [10]), which is itself a special case of nonexpansivity (Lipschitz constant of the denoiser being no larger than 1). As a point of reference, every convex function h satisfies that prox_h is firmly nonexpansive [76]. However, if h is nonconvex, prox_h need not even be Lipschitz—consider projection onto a nonconvex set.

and

$$0 \leq \beta_n \leq \beta \;\; \forall n \geq 1.$$

Consider the iterative scheme

$$(\forall n \geq 1)\; \mathbf{x}_{n+1} \in \underset{\mathbf{u}\in\mathbb{R}^m}{\operatorname{argmin}}\{D_F(\mathbf{u},\mathbf{x}_n) + \alpha_n\langle \mathbf{u}, \nabla h(\mathbf{x}_n)\rangle + \beta_n\langle \mathbf{u}, \mathbf{x}_{n-1} - \mathbf{x}_n\rangle + \alpha_n f(\mathbf{u})\}. \quad \text{(C.7)}$$

Here, $F : \mathbb{R}^m \to \mathbb{R}$ is σ-strongly convex, Fréchet differentiable and ∇F is $L_{\nabla F}$-Lipschitz continuous, with $\sigma, L_{\nabla F} > 0$; D_F is the Bregman distance to F.

Theorem C.2 ([15, Theorem 13])**.** *In the setting of Problem 1, choose $\underline{\alpha}$, $\overline{\alpha}$, β satisfying*

$$\sigma > \overline{\alpha} L_{\nabla_h} + 2\beta\frac{\overline{\alpha}}{\underline{\alpha}}. \quad \text{(C.8)}$$

Assume that $f + h$ is coercive and that

$$H : \mathbb{R}^m \times \mathbb{R}^m \to (-\infty, +\infty],\; H(\mathbf{x},\mathbf{x}') = (f+h)(\mathbf{x}) + \frac{\beta}{2\underline{\alpha}}\|\mathbf{x}-\mathbf{x}'\|^2,\; \forall(\mathbf{x},\mathbf{x}') \in \mathbb{R}^m \times \mathbb{R}^m$$

is a KL function[8]*. Let $(\mathbf{x}_n)_{n\in\mathbb{N}}$ be a sequence generated by Algorithm C.1. Then the following statements are true:*

1. *$\sum_{n\in\mathbb{N}} \|\mathbf{x}_{n+1} - \mathbf{x}_n\| < +\infty$*
2. *there exists $\mathbf{x} \in \operatorname{crit}(f + h)$ such that $\lim_{n\to+\infty} \mathbf{x}_n = \mathbf{x}$.*

Now, we prove Theorem C.1.

Proof of Theorem C.1. By Lemma C.4, there is a coercive function $\phi_\theta : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ such that $f_\theta = \operatorname{prox}_{\phi_\theta}$. The idea of the proof is to apply Theorem C.2 to our setting; this requires us to check that Algorithm 3 maps onto Algorithm C.1, and that our (implicitly-defined) objective function and parameter choices satisfy the requirements of this theorem. To this end, note that the application of f_θ in Algorithm 3 can be written as

$$\begin{aligned}
\mathbf{x}_{k+1} &= f_\theta\left(\mathbf{x}_k - \eta\nabla h(\mathbf{x}_k)\right) \\
&= \underset{\mathbf{x}'\in\mathbb{R}^n}{\operatorname{argmin}}\; \frac{1}{2}\left\|\mathbf{x}' - (\mathbf{x}_k - \eta\nabla h(\mathbf{x}_k))\right\|_2^2 + \phi_\theta(\mathbf{x}') \\
&= \underset{\mathbf{x}'\in\mathbb{R}^n}{\operatorname{argmin}}\; \frac{1}{2}\left\|\mathbf{x}' - \mathbf{x}_k\right\|_2^2 + \langle \mathbf{x}' - \mathbf{x}_k, \eta\nabla h(\mathbf{x}_k)\rangle + \phi_\theta(\mathbf{x}') \\
&= \underset{\mathbf{x}'\in\mathbb{R}^n}{\operatorname{argmin}}\; \frac{1}{2}\left\|\mathbf{x}' - \mathbf{x}_k\right\|_2^2 + \eta\langle \mathbf{x}', \nabla h(\mathbf{x}_k)\rangle + \eta\cdot\frac{1}{\eta}\phi_\theta(\mathbf{x}')
\end{aligned}$$

showing that Algorithm 3 corresponds to Algorithm C.1 with the Bregman distance $D_F(\mathbf{x},\mathbf{y}) = \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2$ (and correspondingly $F(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, which satisfies $\sigma = L_{\nabla F} = 1$), the momentum parameter $\beta = \beta_n = 0$, the step size $\alpha_n = \overline{\alpha} = \underline{\alpha} = \eta$, and $f = \frac{1}{\eta}\phi_\theta$. In the framework of Boţ et al. [15], Algorithm 3 minimizes the implicitly-defined objective $h + \eta^{-1}\phi_\theta$. Moreover, one checks that our choice of constant step size $0 < \eta < 1/L$ verifies the necessary condition (C.8), and because $h \geq 0$, coercivity of ϕ_θ implies that $h + \eta^{-1}\phi_\theta$ is coercive. The final hypothesis to check, which is slightly technical, is to show that the implicit objective $h + \eta^{-1}\phi_\theta$ is a KL function—this suffices to apply Theorem C.2 since for Algorithm 3, the parameter β in Theorem C.2 is zero. To this end, we make use of the fact that any proper lower-semicontinuous function definable in an "o-minimal" structure is a KL function [7, Theorem 4.1]; we will argue that our objective $h + \eta^{-1}\phi_\theta$ is definable to conclude convergence to a critical point of $h + \eta^{-1}\phi_\theta$ with Theorem C.2, then show that the convergence implies the asserted fixed point convergence (C.6). Because finite linear combinations of definable functions are definable and h is assumed definable (see [7, §4.3]: here and below, we make extensive use of the properties asserted in this section of this reference), it suffices to show that ϕ_θ is definable. To this end, notice that the defining equation for ϕ_θ in the $\alpha \in (0, 1)$ setting, namely (C.9), expresses ϕ_θ as a finite linear combination of finite products and compositions of different functions; we will argue that each constituent function is definable.

[8] In this work, a function being KL means it satisfies the Kurdyka-Łojasiewicz property [64], see [15, Definition 1].

1. **α-free LPN ψ_θ.** The definition of ψ_θ in Proposition 2 ensures that whenever (each coordinate function of) the elementwise activation function g is definable, ψ_θ is definable (following the inductive argument in the proof of Lemma C.4), by the fact that finite sums and compositions of definable functions are definable [7, Definition 4.1], and that affine functions are definable. In the present setting, the softplus activation $g = \beta^{-1}\log(1+\exp(\beta x))$ is definable, because $\exp$ is definable in a certain o-minimal structure and inverses of definable functions are definable. Thus ψ_θ is definable.

2. **Gradient of α-free LPN $\nabla\psi_\theta$.** This step of the proof uses the chain rule (essentially, the backpropagation algorithm to compute $\nabla\psi_\theta$), and the fact that finite products of definable functions remain definable. Arguing inductively (as in the inductive argument in the proof of Lemma C.4), it follows that $\nabla\psi_\theta$ is definable if the derivative of the activation function g is definable. We calculate $g'(x) = (1 + \exp(-\beta x))^{-1}$, which is a composition of a linear function (definable), the exponential function (definable), and a rational function with nonvanishing denominator on the range of the exponential function (semialgebraic [28, §2.2.1], hence definable). This shows that $\nabla\psi_\theta$ is definable.

3. **Inverse of α-regularized LPN f_θ^{-1}.** The map $f_\theta(\mathbf{x}) = \nabla\psi_\theta(\mathbf{x}) + \alpha\mathbf{x}$ is definable, as a sum of definable functions (by our work above). Because inverses of invertible definable functions are definable, and because f_θ is invertible (by Lemma C.4), it follows that f_θ^{-1} is definable.

4. **Squared ℓ_2 norm.** This is a polynomial function, hence semialgebraic and definable.

Thus $h + \eta^{-1}\phi_\theta$ is definable, continuous (by Lemma C.4), and proper (as a sum of real-valued functions, again by Lemma C.4), and therefore has the KL property. We can therefore apply Theorem C.2 to conclude convergence to a critical point of $h + \eta^{-1}\phi_\theta$. Finally, by Lemma C.3 and the continuity of f_θ and ∇h, we conclude convergence to a fixed point, $\mathbf{x} = f_\theta(\mathbf{x} - \eta\nabla h(\mathbf{x}))$, which is identical to (C.6). □

Lemma C.3 (Convergence Implies Fixed Point Convergence)**.** *Suppose $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^n$ is a* continuous *map that defines an iterative process, $\mathbf{x}_{k+1} = \mathcal{F}(\mathbf{x}_k)$. Assume $\mathbf{x}_k$ converges, i.e., $\exists\, \mathbf{x}^*$ such that* $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}^*$. *Then, $\mathbf{x}^*$ is a fixed point of $\mathcal{F}$, i.e., $\mathbf{x}^* = \mathcal{F}(\mathbf{x}^*)$.*

Proof.

$$\mathbf{x}^* = \lim_{k\to\infty} \mathbf{x}_k = \lim_{k\to\infty} \mathbf{x}_{k+1} = \lim_{k\to\infty} \mathcal{F}(\mathbf{x}_k) = \mathcal{F}\left(\lim_{k\to\infty} \mathbf{x}_k\right) = \mathcal{F}(\mathbf{x}^*).$$

The fourth equality follows from continuity of $\mathcal{F}$. □

Lemma C.4 (Regularity Properties of LPNs)**.** *Suppose f_θ is a LPN constructed following the recipe in Proposition 2, with softplus activations $\sigma(x) = (1/\beta)\log(1 + \exp(\beta x))$, where $\beta > 0$ is an arbitrary constant, and with strong convexity weight $0 < \alpha < 1$. Let $f_\theta(\mathbf{y}) = \nabla\psi_\theta(\mathbf{y}) + \alpha\mathbf{y}$ be the defining equation of the LPN. Then there is a function $\phi_\theta : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ such that $f_\theta = \mathrm{prox}_{\phi_\theta}$. Moreover, we have the following regularity properties:*

1. *ϕ_θ is coercive, i.e., we have $\phi_\theta(\mathbf{x}) \to +\infty$ as $\|\mathbf{x}\|_2 \to +\infty$.*

2. *$f_\theta : \mathbb{R}^n \to \mathbb{R}^n$ is surjective and invertible, with an inverse mapping $f_\theta^{-1} : \mathbb{R}^n \to \mathbb{R}^n$ which is continuous.*

3. *ϕ_θ is continuously differentiable and real-valued. In particular, it holds*

$$\begin{aligned}\phi_\theta(\mathbf{x}) = (1-\alpha)\langle f_\theta^{-1}(\mathbf{x}), \nabla\psi_\theta(f_\theta^{-1}(\mathbf{x}))\rangle \\ + \frac{\alpha(1-\alpha)}{2}\|f_\theta^{-1}(\mathbf{x})\|_2^2 - \tfrac{1}{2}\|\nabla\psi_\theta(f_\theta^{-1}(\mathbf{x}))\|_2^2 - \psi_\theta(f_\theta^{-1}(\mathbf{x})).\end{aligned} \tag{C.9}$$

Remark. Lemma C.4 does not, strictly speaking, require the softplus activation: the proof shows that any Lipschitz activation function with enough differentiability and slow growth at infinity, such as another smoothed verison of the ReLU activation, the GeLU, or the Swish activation, would also work.

Proof of Lemma C.4. The main technical challenge will be to establish coercivity of ϕ_θ, which always exists as necessary, by Propositions 1 and 2. We will therefore pursue this estimate as the main line of the proof, establishing the remaining assertions in the result statement along the way.

By Proposition 2, there exists ϕ_θ such that $f_\theta = \mathrm{prox}_{\phi_\theta}$. Now, using [44, Theorem 4(a)], for every $\mathbf{y} \in \mathbb{R}^n$,

$$\phi_\theta(f_\theta(\mathbf{y})) = \langle \mathbf{y}, f_\theta(\mathbf{y})\rangle - \tfrac{1}{2}\|f_\theta(\mathbf{y})\|_2^2 - \left(\psi_\theta(\mathbf{y}) + \tfrac{\alpha}{2}\|\mathbf{y}\|_2^2\right).$$

Using the definition of f_θ and minor algebra, we rewrite this as

$$\begin{aligned}\phi_\theta(f_\theta(\mathbf{y})) &= \langle \mathbf{y}, \nabla\psi_\theta(\mathbf{y}) + \alpha\mathbf{y}\rangle - \tfrac{1}{2}\|\nabla\psi_\theta(\mathbf{y}) + \alpha\mathbf{y}\|_2^2 - \left(\psi_\theta(\mathbf{y}) + \tfrac{\alpha}{2}\|\mathbf{y}\|_2^2\right)\\ &= (1-\alpha)\langle \mathbf{y}, \nabla\psi_\theta(\mathbf{y})\rangle + \frac{\alpha(1-\alpha)}{2}\|\mathbf{y}\|_2^2 - \tfrac{1}{2}\|\nabla\psi_\theta(\mathbf{y})\|_2^2 - \psi_\theta(\mathbf{y}). \end{aligned} \tag{C.10}$$

At this point, we observe that by Lemma C.5, the map $f_\theta : \mathbb{R}^n \to \mathbb{R}^n$ is invertible and surjective, with a continuous inverse mapping. This establishes the second assertion that we have claimed. In addition, taking inverses in (C.10) implies (C.9) and as a consequence the fact that ϕ_θ is real-valued, and the fact that it is continuously differentiable on $\mathbb{R}^n$ is then an immediate consequence of [44, Corollary 6(b)]. To conclude, it only remains to show that ϕ_θ is coercive, which we will accomplish by lower bounding the RHS of (C.10). By Lemma C.6, ψ_θ is L-Lipschitz for a constant $L > 0$. Thus, we have for every $\mathbf{y}$ (by the triangle inequality)

$$|\psi_\theta(\mathbf{y})| \le L\|\mathbf{y}\|_2 + K$$

for a (finite) constant $K \in \mathbb{R}$, depending only on θ. Now, the Cauchy-Schwarz inequality implies from the previous two statements (and $\|\nabla\psi_\theta\|_2 \le L$ by the Lipschitz property of ψ_θ)

$$\begin{aligned}\phi_\theta(f_\theta(\mathbf{y})) &\ge -(1-\alpha)\|\mathbf{y}\|_2\|\nabla\psi_\theta(\mathbf{y})\|_2 + \frac{\alpha(1-\alpha)}{2}\|\mathbf{y}\|_2^2 - \tfrac{1}{2}\|\nabla\psi_\theta(\mathbf{y})\|_2^2 - L\|\mathbf{y}\|_2 - K,\\ &\ge -L(1-\alpha)\|\mathbf{y}\|_2 + \frac{\alpha(1-\alpha)}{2}\|\mathbf{y}\|_2^2 - \frac{L^2}{2} - L\|\mathbf{y}\|_2 - K.\end{aligned}$$

We rewrite this estimate with some algebra as

$$\phi_\theta(f_\theta(\mathbf{y})) \ge \|\mathbf{y}\|_2\left(\frac{\alpha(1-\alpha)}{2}\|\mathbf{y}\|_2 - L(1-\alpha) - L\right) - \frac{L^2}{2} - K.$$

Next, we notice that when $0 < \alpha < 1$, the coefficient $\alpha(1-\alpha) > 0$; hence there is a constant $M > 0$ depending only on α and L such that for every $\mathbf{y}$ with $\|\mathbf{y}\|_2 \ge M$, one has

$$\frac{\alpha(1-\alpha)}{2}\|\mathbf{y}\|_2 - L(1-\alpha) - L \ge \frac{\alpha(1-\alpha)}{4}\|\mathbf{y}\|_2.$$

In turn, iterating this exact argument implies that there is another constant $M' > 0$ (depending only on α, L, and K) such that whenever $\|\mathbf{y}\|_2 \ge M'$, one has

$$\phi_\theta(f_\theta(\mathbf{y})) \ge \frac{\alpha(1-\alpha)}{8}\|\mathbf{y}\|_2^2.$$

We can therefore rewrite the previous inequality as

$$\phi_\theta(\mathbf{x}) \ge \frac{\alpha(1-\alpha)}{8}\|f_\theta^{-1}(\mathbf{x})\|_2^2, \tag{C.11}$$

for every $\mathbf{x}$ such that $\|f^{-1}(\mathbf{x})\|_2 \ge M'$. To conclude, we will show that whenever $\|\mathbf{x}\|_2 \to +\infty$, we also have $\|f_\theta^{-1}(\mathbf{x})\|_2 \to +\infty$, which together with (C.11) will imply coercivity of ϕ_θ. To this end, write $\|\cdot\|_{\mathrm{Lip}}$ for the Lipschitz seminorm:

$$\|f\|_{\mathrm{Lip}} = \sup_{\mathbf{y}\ne\mathbf{y}'} \frac{\|f(\mathbf{y}) - f(\mathbf{y}')\|_2}{\|\mathbf{y}-\mathbf{y}'\|_2},$$

and note that $\|f_\theta\|_{\mathrm{Lip}} \le \|\nabla\psi_\theta\|_{\mathrm{Lip}} + \alpha$. By Lemma C.7, $\nabla\psi_\theta$ is $L_{\nabla\psi_\theta}$-Lipschitz continuous, thus f_θ is $(L_{\nabla\psi_\theta} + \alpha)$-Lipschitz continuous,

$$\|f_\theta(\mathbf{y}) - f_\theta(\mathbf{y}')\|_2 \le (L_{\nabla\psi_\theta} + \alpha)\|\mathbf{y}-\mathbf{y}'\|_2.$$

Thus, taking inverses, we have

$$\|f_\theta^{-1}(\mathbf{x}) - f_\theta^{-1}(\mathbf{0})\|_2 \geq \frac{1}{L_{\nabla\psi_\theta} + \alpha}\|\mathbf{x}\|_2,$$

and it then follows from the triangle inequality that whenever $\mathbf{x}$ is such that $\|\mathbf{x}\|_2 \geq 2(L_{\nabla\psi_\theta} + \alpha)\|f_\theta^{-1}(\mathbf{0})\|_2$, we have in fact

$$\|f_\theta^{-1}(\mathbf{x})\|_2 \geq \frac{1}{2(L_{\nabla\psi_\theta} + \alpha)}\|\mathbf{x}\|_2.$$

Combining this estimate with (C.11), we obtain that for every $\mathbf{x}$ such that $\|\mathbf{x}\|_2 \geq 2(L_{\nabla\psi_\theta} + \alpha)\|f_\theta^{-1}(\mathbf{0})\|_2$ and $\|\mathbf{x}\|_2 \geq 2M'(L_{\nabla\psi_\theta} + \alpha)$, it holds

$$\phi_\theta(\mathbf{x}) \geq \frac{\alpha(1-\alpha)}{32(L_{\nabla\psi_\theta} + \alpha)^2}\|\mathbf{x}\|_2^2.$$

Taking limits in this last bound yields coercivity of ϕ_θ, and hence the claim. ☐

Lemma C.5 (Invertibility of f_θ and Continuity of f_θ^{-1})**.** *Suppose f_θ is a LPN constructed following the recipe in Proposition 2, with softplus activations $\sigma(x) = (1/\beta)\log(1 + \exp(\beta x))$, where $\beta > 0$ is an arbitrary constant, and with strong convexity weight $0 < \alpha < 1$. Then $f_\theta : \mathbb{R}^n \to \mathbb{R}^n$ is invertible and surjective, and $f_\theta^{-1} : \mathbb{R}^n \to \mathbb{R}^n$ is C^0.*

Proof. The proof uses the invertibility construction that we describe methodologically in Section 3. By construction, we have $f_\theta = \nabla\psi_\theta + \alpha\,\mathrm{Id}$, where Id denotes the identity operator on $\mathbb{R}^n$ (i.e., $\mathrm{Id}(\mathbf{x}) = \mathbf{x}$ for every $\mathbf{x} \in \mathbb{R}^n$).

For a fixed $\mathbf{x} \in \mathbb{R}^n$, consider the strongly convex minimization problem $\min_{\mathbf{y}} \psi_\theta(\mathbf{y}) + \frac{\alpha}{2}\|\mathbf{y}\|_2^2 - \langle \mathbf{x}, \mathbf{y}\rangle$. By first-order optimality condition, the minimizers are exactly $\{\mathbf{y} \mid \nabla\psi_\theta(\mathbf{y}) + \alpha\mathbf{y} = \mathbf{x}\}$. Furthermore, since the problem is strongly convex, it has a unique minimizer for each $\mathbf{x} \in \mathbb{R}^n$ [17]. Therefore, for each $\mathbf{x} \in \mathbb{R}^n$, there exists a unique $\mathbf{y}$ such that $\mathbf{x} = \nabla\psi_\theta(\mathbf{y}) + \alpha\mathbf{y} = f_\theta(\mathbf{y})$.

The argument above establishes that $f_\theta : \mathbb{R}^n \to \mathbb{R}^n$ is injective and surjective; hence there exists an inverse $f_\theta^{-1} : \mathbb{R}^n \to \mathbb{R}^n$. To conclude the proof, we will argue that f_θ^{-1} is continuous. To this end, we use the characterization of continuity which states that a function $g : \mathbb{R}^n \to \mathbb{R}^n$ is continuous if and only if for every open set $U \subset \mathbb{R}^n$, we have that $g^{-1}(U)$ is open, where $g^{-1}(U) = \{\mathbf{x} \in \mathbb{R}^n \mid g(\mathbf{x}) \in U\}$ (e.g., [82, Theorem 4.8]). To show that f_θ^{-1} is continuous, it is therefore equivalent to show that for every open set $U \subset \mathbb{R}^n$, one has that $f_\theta(U)$ is open. But this follows from invariance of domain, a standard result in algebraic topology (e.g., [32, Proposition 7.4]), since f_θ is injective and continuous. We have thus shown that f_θ is invertible, and that its inverse is continuous, as claimed. ☐

Lemma C.6 (Lipschitzness of ψ_θ)**.** *ψ_θ is L_{ψ_θ}-Lipschitz continuous for a constant $L_{\psi_\theta} > 0$, i.e., $|\psi_\theta(\mathbf{y}) - \psi_\theta(\mathbf{y}')| \leq L_{\psi_\theta}\|\mathbf{y} - \mathbf{y}'\|_2$, for all $\mathbf{y}, \mathbf{y}' \in \mathbb{R}^n$.*

Proof. Note that the derivative σ' of the softplus activation satisfies $\sigma'(x) = 1/(1 + \exp(-\beta x))$, which is no larger than 1, since $\exp(x) > 0$ for $x \in \mathbb{R}$. Here and below, if F is a map between Euclidean spaces we will write DF for its differential (a map from the domain of F to the space of linear operators from the domain of F to the range of F). Hence the activation function g in Proposition 2 is 1-Lipschitz with respect to the ℓ_2 norm, since the induced (by elementwise application) map $g : \mathbb{R}^n \to \mathbb{R}^n$ defined by $g(\mathbf{y}) = [\sigma(x_1), \ldots, \sigma(x_n)]^T$ satisfies

$$Dg(\mathbf{y}) = \begin{bmatrix} \sigma'(x_1) & & \\ & \ddots & \\ & & \sigma'(x_n) \end{bmatrix},$$

which is bounded in operator norm by $\sup_x |\sigma'(x)| \leq 1$. First, notice that

$$\begin{aligned} \|\psi_\theta(\mathbf{y}) - \psi_\theta(\mathbf{y}')\|_2 &= \|\mathbf{w}^T(\mathbf{z}_K(\mathbf{y}) - \mathbf{z}_K(\mathbf{y}'))\|_2 \\ &\leq \|\mathbf{w}\|_2\|\mathbf{z}_K(\mathbf{y}) - \mathbf{z}_K(\mathbf{y}')\|_2 \end{aligned}$$

by Cauchy-Schwarz. Meanwhile, we have similarly

$$\|\mathbf{z}_1(\mathbf{y}) - \mathbf{z}_1(\mathbf{y}')\|_2 \le \|\mathbf{H}_1\|\|\mathbf{y} - \mathbf{y}'\|_2,$$

where $\|\cdot\|$ denotes the operator norm of a matrix, and for integer $0 < k < K+1$

$$\|\mathbf{z}_k(\mathbf{y}) - \mathbf{z}_k(\mathbf{y}')\|_2 \le \|\mathbf{W}_k\|\|\mathbf{z}_{k-1}(\mathbf{y}) - \mathbf{z}_{k-1}(\mathbf{y}')\|_2 + \|\mathbf{H}_k\|\|\mathbf{y} - \mathbf{y}'\|_2.$$

By a straightforward induction, it follows that ψ_θ is L-Lipschitz for a constant $L > 0$ (depending only on θ). $\square$

Lemma C.7 (Lipschitzness of $\nabla\psi_\theta$)**.** *$\nabla\psi_\theta$ is $L_{\nabla\psi_\theta}$-Lipschitz continuous, for a constant $L_{\nabla\psi_\theta} > 0$.*

Proof. We will upper bound $\|\nabla\psi_\theta\|_{\mathrm{Lip}}$ by deriving an explicit expression for the gradient. By the defining formulas in Proposition 2, we have

$$\psi_\theta(\mathbf{y}) = \mathbf{w}^T \mathbf{z}_K(\mathbf{y}) + \mathbf{b}.$$

The chain rule gives

$$\nabla\psi_\theta(\mathbf{y}) = D\mathbf{z}_K(\mathbf{y})^* \mathbf{w},$$

where * denotes the adjoint of a linear operator, so for any $\mathbf{y}, \mathbf{y}'$ we have

$$\begin{aligned}
\|\nabla\psi_\theta(\mathbf{y}) - \nabla\psi_\theta(\mathbf{y}')\|_2 &= \left\|\left(D\mathbf{z}_K(\mathbf{y}) - D\mathbf{z}_K(\mathbf{y}')\right)^* \mathbf{w}\right\|_2 \\
&\le \left\|\left(D\mathbf{z}_K(\mathbf{y}) - D\mathbf{z}_K(\mathbf{y}')\right)^*\right\| \|\mathbf{w}\|_2 \\
&= \|D\mathbf{z}_K(\mathbf{y}) - D\mathbf{z}_K(\mathbf{y}')\| \|\mathbf{w}\|_2 \\
&\le \|D\mathbf{z}_K(\mathbf{y}) - D\mathbf{z}_K(\mathbf{y}')\|_{\mathrm{F}} \|\mathbf{w}\|_2,
\end{aligned}$$

where the first inequality uses Cauchy-Schwarz, the third line uses that the operator norm of a linear operator is equal to that of its adjoint, and the third line uses that the operator norm is upper-bounded by the Frobenius norm. This shows that we obtain a Lipschitz property in ℓ_2 for $\nabla\psi_\theta$ by obtaining one for the differential $D\mathbf{z}_K$ of the LPN's last-layer features. To this end, we can use the chain rule to compute for any integer $1 < k < K+1$ and any $\boldsymbol{\delta} \in \mathbb{R}^n$

$$D\mathbf{z}_k(\mathbf{y})(\boldsymbol{\delta}) = g'\left(\mathbf{W}_k\mathbf{z}_{k-1}(\mathbf{y}) + \mathbf{H}_k\mathbf{y} + \mathbf{b}_k\right) \odot \left[\mathbf{W}_k D\mathbf{z}_{k-1}(\mathbf{y})(\boldsymbol{\delta}) + \mathbf{H}_k\boldsymbol{\delta}\right],$$

where g' is the derivative of the softplus activation function g, applied elementwise, and $\odot$ denotes elementwise multiplication, and similarly

$$D\mathbf{z}_1(\mathbf{y})(\boldsymbol{\delta}) = g'\left(\mathbf{H}_1\mathbf{y} + \mathbf{b}_1\right) \odot \left[\mathbf{H}_1\boldsymbol{\delta}\right].$$

Now notice that for any vectors $\mathbf{v}$ and $\mathbf{y}$ and any matrix $\mathbf{A}$ such that the sizes are compatible, we have $\mathbf{v} \odot (\mathbf{A}\mathbf{y}) = \mathrm{diag}(\mathbf{v})\mathbf{A}\mathbf{y}$. Hence we can rewrite the above recursion in matrix form as

$$D\mathbf{z}_k(\mathbf{y}) = \underbrace{\mathrm{diag}\left(g'\left(\mathbf{W}_k\mathbf{z}_{k-1}(\mathbf{y}) + \mathbf{H}_k\mathbf{y} + \mathbf{b}_k\right)\right)}_{\mathbf{D}_k(\mathbf{y})} \left[\mathbf{W}_k D\mathbf{z}_{k-1}(\mathbf{y}) + \mathbf{H}_k\right],$$

and similarly

$$D\mathbf{z}_1(\mathbf{y}) = \underbrace{\mathrm{diag}\left(g'\left(\mathbf{H}_1\mathbf{y} + \mathbf{b}_1\right)\right)}_{\mathbf{D}_1(\mathbf{y})} \mathbf{H}_1.$$

We will proceed with an inductive argument. First, by the submultiplicative property of the Frobenius norm and the triangle inequality for the Frobenius norm, note that we have if $1 < k < K+1$

$$\begin{aligned}
\|D\mathbf{z}_k(\mathbf{y}) - D\mathbf{z}_k(\mathbf{y}')\|_{\mathrm{F}} &\le \|\mathbf{D}_k(\mathbf{y}) - \mathbf{D}_k(\mathbf{y}')\|_{\mathrm{F}} \\
&\quad + \|\mathbf{D}_k(\mathbf{y})\mathbf{W}_k D\mathbf{z}_{k-1}(\mathbf{y}) - \mathbf{D}_k(\mathbf{y}')\mathbf{W}_k D\mathbf{z}_{k-1}(\mathbf{y}')\|_{\mathrm{F}} \\
&\le \|\mathbf{D}_k(\mathbf{y}) - \mathbf{D}_k(\mathbf{y}')\|_{\mathrm{F}} \\
&\quad + \|\mathbf{D}_k(\mathbf{y})\mathbf{W}_k D\mathbf{z}_{k-1}(\mathbf{y}) - \mathbf{D}_k(\mathbf{y})\mathbf{W}_k D\mathbf{z}_{k-1}(\mathbf{y}')\|_{\mathrm{F}} \\
&\quad + \|\mathbf{D}_k(\mathbf{y})\mathbf{W}_k D\mathbf{z}_{k-1}(\mathbf{y}') - \mathbf{D}_k(\mathbf{y}')\mathbf{W}_k D\mathbf{z}_{k-1}(\mathbf{y}')\|_{\mathrm{F}} \\
&\le \|\mathbf{D}_k(\mathbf{y}) - \mathbf{D}_k(\mathbf{y}')\|_{\mathrm{F}} \\
&\quad + \|\mathbf{D}_k(\mathbf{y})\mathbf{W}_k\|_{\mathrm{F}} \|D\mathbf{z}_{k-1}(\mathbf{y}) - D\mathbf{z}_{k-1}(\mathbf{y}')\|_{\mathrm{F}} \\
&\quad + \|D\mathbf{z}_{k-1}(\mathbf{y}')\|_{\mathrm{F}} \|\mathbf{D}_k(\mathbf{y})\mathbf{W}_k - \mathbf{D}_k(\mathbf{y}')\mathbf{W}_k\|_{\mathrm{F}}
\end{aligned}$$

$$\leq (1 + \|\mathbf{W}_k\|_{\mathrm{F}})\, \|\mathbf{D}_k(\mathbf{y}) - \mathbf{D}_k(\mathbf{y}')\|_{\mathrm{F}} + \|\mathbf{D}_k(\mathbf{y})\|_{\mathrm{F}} \|\mathbf{W}_k\|_{\mathrm{F}} \|D\mathbf{z}_{k-1}(\mathbf{y}) - D\mathbf{z}_{k-1}(\mathbf{y}')\|_{\mathrm{F}}.$$

Now, as we have shown above, $g'(x) = (1 + \exp(-\beta x))^{-1} \leq 1$ for every $x \in \mathbb{R}$. This implies

$$\|\mathbf{D}_k(\mathbf{y})\|_{\mathrm{F}} \leq \sqrt{n_k},$$

where n_k is the output dimension of k-th layer. Moreover, we calculate with the chain rule

$$g''(x) = \frac{\beta e^{-\beta x}}{(1 + e^{-\beta x})^2},$$

and by L'Hôpital's rule, we have that $\lim_{x \to +\infty} \frac{x}{(1+x)^2} = 0$, so that by continuity, g'' is bounded for $x \in \mathbb{R}$. It follows that g' is Lipschitz. Notice now that

$$\begin{aligned}\|\mathbf{D}_k(\mathbf{y}) - \mathbf{D}_k(\mathbf{y}')\|_{\mathrm{F}} &= \|g'\left(\mathbf{W}_k \mathbf{z}_{k-1}(\mathbf{y}) + \mathbf{H}_k \mathbf{y} + \mathbf{b}_k\right) - g'\left(\mathbf{W}_k \mathbf{z}_{k-1}(\mathbf{y}') + \mathbf{H}_k \mathbf{y}' + \mathbf{b}_k\right)\|_2 \\ &\leq \|g'\|_{\mathrm{Lip}} \left(\|\mathbf{W}_k\|_{\mathrm{F}} \|\mathbf{z}_{k-1}(\mathbf{y}) - \mathbf{z}_{k-1}(\mathbf{y}')\|_2 + \|\mathbf{H}_k\|_{\mathrm{F}} \|\mathbf{y} - \mathbf{y}'\|_2,\right)\end{aligned}$$

where in the second line we used the fact that the derivative of an elementwise function is a diagonal matrix together with the triangle inequality and Cauchy-Schwarz. However, we have already argued previously by induction that ψ_θ is Lipschitz, and in particular each of its feature maps $\mathbf{z}_k$ is Lipschitz. We conclude that $\mathbf{D}_k$ is Lipschitz, and the Lipschitz constant depends only on θ. This means that there are constants L_k, L'_k depending only on n and θ such that

$$\|D\mathbf{z}_k(\mathbf{y}) - D\mathbf{z}_k(\mathbf{y}')\|_{\mathrm{F}} \leq L_k \|\mathbf{y} - \mathbf{y}'\|_2 + L'_k \|D\mathbf{z}_{k-1}(\mathbf{y}) - D\mathbf{z}_{k-1}(\mathbf{y}')\|_{\mathrm{F}}.$$

Meanwhile, following the same arguments as above, but in a slightly simplified setting, we obtain

$$\begin{aligned}\|D\mathbf{z}_1(\mathbf{y}) - D\mathbf{z}_1(\mathbf{y}')\|_{\mathrm{F}} &= \|\mathbf{D}_1(\mathbf{y})\mathbf{H}_1 - \mathbf{D}_1(\mathbf{y}')\mathbf{H}_1\|_{\mathrm{F}} \\ &\leq \|\mathbf{H}_1\|_{\mathrm{F}} \|\mathbf{D}_1(\mathbf{y}) - \mathbf{D}_1(\mathbf{y}')\|_{\mathrm{F}} \\ &\leq \|g'\|_{\mathrm{Lip}} \|\mathbf{H}_1\|_{\mathrm{F}}^2 \|\mathbf{y} - \mathbf{y}'\|_2,\end{aligned}$$

which demonstrates that $D\mathbf{z}_1$ is also Lipschitz, with the Lipschitz constant depending only on θ. By induction, we therefore conclude that there is $L_{\nabla \psi_\theta} > 0$ such that

$$\|\nabla \psi_\theta(\mathbf{y}) - \nabla \psi_\theta(\mathbf{y}')\|_2 \leq L_{\nabla \psi_\theta} \|\mathbf{y} - \mathbf{y}'\|_2,$$

with $L_{\nabla \psi_\theta}$ depending only on θ and n_k. □

D Algorithms

D.1 Algorithm for Prior Estimation

Algorithm 1 Prior estimation from LPN

Input: Learned proximal network $f_\theta(\cdot)$, $\psi_\theta(\cdot)$ that satisfies $f_\theta = \nabla \psi_\theta$, query point $\mathbf{x}$
Find $\mathbf{y}$ such that $f_\theta(\mathbf{y}) = \mathbf{x}$, by solving $\min_{\mathbf{y}} \psi_\theta(\mathbf{y}) - \langle \mathbf{x}, \mathbf{y} \rangle$ or $\min_{\mathbf{y}} \|f_\theta(\mathbf{y}) - \mathbf{x}\|_2^2$
$\phi \leftarrow \langle \mathbf{y}, \mathbf{x} \rangle - \frac{1}{2}\|\mathbf{x}\|^2 - \psi_\theta(\mathbf{y})$
Output: ϕ ▷ The prior at $\mathbf{x}$

D.2 Algorithm for LPN training

Algorithm 2 Training the LPN with proximal matching loss

Input: Training dataset $\mathcal{D}$, initial LPN parameter θ, loss schedule $\gamma(\cdot)$, noise standard deviation σ, number of iterations K, network optimizer $\mathrm{Optm}(\cdot, \cdot)$
$k \leftarrow 0$
repeat
    Sample $\mathbf{x} \sim \mathcal{D}$, $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$
    $\mathbf{y} \leftarrow \mathbf{x} + \sigma\varepsilon$
    $\mathcal{L}_{PM} \leftarrow m_{\gamma(k)}(\|f_\theta(\mathbf{y}) - \mathbf{x}\|_2)$
    $\theta \leftarrow \mathrm{Optm}(\theta, \nabla_\theta \mathcal{L}_{PM})$ ▷ Update network parameters
    $k \leftarrow k + 1$
until $k = K$
Output: θ ▷ Trained LPN

D.3 Algorithm for solving inverse problems with LPN and PnP-PGD

Algorithm 3 Solving inverse problems with LPN and PnP-PGD

Input: Trained LPN f_θ, measurement operator A, measurement $\mathbf{y}$, data fidelity function $h(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - A(\mathbf{x})\|_2^2$, initial estimation $\mathbf{x}_0$, step size η, number of iterations K

1: **for** $k = 0$ **to** $K - 1$ **do**
2: $\quad \mathbf{x}_{k+1} \leftarrow f_\theta(\mathbf{x}_k - \eta \nabla h(\mathbf{x}_k))$
3: **end for**

Output: $\mathbf{x}_K$

D.4 Algorithm for solving inverse problems with LPN and PnP-ADMM

Algorithm 4 Solving inverse problem with LPN and PnP-ADMM

Input: Trained LPN f_θ, measurement operator A, measurement $\mathbf{y}$, initial estimation $\mathbf{x}_0$, number of iterations K, penalty parameter ρ

1: $\mathbf{u}_0 \leftarrow 0$, $\mathbf{z}_0 \leftarrow \mathbf{x}_0$
2: **for** $k = 0$ **to** $K - 1$ **do**
3: $\quad \mathbf{x}_{k+1} \leftarrow \operatorname{argmin}_{\mathbf{x}} \left\{ \frac{1}{2}\|\mathbf{y} - A(\mathbf{x})\|_2^2 + \frac{\rho}{2}\|\mathbf{z}_k - \mathbf{u}_k - \mathbf{x}\|_2^2 \right\}$
4: $\quad \mathbf{z}_{k+1} \leftarrow f_\theta(\mathbf{u}_k + \mathbf{x}_{k+1})$
5: $\quad \mathbf{u}_{k+1} \leftarrow \mathbf{u}_k + \mathbf{x}_{k+1} - \mathbf{z}_{k+1}$
6: **end for**

Output: $\mathbf{x}_K$

E Experimental Details

E.1 Details of Laplacian experiment

The LPN architecture contains four linear layers and 50 hidden neurons at each layer, with $\beta = 10$ in softplus activation. The LPN is trained by Gaussian noise with $\sigma = 1$, Adam optimizer [55] and batch size of 2000. For either ℓ_2 or ℓ_1 loss, the model is trained for a total of $20k$ iterations, including $10k$ iterations with learning rate $lr = 1e-3$, and another $10k$ with $lr = 1e-4$. For the proximal matching loss, we initialize the model from the ℓ_1 checkpoint, and train according to the schedule in Table 2.

Number of iterations	γ in $\mathcal{L}_{\mathcal{PM}}$	Learning rate
$2k$	0.5	$1e-3$
$2k$	0.5	$1e-4$
$4k$	0.4	$1e-4$
$4k$	0.3	$1e-4$
$4k$	0.2	$1e-5$
$4k$	0.1	$1e-5$
$4k$	0.1	$1e-6$

Table 2: The schedule of training LPN with proximal matching loss in the Laplacian experiment.

E.2 Details of MNIST experiment

The LPN architecture is implemented with four convolution layers and 64 hidden neurons at each layer, with $\alpha = 0.01$ and softplus $\beta = 10$. The model is trained on the MNIST training set containing $50k$ images, with Gaussian noise with $\sigma = 0.1$ and batch size of 200. The LPN is first trained by ℓ_1 loss for $20k$ iterations; and then by the proximal matching loss for $20k$ iterations, with γ initialized at $0.64 * 28 = 17.92$ and halved every $5k$ iterations. The learned prior is evaluated on 100 MNIST test images. Conjugate gradient is used to solve the convex inversion problem: $\min_{\mathbf{y}} \psi_\theta(\mathbf{y}) - \langle \mathbf{x}, \mathbf{y} \rangle$ in prior estimation.

E.3 Details of CelebA experiment

We center-cropped CelebA images from 178×218 to 128×128, and normalized the intensities to $[0, 1]$. Since CelebA images are larger and more complex than MNIST, we use a deeper and wider network. The LPN architecture includes 6 convolution layers with 128 hidden neurons per layer, with $\alpha = 1e-6$ and $\beta = 10$. For LPN training, we use Gaussian noise with standard deviation $\sigma = 0.1$. We pretrain the network with ℓ_1 loss for $20k$ iterations with $lr = 1e-3$. Then, we train with proximal matching $\mathcal{L}_{\mathcal{PM}}$ for $20k$ iterations using $lr = 1e-4$, with the schedule of γ as follows: initialized at $0.64 \times \sqrt{128 \times 128 \times 3} \approx 142$, and multiplied by 0.5 every $5k$ iterations. We use a batch size of 64 throughout the training.

PnP We use PnP-ADMM to perform deblurring on CelebA for all the denoisers concerned, i.e., BM3D, DnCNN, and our LPN (see Algorithm 4). For all models, we use the same set of hyperparameters for ADMM: number of iterations $K = 10$ and penalty parameter $\rho = 0.2$. We implement the PnP-ADMM algorithm based on the SCICO package [9].

E.4 Details of Mayo-CT experiment

We use the public dataset from Mayo-Clinic for the low-dose CT grand challenge (Mayo-CT) [69], which contains abdominal CT scans from 10 patients and a total of 2378 images of size 512×512. Following [65], we use 128 images for testing and leave the rest for training. The LPN architecture contains 7 convolution layers with 256 hidden neurons per layer, with $\alpha = 0$ and $\beta = 100$. During training, we randomly cropped training images into patches of size 128×128. At test time, LPN is applied with a sliding window of the same size and a stride of 64. The training procedure of LPN is the same as for CelebA, except that γ in proximal matching loss is initialized to $0.64 \times \sqrt{128 \times 128} \approx 82$.

Sparse-view CT Following Lunz et al. [65], we simulate CT sinograms using a parallel-beam geometry with 200 angles and 400 detectors. The angles are uniformly spaced between $-90°$ and $90°$. White Gaussian noise with standard deviation $\sigma = 2.0$ is added to the sinogram data to simulate noise in measurement. We implement AR in PyTroch based on its public TensorFlow code[9]; for UAR, we use the publicly available code and model weights [10].

Compressed sensing For compressed sensing, we implement the random Gaussian sampling matrix following Jalal et al. [50], and add noise of $\sigma = 0.001$ to the measurements. The wavelet-based sparse recovery method for compressed sensing minimizes the object $\frac{1}{2}\|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda \|W\mathbf{x}\|_1$, where A is the sensing matrix and W is a suitable wavelet transform. We select the "db4" wavelet and $\lambda = 0.01$. We use proximal gradient descent with a step size of 0.5, stopping criterion $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_1 < 1e-4$, and maximum number of iterations $= 1000$.

PnP We use LPN with PnP-ADMM. For sparse-view CT, we use the following hyperparameters: number of iterations $K = 15$, scale of data fidelity term $= 8$, and penalty parameter $\rho = 0.05$. For compressed sensing, we use the following: number of iterations $K = 80$, scale of data fidelity term $= 1$, and penalty parameter $\rho = 0.05$.

F Discussions

F.1 Other ways to parameterize gradients of convex functions via neural networks

Input convex gradient networks (ICGN) [77] provide another way to parameterize gradients of convex functions. The model performs line integral over Positive Semi-Definite (PSD) Hessian matrices, where the Hessians are implicitly parameterized by the Gram product of Jacobians of neural

[9]AR: https://github.com/lunz-s/DeepAdverserialRegulariser.

[10]UAR: https://github.com/Subhadip-1/unrolling_meets_data_driven_regularization.

networks, hence guaranteed to be PSD. However, this approach only permits single-layer networks in order to satisfy a crucial PDE condition in its formulation [77], significantly limiting its representation capacity. Furthermore, the evaluation of the convex function is less straightforward than ICNN, which is an essential step in prior estimation from LPN (see Section 3). We therefore adopt the differentiation-based parameterization in this work and leave the exploration of other possibilities to future research.

G Additional Experimental Results

G.1 Learning soft-thresholding from Laplacian distribution

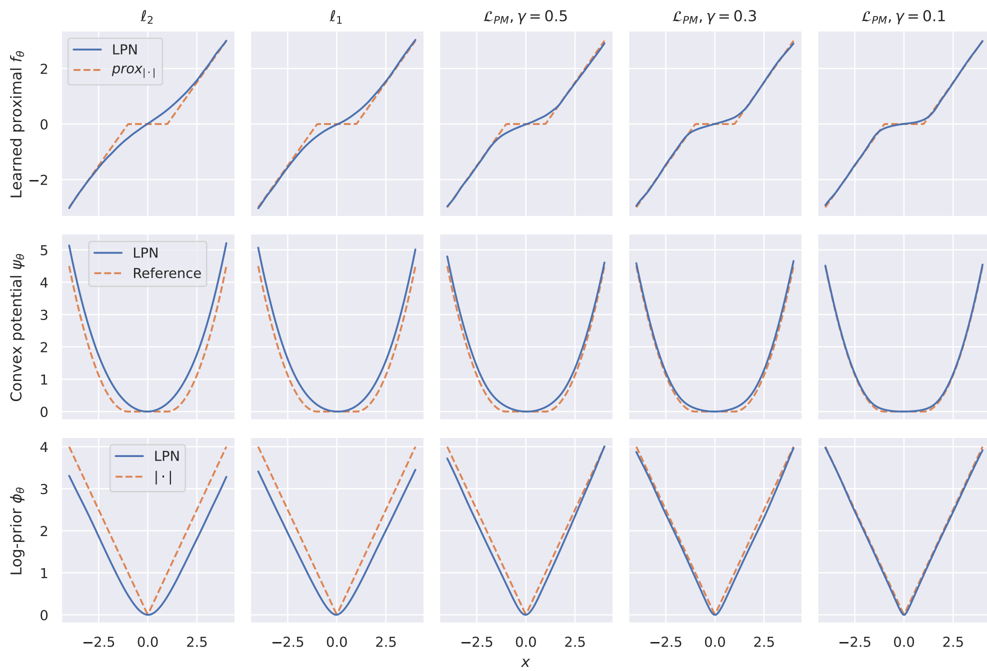


Figure 4: The proximal operator f_θ, convex potential ψ_θ, and log-prior ϕ_θ learned by LPN via different losses: the square ℓ_2 loss, ℓ_1 loss, and the proposed proximal matching loss $\mathcal{L}_{PM}$ with different $\gamma \in \{0.5, 0.3, 0.1\}$. The ground-truth data distribution is the Laplacian $p(x) = \frac{1}{2}\exp(-|x|)$, with log-prior $-\log p(x) = |x| - \log(\frac{1}{2})$. With proximal matching loss, the learned proximal f_θ progressively approaches the ground-truth $\text{prox}_{|\cdot|}$ as γ shrinks from 0.5 to 0.1.

G.2 Learning a prior for MNIST – image blur

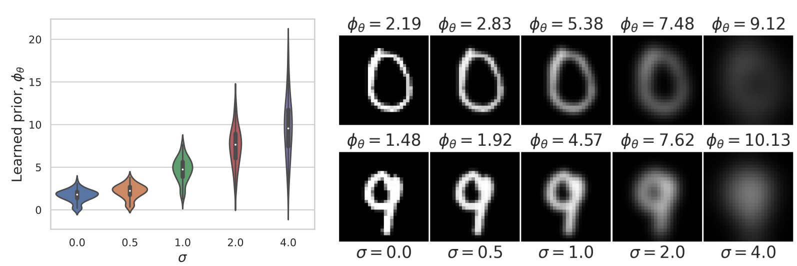
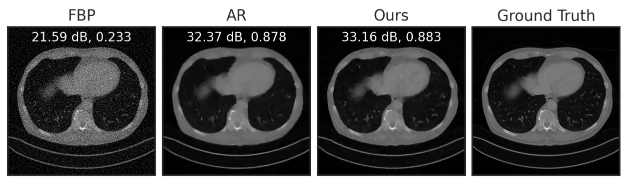


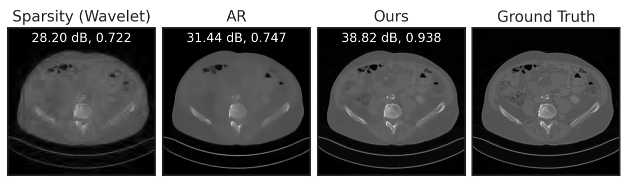
Figure 5: The prior ϕ_θ learned by LPN on MNIST, evaluated at images blurred by Gaussian kernels with an increasing standard deviation σ. Left: the prior over 100 test images. Right: the prior at individual examples.

Besides perturbing the images by Gaussian noise and convex combination in Section 4, we also blur the images by Gaussian kernels with increasing standard deviations, with results shown in Figure 5. Again, the prior increases as the images become blurrier, coinciding with the distribution of real hand-written digit images.

G.3 Inverse problems on Mayo-CT



(a) Sparse-view tomographic reconstruction.

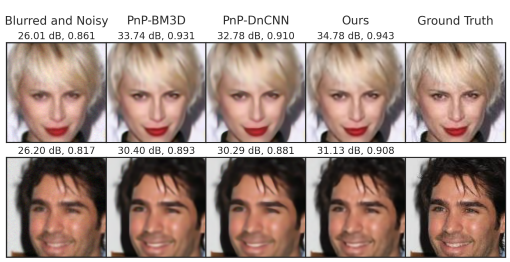


(b) Compressed sensing (compression rate $= 1/16$).

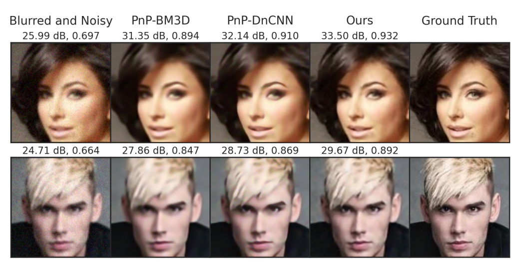
Figure 6: Inverse problems on Mayo-CT.

G.4 Deblurring on CelebA

We showcase the capability of LPN for a realistic inverse problem: deblurring on CelebA. We employ the PnP-ADMM framework for deblurring, and compare with other state-of-the-art PnP approaches: PnP-BM3D [103, 29] and PnP-DnCNN [110, 108]. Table 3 and Figure 7 demonstrate superior performance of LPN compared to state-of-the-art methods, while allowing for explicit evaluation of the used prior.



(a) $\sigma_{blur} = 1.0, \sigma_{noise} = 0.02$.



(b) $\sigma_{blur} = 1.0, \sigma_{noise} = 0.04$.

Figure 7: Deblurring results on CelebA using ADMM-based plug-and-play with different denoisers (BM3d, DnCNN, and our LPN), for different Gaussian blur kernel standard deviation σ_{blur} and noise standard deviation σ_{noise}. PSNR and SSIM are presented above each prediction.

Table 3: Deblurring on CelebA. Results are averaged over 20 test images.

METHOD	$\sigma_{blur} = 1$, $\sigma_{noise} = .02$		$\sigma_{blur} = 1$, $\sigma_{noise} = .04$		$\sigma_{blur} = 2$, $\sigma_{noise} = .02$		$\sigma_{blur} = 2$, $\sigma_{noise} = .04$	
	PSNR(↑)	SSIM(↑)	PSNR(↑)	SSIM(↑)	PSNR(↑)	SSIM(↑)	PSNR(↑)	SSIM(↑)
Blurred and Noisy	27.0 ± 1.6	$.80 \pm .03$	24.9 ± 1.0	$.63 \pm .05$	24.0 ± 1.7	$.69 \pm .04$	22.8 ± 1.3	$.54 \pm .04$
PnP-BM3D [103]	31.0 ± 2.7	$.88 \pm .04$	29.5 ± 2.2	$.84 \pm .05$	28.5 ± 2.2	$.82 \pm .05$	27.6 ± 2.0	$.79 \pm .05$
PnP-DnCNN [110]	30.7 ± 2.5	$.87 \pm .04$	30.3 ± 2.2	$.86 \pm .04$	28.2 ± 2.0	$.80 \pm .05$	28.0 ± 2.0	$.80 \pm .05$
Ours	$\mathbf{31.7 \pm 2.9}$	$\mathbf{.90 \pm .04}$	$\mathbf{31.1 \pm 2.5}$	$\mathbf{.89 \pm .04}$	$\mathbf{28.8 \pm 2.2}$	$\mathbf{.83 \pm .05}$	$\mathbf{28.5 \pm 2.1}$	$\mathbf{.82 \pm .05}$