SIAM J. IMAGING SCIENCES
Vol. 17, No. 1, pp. 91–115

Learning Weakly Convex Regularizers for Convergent Image-Reconstruction Algorithms*

Alexis Goujon†, Sebastian Neumayer†, and Michael Unser†

Abstract. We propose to learn non-convex regularizers with a prescribed upper bound on their weak-convexity modulus. Such regularizers give rise to variational denoisers that minimize a convex energy. They rely on few parameters (less than 15,000) and offer a signal-processing interpretation as they mimic handcrafted sparsity-promoting regularizers. Through numerical experiments, we show that such denoisers outperform convex-regularization methods as well as the popular BM3D denoiser. Additionally, the learned regularizer can be deployed to solve inverse problems with iterative schemes that provably converge. For both CT and MRI reconstruction, the regularizer generalizes well and offers an excellent tradeoff between performance, number of parameters, guarantees, and interpretability when compared to other data-driven approaches.

Key words. inverse problems, denoising, data-driven priors, weak convexity, bilevel optimization, splines

MSC codes. 26B25, 47A52, 49N45, 68U10, 65D07, 68T05, 90C26

DOI. 10.1137/23M1565243

1. Introduction.

Linear inverse problems are ubiquitous in imaging, with applications in medical imaging [36], including magnetic-resonance imaging (MRI) and X-ray computed tomography (CT). In a discretized linear inverse problem [48], the goal is to reconstruct an (unknown) image of interest $\mathbf{x} \in \mathbb{R}^d$ from a given noisy observation

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \in \mathbb{R}^m, \tag{1.1}$$

where $\mathbf{H} \in \mathbb{R}^{m \times d}$ denotes the measurement operator and $\mathbf{n} \in \mathbb{R}^m$ is a noise term. To overcome a possibly ill-conditioned $\mathbf{H}$ and the presence of noise, it is standard to compute the reconstruction $\hat{\mathbf{x}}$ as a solution of the variational problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{2}\|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2 + R(\mathbf{x}), \tag{1.2}$$

where the regularizer $R\colon \mathbb{R}^d \to \mathbb{R}$ incorporates prior information about $\mathbf{x}$. Convex regularizers, such as the Tikhonov [54] or total-variation (TV) [50, 15] ones, are popular as they allow one to efficiently solve (1.2). Unfortunately, such regularizers do not yield state-of-the-art reconstructions and have known limitations. For instance, they typically struggle to preserve textures in the image $\mathbf{x}$ [43].

*Received by the editors April 14, 2023; accepted for publication (in revised form) October 20, 2023; published electronically January 18, 2024.
https://doi.org/10.1137/23M1565243
Funding: The research leading to this publication was supported by the European Research Council (ERC) under European Union's Horizon 2020 (H2020), grant agreement 101020573 FunLearn, and by the Swiss National Science Foundation, grant 200020 184646/1.
†Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne PFLCH-1015 Lausanne (alexis.goujon@epfl.ch, sebastian.neumayer@epfl.ch, michael.unser@epfl.ch).

1.1. The convex non-convex framework for denoising. The reliance on a well-chosen non-convex R leads to improved performance [49, 11], with the caveat that finding a global minimum of (1.2) becomes intractable in general. A possible remedy is provided by the convex non-convex (CNC) framework. It consists in the deployment of a non-convex R_{CNC} such that the global objective

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}\|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2 + R_{\mathrm{CNC}}(\mathbf{x}) \tag{1.3}$$

is convex; see [31] for an overview. Over the past few years, the use of CNC approaches has led to improved results in various settings, including dictionary learning [53], plug-and-play (PnP) algorithms [34, 24], and matrix completion [1].

For the case of image denoising, namely, $\mathbf{H} = \mathbf{I}$, the data-fidelity term $\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$ is 1-strongly convex. Hence, to ensure the convexity of $\mathcal{J}$, the regularizer R_{CNC} needs to be 1-weakly convex from the definition of weak convexity (see section 2). Various strategies have been proposed to design a weakly convex R_{CNC}.

Explicit design. The commonly used $\|\cdot\|_1$-norm for sparse regularization can be replaced by a non-convex penalty function that better mimics the behavior of the $\|\cdot\|_0$-norm, while ensuring that one remains within the CNC framework. This includes properly scaled versions of the logarithm and the minimax concave penalty [30]. Although non-convex, these functions are quasi-convex. In particular, they are such that large values are more penalized than smaller ones. The potentials are then combined with convolutional filters. This yields, for instance, TV-like regularizers [57, 51], which extend and improve upon their convex counterparts.

Implicit design with Moreau envelopes. There is a systematic method to convert any convex regularizer into a non-convex but still 1-weakly convex one, utilizing its (generalized) Moreau envelope [1, 31]. Such a regularizer, however, does not admit a closed form, and the existing algorithms to solve (1.2) involve a computationally intensive bilevel optimization task.

Implicit design via the learning of proximal operators. Although R_{CNC} is non-convex, its proximity operator $\mathrm{prox}_{R_{\mathrm{CNC}}}$ is well-defined under mild conditions [20]. In [24], the authors propose to directly learn $\mathrm{prox}_{R_{\mathrm{CNC}}}$ such that it is a good Gaussian denoiser. To do so, they explicitly parameterize the proximal operator, in line with the recently introduced gradient-step denoisers [12, 23]. More precisely, they express the residual map $(\mathrm{prox}_{R_{\mathrm{CNC}}} - \mathrm{Id})$ as the gradient of a deep convolutional neural network (CNN), and require that the residual is contractive by enforcing that it has a Lipschitz constant smaller than 1. This yields excellent performance, with the caveat that it is challenging to enforce strict Lipschitz constraints on the gradient of a CNN. For this reason, the authors of [24] propose to regularize the spectral norm of the Jacobian of $(\mathrm{prox}_{R_{\mathrm{CNC}}} - \mathrm{Id})$ at a finite number of locations. This method works well in practice but does not offer any provable guarantee on the weak-convexity property of the underlying (implicit) objective $\mathcal{J}$.

1.2. Extension to ill-posed inverse problems. The design of CNC models is difficult when the forward matrix $\mathbf{H}$ is noninvertible. Since the data term $\frac{1}{2}\|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2$ is no longer strongly convex, the 1-weak convexity of R is not sufficient. Then the condition on R depends on $\mathbf{H}$, and CNC models are therefore usually tailored to a specific problem. One can partially circumvent this limitation by combining a proximal algorithm with a generic weakly convex regularizer, for which the proximal operator is well-defined. The convergence to stationary

points of the objective is established in [24, 22] for the forward-backward splitting [5] based on the very general convergence result for functions with the Kurdyka–Łojasiewicz (KL) property given in [3]. When R is differentiable, as will be assumed in our setting, similar results can be obtained for gradient descent applied to the non-convex objective (1.2); see [3]. From a stochastic perspective, it is known that such first-order methods do not get trapped into strict saddle points of the objective [32]. This is a possible explanation for the good empirical performance of non-convex reconstruction frameworks.

1.3. Other deep-learning-based variational methods with some guarantees. The emergence of deep-learning-based methods has led to significant improvements in the quality of reconstruction for inverse problems. Yet, due to the blackbox nature of deep NNs, this often comes with a loss of interpretability and reliability. Thus, there is a growing interest to mitigate these limitations; see [39] for a survey. In the following, we briefly comment on works that rely on the variational formulation (1.2) with a learned regularizer R but that are not directly within the CNC framework. To provide maximal theoretical guarantees within iterative image reconstruction, it was proposed in [38] to learn a convex R based on a deep CNN, and was shown in [19] that a shallow model, namely, a convex ridge regularizer NN (CRR-NN) with few parameters, was sufficient. The latter offers the opportunity to learn a collection of filters and sparsity-promoting profile functions to build R. This is inspired by the Fields-of-Experts (FoE) framework [49] and its many variants, such as [11], to design and learn a non-convex R. While [11] yields good performance, it does not guarantee that the objective is convex. Another popular extension of FoE is trainable nonlinear reaction diffusion (TNRD) [10]. There, the minimization scheme associated with (1.2) is unrolled and different filters and potential functions are learned at each step. This improves the performance over [11] but does not correspond to an energy minimization anymore. More recently, all these frameworks have been unified in the context of variational networks [28]. The combination of these with recent findings in deep CNN research and early stopping techniques has then led to the total deep variation framework [27]. Although this model has several layers, some interpretability remains possible through an eigenfunction analysis. Another deep-learning-based variational method with convergence guarantees and a regularization scheme is found in [33].

1.4. Outline and main contributions. In this work, we propose a framework to learn a 1-weakly convex regularizer that yields an interpretable proximal denoiser. The general framework is introduced in section 2. Then the principal contributions are as follows.

- **Denoising:** In section 3, we propose a scheme for the training of weakly convex ridge regularizer neural networks (WCRR-NN), with a significant increase in performance over their convex counterparts but with the same guarantees and interpretability. Based on a condition introduced in Proposition 3.2, the associated denoising problem is convex, which allows for global minimization. Numerical experiments indicate that the learning of both the profiles and the filters leads to a sparsity prior that is state of the art in the CNC framework across various noise levels for the BSD68 test set. In particular, it is the first convex-energy-based model that outperforms BM3D [13], which has been one of the most popular benchmarks for nearly 15 years now.
- **Inverse problems:** In section 4, we deploy the learned regularizer to solve generic inverse problems by minimizing (1.2) with an accelerated gradient-descent (AGD)

scheme that is tailored to our weakly convex regularizer (Algorithm 4.1). Further, we prove that the algorithm reaches some critical point of the objective (Theorem 4.3). Numerical experiments for CT and MRI demonstrate that the regularizer empirically generalizes well. We find that it outperforms several energy-based reconstruction methods that come with convergence guarantees.

Finally, conclusions are drawn in section 5.

The implementation of WCRR-NNs and pretrained models are publicly available,[1] as is their usage in solving inverse problems.[2]

2. Weakly convex regularizers. Our goal is to construct a regularizer R for the variational reconstruction model (1.2) that performs well across a variety of inverse problems, while maintaining the theoretical guarantees and interpretability of classical schemes. A particularly promising direction is given by the CNC framework, where one can efficiently find a global minimum of the objective in (1.2). As commonly done in practice, our strategy is to design and train the regularizer based on the denoising task

$$\hat{\mathbf{x}} = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + R(\mathbf{x}), \tag{2.1}$$

where $\mathbf{y}$ is a noisy version of a clean image. The minimization of (1.2) for generic inverse problems and weakly convex regularizers is then discussed in section 4.

To obtain a CNC model in (2.1), R needs to be 1-weakly convex so that the overall objective remains convex.

Definition 2.1. *A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is*

(i) convex *if $f(\lambda\mathbf{x}+(1-\lambda)\mathbf{y}) \le \lambda f(\mathbf{x}) + (1-\lambda)f(\mathbf{y})$ for all $\mathbf{x},\mathbf{y}\in\mathbb{R}^d$ and $\lambda\in[0,1]$;*

(ii) ρ-strongly convex *if $(f - \frac{\rho}{2}\|\cdot\|^2)$ is convex with $\rho \ge 0$;*

(iii) ρ-weakly convex *if $f + \frac{\rho}{2}\|\cdot\|^2$ is convex with $\rho \ge 0$.*

Note that a ρ-weakly convex R is also μ-weakly convex for any $\mu \ge \rho$. A convex R is ρ-weakly convex for any $\rho \ge 0$ and, in particular, 0-weakly convex. For a differentiable R, convexity is equivalent to the monotonicity of ∇R. Hence, a differentiable R is ρ-weakly convex iff

$$(\nabla R(\mathbf{y}) - \nabla R(\mathbf{x}))^T(\mathbf{y}-\mathbf{x}) \ge -\rho\|\mathbf{y}-\mathbf{x}\|_2^2 \tag{2.2}$$

for any $\mathbf{x},\mathbf{y}\in\mathbb{R}^d$. Given a twice-differentiable R, ρ-weak convexity is equivalent to

$$H_R(\mathbf{x}) \succeq -\rho\mathbf{I} \tag{2.3}$$

for any $\mathbf{x}\in\mathbb{R}^d$, where $H_R(\mathbf{x})$ denotes the Hessian of R at $\mathbf{x}$. In other words, the Hessian of a ρ-weakly convex function has all its eigenvalues in the range $[-\rho,+\infty)$.

Remark 2.2. Any differentiable function R with L-Lipschitz gradient is L-weakly convex. This estimate is, however, not necessarily tight, in the sense that R might also be ρ-weakly

[1] https://github.com/axgoujon/weakly_convex_ridge_regularizer
[2] https://github.com/axgoujon/convex_ridge_regularizers

convex for some $0 \leq \rho \ll L$ all the way to zero. For instance, any convex R with L-Lipschitz gradient is L-weakly convex, but it is also trivially 0-weakly convex because it is equivalent to being convex.

Weak convexity provides more flexibility, while still maintaining most of the desirable properties of usual convex-regularization frameworks. In particular, the proximal operator

$$\operatorname{prox}_R(\mathbf{y}) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + R(\mathbf{x}) \tag{2.4}$$

is well-defined for any ρ-weakly convex R with $\rho < 1$. Indeed, the objective in (2.4) is $(1-\rho)$-strongly convex, which ensures the existence of a unique minimizer. The properties of the proximal operator in a generic non-convex setting are characterized in detail in [20]. The main implication here is the Lipschitz continuity of our denoiser (2.4) (Proposition 2.3).

Proposition 2.3 ([20]). *For any ρ-weakly convex regularizer R with $\rho < 1$, there exists a convex lower semicontinuous potential $g\colon \mathbb{R}^d \to \mathbb{R}$ such that $\operatorname{prox}_R(\mathbf{x}) \in \partial g(\mathbf{x})$ holds for every $\mathbf{x} \in \mathbb{R}^d$. Conversely, the subgradient of any such g coincides with prox_R for some R that is 1-weakly convex on any convex subset of its domain. Furthermore, prox_R is $(\frac{1}{1-\rho})$-Lipschitz, in the sense that*

$$\|\operatorname{prox}_R(\mathbf{y}_2) - \operatorname{prox}_R(\mathbf{y}_1)\|_2 \leq \frac{1}{1-\rho}\|\mathbf{y}_2 - \mathbf{y}_1\|_2 \tag{2.5}$$

for any $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^d$. More generally, C^{k+1} regularity of the potential g leads to C^k regularity of prox_R. Finally, prox_R is invertible on its range in this setting.

For a convex regularizer R, $\rho = 0$ and, hence, prox_R is non-expansive (1-Lipschitz). For a non-convex but weakly convex R, namely, $\rho > 0$, this is not necessarily the case anymore. We conjecture that this is key to the boost in performance since non-expansive denoisers have intrinsic limitations; see, for instance, [19, Figure 1].

3. Design of a learnable and provably 1-weakly convex regularizer for denoising.

In this section, we discuss the construction and training of R, and compare it with several other variational frameworks.

3.1. Regularizer architecture.

The weakly convex regularizer R is chosen as the sum of convolutional ridges

$$R\colon x[\cdot] \mapsto \sum_{i=1}^{N_C} \sum_{\mathbf{k}\in\mathbb{R}^2} \psi_i((h_i * x)[\mathbf{k}]), \tag{3.1}$$

where $x[\cdot]$ represents a 2D image, $(h_i[\cdot])_{i=1}^{N_C}$ are the impulse responses of a collection of linear and shift-invariant filters, and $(\psi_i)_{i=1}^{N_C}$ are potential functions with Lipschitz continuous derivative. In practice, the finite-size input images are zero-padded so that the outputs of the convolutions have the same spatial size as the input image. We also choose potential functions with a shared profile ψ, so that $\psi_i = \alpha_i^{-2}\psi(\alpha_i\cdot)$ with $\alpha_i > 0$. As the $h_i[\cdot]$ can absorb the α_i in the definition of ψ_i, this is just a different parameterization for adding weights α_i^{-2} in front of the profile ψ. The advantage of our parameterization is that $\operatorname{Lip}(\psi_i')$ does not depend on α_i,

which will simplify the reasoning throughout this section. The number N_C of filters is also referred to as the number of channels or feature maps of the model. The motivation behind our choice is threefold.

- **Interpretability:** Model (3.1) includes many traditional sparsity-promoting regularizers and, as shown, in section 3.5, the trained regularizer will have a simple signal-processing interpretation. The parameters for deeper CNN-based regularizers are usually much harder to interpret than those in (3.1).
- **Control of ρ:** The weak-convexity modulus of (3.1) can be upper-bounded using Proposition 3.2. This is far less obvious for deeper CNNs architectures. There, weak convexity is usually promoted via regularization during training [24]. While this works qualitatively, it does not generate provably ρ-weakly convex maps for some prescribed ρ.
- **Model expressivity:** There is evidence that, in constrained settings, (3.1) has good expressive power. For instance, when learning convex regularizers for (1.2), architectures of the form (3.1) are on par with deep CNNs such as the input convex NN (ICNN), all the while depending on much fewer parameters [19].

To simplify the notation in what follows, the regularizer (3.1) is written whenever needed in the generic form

$$R\colon \mathbf{x} \mapsto \sum_{j=1}^{d\times N_C} \psi_j(\mathbf{w}_j^T \mathbf{x}), \tag{3.2}$$

where $\mathbf{x} = (x[\mathbf{k}])_{\mathbf{k}\in\Omega} \in \mathbb{R}^d$ is the vectorized representation of $x[\cdot]$, the $\mathbf{w}_j \in \mathbb{R}^d$ correspond to shifted versions of the filter kernels, and j indexes at the same time along the channels and the 2D shifts of the kernels. The gradient of this differentiable regularizer R reads

$$\boldsymbol{\nabla} R(\mathbf{x}) = \mathbf{W}^T \boldsymbol{\varphi}(\mathbf{W}\mathbf{x}), \tag{3.3}$$

where $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_{dN_C}]^T \in \mathbb{R}^{dN_C\times d}$ and $\boldsymbol{\varphi}$ is the pointwise *activation function* given by $\boldsymbol{\varphi}(\mathbf{z}) = (\psi_j'(z_j))_{j=1}^{dN_C} = (\alpha_j^{-1}\psi'(\alpha_j z_j))_{j=1}^{dN_C}$. Note that $\mathbf{W}\mathbf{x}$ is a multichannel filtered version of the image $\mathbf{x}$. Since ψ can absorb the spectral norm of $\mathbf{W}$, we enforce that $\|\mathbf{W}\| = 1$, where $\|\cdot\|$ denotes the spectral norm, in order to remove some redundancy from the model and simplify the explanations.

In the following, we use that the Lipschitz continuity of ψ' implies differentiability of ψ' almost everywhere (Rademacher's theorem) and that the essential infimum $\operatorname{ess\,inf}_{t\in\mathbb{R}} \psi''(t)$ is well-defined and satisfies $|\operatorname{ess\,inf}_{t\in\mathbb{R}} \psi''(t)| \le \operatorname{Lip}(\psi')$.

Lemma 3.1. *Let $\psi\colon \mathbb{R}\to\mathbb{R}$ have a Lipschitz continuous derivative. Then ψ is ρ-weakly convex for any $\rho \ge s_{\inf} = \max(0, -\operatorname{ess\,inf}_{t\in\mathbb{R}} \psi''(t))$.*

Proof. The Lipschitz continuity of ψ' implies that $\psi'(t_2) - \psi'(t_1) = \int_{t_1}^{t_2} \psi''(t)\mathrm{d}t$ for any $t_1, t_2 \in \mathbb{R}$. From this, we infer that $(\psi(t_2) - \psi(t_1))(t_2 - t_1) \ge (\operatorname{ess\,inf}_{t\in\mathbb{R}} \psi''(t))(t_2 - t_1)^2$, which is precisely condition (2.2). ∎

Proposition 3.2. *Any R of the form (3.2) with $\|\mathbf{W}\| = 1$ and a ρ-weakly convex ψ is ρ-weakly convex. In particular, assuming that ψ' is Lipschitz continuous, this holds for any $\rho \ge s_{\inf}$ as defined in Lemma 3.1.*

Proof. Since $\alpha_i > 0$ and $\psi_i = \alpha_i^{-2}\psi(\alpha_i \cdot)$, the convexity of $t \mapsto \psi(t) + \frac{\rho}{2}t^2$ implies the convexity of $t \mapsto \psi_i(t) + \frac{\rho}{2}\alpha_i^{-2}(\alpha_i t)^2$. Thus, $\mathbf{x} \mapsto \psi_j(\mathbf{w}_j^T\mathbf{x}) + \frac{\rho}{2}(\mathbf{w}_j^T\mathbf{x})^2$ and $\mathbf{x} \mapsto R(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{W}\mathbf{x}\|_2^2$ are also convex. Since $\|\mathbf{W}\| = 1$ and $\rho > 0$, $\mathbf{x} \mapsto \frac{\rho}{2}(\|\mathbf{x}\|_2^2 - \|\mathbf{W}\mathbf{x}\|_2^2)$ is convex, and we infer that $\mathbf{x} \mapsto R(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x}\|_2^2$ is convex. ∎

Hence, we can obtain a 1-weakly regularizer R by enforcing that $s_{\text{inf}} \leq 1$.

Remark 3.3. The ridge decomposition (3.2) of R is also used within the CRR-NN framework [19], which involves the learning of a convex-ridge regularizer R with learnable spline potentials ψ_j. For CRR-NNs, $s_{\text{inf}} = 0$ is enforced to ensure that the ψ_j are convex. On the contrary, the present WCRR-NN model with $s_{\text{inf}} \in [0, 1]$ has more freedom and therefore extends upon [19].

The present parameterization of R is greatly inspired by [19]. However, instead of its single (non-decreasing) spline nonlinearity used in [19], we decompose the activation $\varphi = \psi'$ into the difference of two splines as

$$\varphi = \mu\,\varphi_+ - \varphi_-, \tag{3.4}$$

where $\mu \in \mathbb{R}_{\geq 0}$ is a learnable parameter and the φ_+, φ_- are trainable, non-decreasing, non-expansive linear splines. Although theoretically equivalent to the use of a single linear spline φ with $s_{\text{inf}} \leq 1$, we found the decomposition (3.4) to be more effective for the training. Theoretical motivations for using splines in a constrained NN have been proposed in [42], and a discussion of the expressivity of the resulting NN architecture can be found in [18]. Our choice ensures that the following properties are met.

- R is 1-weakly convex, which follows from Proposition 3.2;
- the Lipschitz constant is bounded as

$$\text{Lip}(\boldsymbol{\nabla} R) \leq \|\mathbf{W}\|^2 \,\text{Lip}(\varphi) \leq \max(\mu, 1). \tag{3.5}$$

In the following, we provide more parameterization details regarding the parameterization.

Parameterization of learnable linear splines. Both linear splines φ_+ and φ_- are parameterized in the same way with our spline toolbox [7, 16]. In what follows, we abbreviate their respective learnable parameters $\mathbf{c}_+$ and $\mathbf{c}_-$ by $\mathbf{c}$. We use $\varphi_{\mathbf{c}} \colon \mathbb{R} \to \mathbb{R}$ with knots $\tau_m = (m - M/2)\Delta$, $m = 0, \ldots, M$, where Δ is the spacing. For simplicity, we assume that M is even. The learnable parameter $\mathbf{c} = (c_m)_{m=0}^M \in \mathbb{R}^{M+1}$ defines the values $\varphi_{\mathbf{c}}(\tau_m) = c_m$ of $\varphi_{\mathbf{c}}$ at the knots. To fully characterize $\varphi_{\mathbf{c}}$, we extend it by the constant value c_0 on $(-\infty, \tau_0]$ and c_M on $[\tau_M, +\infty)$. Consequently, any primitive ψ of $\varphi_{\mathbf{c}}$ is piecewise quadratic on $[\tau_0, \tau_M]$ with affine extensions.

Constraints on the linear splines. To ensure that the φ_i are non-decreasing and non-expansive, we follow the strategy introduced in [19]. Let $\mathbf{D} \in \mathbb{R}^{M \times (M+1)}$ be the one-dimensional finite-difference matrix with $(\mathbf{D}\mathbf{c})_m = (c_{m+1} - c_m)$ for $m = 1, \ldots, M$. As $\varphi_{\mathbf{c}}$ is piecewise linear, it holds that

$$\varphi_{\mathbf{c}} \text{ is nondecreasing and nonexpansive} \Leftrightarrow 0 \leq (\mathbf{D}\mathbf{c})_m \leq \Delta,\ m = 1, \ldots, M. \tag{3.6}$$

To optimize over $\{\varphi_{\mathbf{c}}\colon 0 \le (\mathbf{D}\mathbf{c})_m \le \Delta,\ m = 1, \dots, M\}$, we reparameterize the linear splines as $\varphi_{\boldsymbol{P}_\uparrow(\mathbf{c})}$, where

$$\boldsymbol{P}_\uparrow(\mathbf{c}) = \mathbf{S}\mathrm{Clip}_{[0,\Delta]}(\mathbf{D}\mathbf{c}) + \mathbf{1}^T\mathbf{c} \tag{3.7}$$

is a nonlinear projection onto the feasible set (3.6). In (3.7), $\mathrm{Clip}_{[0,\Delta]}$ is the pointwise clipping operation with $\mathrm{Clip}_{[0,\Delta]}(t) = \min(\max(0,t),\Delta)$, and $\mathbf{S}$ denotes the cumulative-sum operation with $(\mathbf{S}\mathbf{d})_{m+1} = \sum_{k=1}^{m} d_k$ for $m = 0, \dots, M$ and any $\mathbf{d} \in \mathbb{R}^m$. In other words, $\boldsymbol{P}_\uparrow$ clips the finite differences between entries in $\mathbf{c}$ that are either greater than Δ or negative and sets them to the closest admissible value, while it preserves the mean due to the additional term $\mathbf{1}^T\mathbf{c}$.

Further, we enforce that φ_+ and φ_- are odd, which is natural for imaging as it results in even potentials. To get this symmetry while still satisfying (3.6), we use the change of variable $\mathbf{c} \to \frac{1}{2}(\boldsymbol{P}_\uparrow(\mathbf{c}) - \texttt{reverse}(\boldsymbol{P}_\uparrow(\mathbf{c}))$, where `reverse` flips the order of the entries of $\mathbf{c}$. Hence, all constraints are embedded into the parameterization, and the parameter $\mathbf{c}$ that is learned remains unconstrained.

Parameterization of convolutional filters. The learnable convolution layer $\mathbf{W}$ is required to be of unit norm. Hence, we parameterize $\mathbf{W}$ as $\mathbf{W} = \mathbf{U}/\|\mathbf{U}\|$, where $\mathbf{U}$ represents a convolutional layer with the same dimensions as $\mathbf{W}$. The computation of the spectral norm $\|\mathbf{U}\|$ will be described in section 3.3. To efficiently explore a large field of view (see also [19]), we decompose $\mathbf{U}$ into a composition of three zero-padded convolutions with kernels of size $(k_s \times k_s)$, k_s odd, and an increasing number of output channels. Similarly to [11], the convolution kernels are constrained to have zero mean. The equivalent (up to boundary effects) *single-convolution* layer would have a kernel of size $(K_s \times K_s)$ with $K_s = 3k_s - 2$.

3.2. Multi-noise-level denoiser.

So far, we only introduced a generic R that is not adapted to diverse noise levels. To obtain a denoiser for various noise levels σ, a common approach is to incorporate an adjustable parameter $\lambda_\sigma \in \mathbb{R}$ as in

$$\hat{\mathbf{x}} = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda_\sigma R(\mathbf{x}). \tag{3.8}$$

In principle, this leads to a noise-level-dependent regularizer $R_\sigma = \lambda_\sigma R$, but this dependence on σ turns out to be too simple to ensure good performance across multiple noise levels. Another limitation in this setting is that R_σ is λ_σ-weakly convex. Hence, for $\lambda_\sigma > 1$, one is not guaranteed to remain within the CNC framework and, for $\lambda_\sigma < 1$, we might not exploit the full freedom given by CNC models. Therefore, we instead express the parameters α_i introduced in section 3.1 as functions of the noise level[3]

$$\alpha_i(\sigma) = \mathrm{e}^{s_{\alpha_i}(\sigma)}/(\sigma + \epsilon), \tag{3.9}$$

where we set $\epsilon = 1 \cdot 10^{-5}$ to prevent instabilities for small σ. Here, s_{α_i} is a learnable linear spline with underlying parameter $\mathbf{c}_\alpha^i$, which is parameterized similarly to $\varphi_\pm$ but without constraints. The exponential parameterization in (3.9) allows for efficiently exploring a large range at training, and such a scheme is quite common in learning, e.g., in the popular TNRD

[3]In preliminary investigations, we also attempted the learning of the parameter μ as a function of the noise level, but it did not improve performance. Hence, μ is chosen to be constant across noise levels.

framework [10]. The scaling by σ in (3.9) allows for normalizing the noise distribution before the activation and was found to be very helpful in practice. Ultimately, our noise-level-dependent profile functions $\psi_i(t,\sigma) = (1/\alpha_i(\sigma))^2\psi(\alpha_i(\sigma)t)$ satisfy

$$\frac{\partial^2\psi_i}{\partial t^2}(t,\sigma) = \mu\varphi_+'(\alpha_i(\sigma)t) - \varphi_-'(\alpha_i(\sigma)t) \in [-1,+\infty). \tag{3.10}$$

Remark 3.4. The bound on the weak-convexity modulus given in Proposition 3.2 does not depend on the parameters α_i. Consequently, the addition of the α_i as learnable parameters does not compromise the weak-convexity guarantees on R.

In the remainder of the paper, $\boldsymbol{\theta}$ represents the aggregated set of learnable parameters (as detailed in section 3.3) and we use the notation $R_{\boldsymbol{\theta}}$ whenever an explicit reference to the parameters is needed. Likewise, with a slight abuse of notation, we use $R_{\boldsymbol{\theta}(\sigma)}$ to denote the regularizer at noise level σ. This noise-dependent regularizer $R_{\boldsymbol{\theta}(\sigma)}$ then yields the proximal denoiser

$$D_{\boldsymbol{\theta}(\sigma)}(\mathbf{y}) = \operatorname{prox}_{R_{\boldsymbol{\theta}(\sigma)}}(\mathbf{y}) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + R_{\boldsymbol{\theta}(\sigma)}(\mathbf{x}). \tag{3.11}$$

In general, $D_{\boldsymbol{\theta}(\sigma)}$ does not have a closed-form expression but, due to the convexity and smoothness of the underlying objective, $D_{\boldsymbol{\theta}(\sigma)}(\mathbf{y})$ can be computed efficiently with gradient-based solvers. In practice, we use AGD [41] combined with the standard gradient-based restart technique introduced in [46]. The stepsize is chosen as $1/(1+\max(1,\mu))$, which ensures convergence to a global minimizer as a consequence of the Lipschitz bound (3.5).

3.3. Training procedure.

In this section, we detail how the parameters $\boldsymbol{\theta}$ are learned so that $D_{\boldsymbol{\theta}(\sigma)}(\mathbf{y})$ is a good Gaussian denoiser across multiple noise levels.

3.3.1. Training problem.

Let $\{\mathbf{x}^m\}_{m=1}^M$ be a set of clean images. Each image $\mathbf{x}^m$ is corrupted as $\mathbf{y}^m = \mathbf{x}^m + \sigma^m\mathbf{n}^m$ with Gaussian noise $\mathbf{n}^m \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ and a noise level $\sigma^m \sim \mathcal{U}[0,\sigma_{\max}]$. Then we define the following multi-noise-level training problem:

$$\hat{\boldsymbol{\theta}} \in \operatorname*{arg\,min}_{\boldsymbol{\theta}} \sum_{m=1}^{M} \mathbb{E}_{(\mathbf{n}^m,\sigma^m)}(\|D_{\boldsymbol{\theta}(\sigma^m)}(\mathbf{y}^m) - \mathbf{x}^m\|_1). \tag{3.12}$$

Here, the ℓ_1 loss is chosen because it is known to be robust and well performing for the training of CNNs [56, 28].

3.3.2. Optimization.

For clarity, we briefly recall the various parameters contained in $\boldsymbol{\theta}$ before outlining the actual optimization procedure.

Profile-related parameters. The linear splines φ_+ and φ_- are parameterized by $\mathbf{c}_+$ and $\mathbf{c}_-$ via the constrained coefficients $\tilde{\mathbf{c}}_\pm = \frac{1}{2}(\boldsymbol{P}_\uparrow(\mathbf{c}_\pm) - \texttt{reverse}(\boldsymbol{P}_\uparrow(\mathbf{c}_\pm)))$ so that they are odd, non-decreasing, and non-expansive. Together with $\mu > 0$, (3.4) then leads to the linear-spline activation function φ. Recall that its primitive defines the profile ψ. The parameters $\mathbf{c}_\alpha^i$ specify the linear-spline functions $\alpha_i(\sigma)$, which rescale the profile ψ across the channels in (3.2) and across the noise levels.

Spectral normalization. The convolution operation represented by $\mathbf{W}$ is parameterized as $\mathbf{W} = \mathbf{U}/\|\mathbf{U}\|$, $\mathbf{U}$ consisting in the composition of 3 zero-padded convolutions. Here, $\|\mathbf{U}\|$ is computed as follows.

- **Training stage:** By assuming circular boundary conditions instead of zero-padding, we can consider $\mathbf{U}$ as a *single-convolution* layer. Then $\mathbf{U}^T\mathbf{U}$ encodes a 2D convolution from a one-channel input to a one-channel output. Hence, it can be represented by a kernel $\mathbf{K}_{\mathbf{U}^T\mathbf{U}} \in \mathbb{R}^{(2K_s-1)\times(2K_s-1)}$. In this setting, the spectrum of $\mathbf{U}^T\mathbf{U}$ can be computed using the 2D discrete Fourier transform (DFT) [52] as

$$\operatorname{spec}(\mathbf{U}^T\mathbf{U}) = \{|\mathrm{DFT}(\mathrm{Pad}_{\sqrt{d}}(K_{\mathbf{U}^T\mathbf{U}}))_{k_1k_2}| \colon 1 \le k_1, k_2 \le \sqrt{d}\}, \tag{3.13}$$

  where $\mathrm{Pad}_{\sqrt{d}}$ zero-pads $\mathbf{K}_{\mathbf{U}^T\mathbf{U}}$ into a $(\sqrt{d}\times\sqrt{d})$ image. We rely on (3.13) to estimate $\|\mathbf{U}\| \simeq \max(\operatorname{spec}(\mathbf{U}^T\mathbf{U}))$ during training since it is efficient to compute and can be incorporated into the computational graph of the computation of $\boldsymbol{\nabla} R$.
- **Test stage:** Subsequently, when evaluating a trained WCRR-NN model, the true $\|\mathbf{U}\|$ is computed with high precision using the power method (1000 steps). This firm normalization guarantees the 1-weak convexity of the underlying R (up to numerical imprecision).

Implicit differentiation. The learning of the proximal denoiser comes with the challenge that $D_{\boldsymbol{\theta}(\sigma)}$ depends implicitly on $\boldsymbol{\theta}$. As shown in the deep-equilibrium (DEQ) framework [4], it is possible to compute the Jacobian $J_{\boldsymbol{\theta}} D_{\boldsymbol{\theta}(\sigma)}$ of the denoiser with respect to the parameters via implicit differentiation. For this purpose, two steps are required.

- **Image denoising:** First, given a noisy input $\mathbf{y}^m$, one needs to perform the forward pass, which consists in the computation of $\hat{\mathbf{x}} = D_{\boldsymbol{\theta}(\sigma^m)}(\mathbf{y}^m)$. The deployed AGD is run until the relative change of norm between consecutive iterates is lower than 10^{-4}.
- **Gradient computation:** We use the DEQ implementation introduced in [4] and now briefly discuss the general concept within our setting. The differentiability of R implies that the denoised images satisfy

$$\hat{\mathbf{x}}(\boldsymbol{\theta}) - \mathbf{y} + \boldsymbol{\nabla}_{\mathbf{x}} R(\boldsymbol{\theta}, \hat{\mathbf{x}}(\boldsymbol{\theta})) = \mathbf{0}, \tag{3.14}$$

  where the dependence on σ is dropped for clarity and the dependence on $\boldsymbol{\theta}$ is made explicit. The application of the implicit-function theorem for (3.14) leads to

$$(\mathbf{I} + \mathbf{H}_R(\boldsymbol{\theta}, \hat{\mathbf{x}}(\boldsymbol{\theta}))) J_{\boldsymbol{\theta}} \hat{\mathbf{x}}(\boldsymbol{\theta}) = J_{\boldsymbol{\theta}}(\boldsymbol{\nabla}_{\mathbf{x}} R)(\boldsymbol{\theta}, \hat{\mathbf{x}}(\boldsymbol{\theta})). \tag{3.15}$$

  Hence, we evaluate the matrix-vector products with $(J_{\boldsymbol{\theta}} D_{\boldsymbol{\theta}}(\mathbf{y}))^T = (J_{\boldsymbol{\theta}} \hat{\mathbf{x}}(\boldsymbol{\theta}))^T$ (which are required for computing the gradients of (3.12) within the backpropagation algorithm) by solving a simple linear system. This is carried out with the Anderson routine given in [4]. While deriving $J_{\boldsymbol{\theta}}(\boldsymbol{\nabla}_{\mathbf{x}} R)$ is cumbersome and usually left to automatic differentiation, we use the explicit expression

$$\mathbf{H}_R(\hat{\mathbf{x}})\mathbf{u} = \mathbf{W}^T(\boldsymbol{\varphi}'(\mathbf{W}\hat{\mathbf{x}}) \odot (\mathbf{W}\mathbf{u})), \tag{3.16}$$

  where $\odot$ is the Hadamard product and the piecewise-constant function $\boldsymbol{\varphi}'$ is analytically derived from the B-spline representation of the linear spline φ. This yields the same results as automatic differentiation, but was found to be more efficient.

Optimization. The non-convex training problem in (3.12) is solved with the stochastic Adam optimizer [25], where we sample for each batch the $\mathbf{x}^m$, the corresponding noise-level σ^m, and the noise $\mathbf{n}^m$. Note that, within each batch, images are corrupted with different noise levels and, likewise, in different epochs different noise levels can be applied to the same $\mathbf{x}^m$.

3.4. Training and denoising performance. The proposed weakly convex regularizer is learned over the Gaussian-denoising task described in section 3.3, with $\sigma_{\max} = 30/255$. The same procedure as in [8] is used to form 238,400 grayscale patches[4] of size (40×40) from 400 images of the BSD500 dataset [2], while 12 other images are kept for validation. In accordance with the ablation study reported in Tables 3.1 and 3.2, the three filters in $\mathbf{U}$ have kernels with $k_s = 5$ and 4, 8, and 60 output channels, respectively. The linear splines φ_i have $M + 1 = 101$ equally distant knots with $\Delta = 2 \cdot 10^{-3}$. We initially set $\mathbf{c}_+ = \mathbf{0}$ and $(\mathbf{c}_-)_m = \tau_m$, which was found to be important to help the training. Intuitively, this choice helps the regularizer use weak convexity, which is only permitted through $\mathbf{c}_-$. The linear splines parameterizing $\alpha_i(\sigma)$ have 11 equally distant knots in the range $[0, \sigma_{\max}]$ and are initialized with the constant value 5. Our model is trained with the Adam optimizer for 6000 steps with batches of size 128, which takes less than 2 hours on a Tesla V100 GPU. The learning rates are initially set to $5 \cdot 10^{-2}$ for μ, to $5 \cdot 10^{-3}$ for $\mathbf{U}$ and $\mathbf{c}_\alpha^i$, and to $5 \cdot 10^{-4}$ for $\mathbf{c}_+$ and $\mathbf{c}_-$. Then they are decayed by 0.75 every 500 batches. For evaluation, the denoising (3.11) is performed with AGD and a tolerance of 10^{-4} for the relative change of norm between consecutive iterates. An example of convergence curves is provided in Figure 3.1.

The numerical evaluation of our WCRR-NNs and several other methods on the BSD68 test set is provided in Table 3.3. The task is non-blind, in the sense that the noise level is used either directly as an input (as in BM3D) or indirectly via a regularization parameter that is tuned on a corresponding validation set (as in TV). The first important observation is that WCRR-NNs, which implement a convex energy, outperform the popular BM3D denoiser [13]. To the best of our knowledge, this is the first time a (learnable) convex model surpasses BM3D. A visual comparison of BM3D and (W)CRR-NNs is provided in Figure 3.2. The

Table 3.1
WCRR-NN: PSNR on BSD68 vs. number of filters.

N_c	10	20	40	60	80
$\sigma = 5/255$	37.43	37.59	37.63	37.66	37.66
$\sigma = 15/255$	30.85	31.15	31.19	31.21	31.20
$\sigma = 25/255$	28.16	28.62	28.67	28.68	28.69

Table 3.2
WCRR-NN: PSNR on BSD68 vs. kernel size.

k_s	3	5	7
K_s	7	13	19
$\sigma = 5/255$	37.64	37.66	37.65
$\sigma = 15/255$	31.14	31.21	31.21
$\sigma = 25/255$	28.56	28.68	28.68

[4]WCRR-NNs are fully convolutional and can process input of any spatial size.

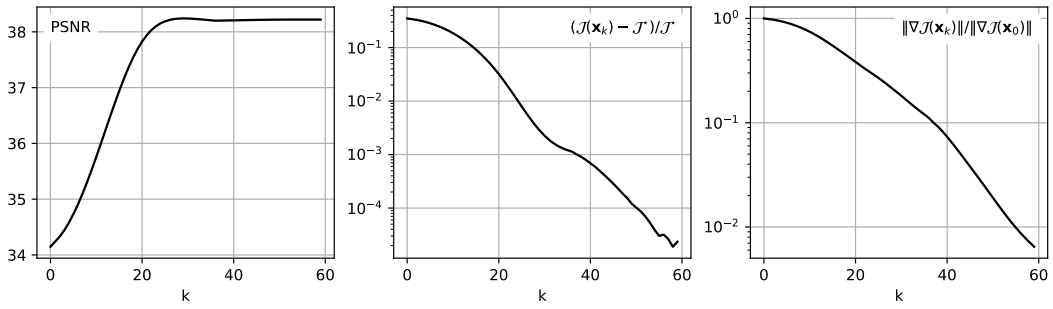

Figure 3.1. *Example of convergence curves for denoising with the WCRR-NN and AGD.*

Table 3.3
Denoising performance on the BSD68 test set.

		$\sigma = 5/255$	$\sigma = 15/255$	$\sigma = 25/255$
Convex	TV[1,2,3] [50]	36.41	29.90	27.48
	Higher-order MRFs convex [1,2,3] [11]	-	30.45	28.04
	CRR-NN[1,2,3] [19]	36.96	30.55	28.11
Provably CNC	TV CNC[1,2]	36.53	29.92	27.49
	WCRR-NN[1,2]	37.68	31.22	28.69
Approx. CNC	Prox-DRUNet	37.98	31.70	29.18
Others	Higher-order MRFs [1] [11]	-	31.22	28.70
	BM3D [13]	37.54	31.11	28.60

[1]Ridge-based regularizer. [2]Minimization of convex functional. [3]Convex regularizer.

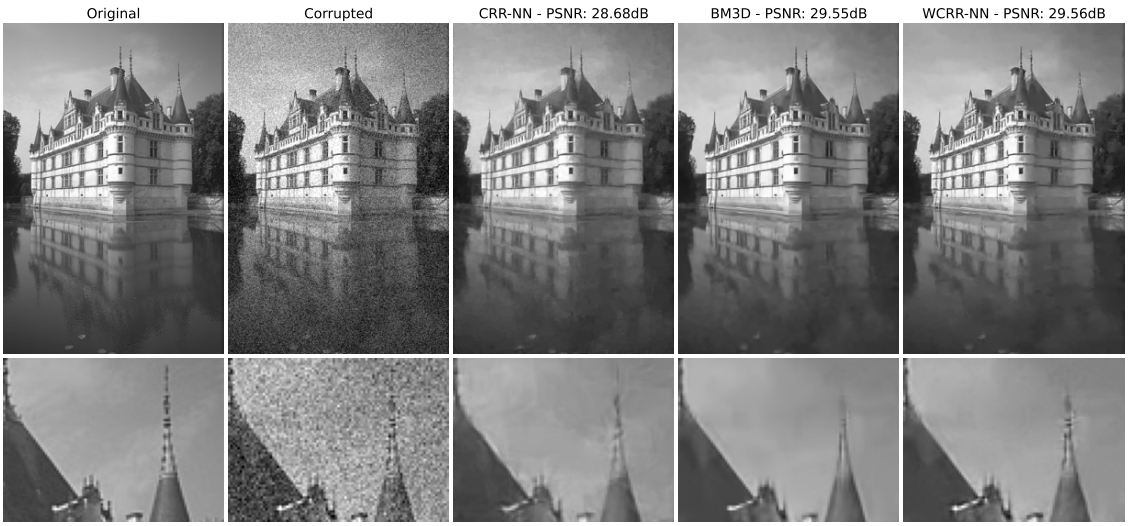

Figure 3.2. *Denoising of the "castle" image from the BSD68 test set for noise level $\sigma = 25$.*

results obtained with the second and sixth methods in Table 3.3 on the same image can be found in the original paper [11]. Next, we discuss in more depth the frameworks from Table 3.3 that are close in spirit to WCRR-NNs.

CNC-based total variation. The WCRR-NN model is inspired by earlier works that extend TV denoising to the CNC framework using non-convex potential functions [57, 51]. The publicly available implementations outperform TV for specific classes of images, typically for cartoon-like ones with sharp edges. However, we did not observe any significant improvements for the denoising of the natural images in BSD68. Hence, in our comparison, we used our own version of CNC-TV, which was obtained by training a WCRR-NN with two fixed filters, namely, the horizontal and vertical finite differences. This corresponds to an anisotropic TV denoising with learned profiles—the CNC counterpart of the standard anisotropic TV denoising model. As reported in Table 3.3, this only yields marginal improvements over TV.

Fields of experts and higher-order Markov random fields (MRFs). The FoE approach corresponds to learning the filters associated with a regularizer of the form (3.2) with hand-picked profile functions [49]. It was successfully applied in [11], with both convex and non-convex profiles. A key difference with WCRNNs lies in the theoretical guarantees: The non-convex profiles are unconstrained in [49, 11]. Hence, the objective function is not provably convex. This means that the optimization is delicate and the convergence to a global optimum cannot be guaranteed. Interestingly, WCRR-NN offer the same performance as in [11] while minimizing a convex energy.

CRR-NNs. Our work extends the convex regularizers learned with CRR-NNs [19]. The substitution of weak convexity for convexity makes a significant difference as it yields a gain of at least 0.6dB for all noise levels (see Table 3.3). In contrast with the simpler 2-filter TV setting, the improvement is substantial. This indicates that the learning of sufficiently many filters is necessary to fully exploit the additional freedom provided by weak convexity.

Gradient-step denoisers. Proposition 2.3 allows one to (implicitly) construct non-convex regularizers R by learning their proximal operator. This result is exploited in [24], where $\operatorname{prox}_R = \boldsymbol{\nabla}\psi$ is parameterized through the potential $\psi = \frac{1}{2}\|\cdot\|^2 + g$, where $\boldsymbol{\nabla} g$ must be contractive. To leverage the power of deep-learning, the authors choose $g = \frac{1}{2}\|\cdot - \text{DRUNet}(\cdot)\|_2^2$, where DRUNet [55] is a deep CNN with ~ 17 million parameters. As there are currently no efficient methods to globally bound the Lipschitz constant of the gradient of a deep CNN, they propose to instead regularize the norm of the Jacobian of $\boldsymbol{\nabla} g$ at finitely many locations during training. This yields the Prox-DRUNet denoiser, which performs very well in practice (see Table 3.3[5]). Note that Prox-DRUNet only approximately satisfies the conditions to be a truly CNC method because $\|\mathbf{H}_g(\mathbf{x})\|$ can be greater than 1 for some $\mathbf{x}$, meaning that $\boldsymbol{\nabla} g$ is not contractive (as already reported in [24]). On noisy BSD68 images, we found[6] that $\|\mathbf{H}_g\|$ can be as large as 1.07 ($\sigma = 5$), 1.08 ($\sigma = 15$), 1.18 ($\sigma = 25$), and on a set of 68 random images (i.i.d. uniformly distributed pixels in $[0, 1]$) as large as 1.69 ($\sigma = 5$), 1.20 ($\sigma = 15$), and 1.43 ($\sigma = 25$). Overall, we believe that WCRR-NNs and Prox-DRUNet offer a very complementary perspective. In fact, the good performance of Prox-DRUNet suggests that there could even be some room for further improvements with provably CNC methods.

[5]The Prox-DRUNet denoiser given in [24] is trained on color images. For grayscale denoising, we plug the image into all three color channels, average the output across the channels, and tune the denoising strength parameter σ to optimize performance. As expected, the obtained metrics are on par with DnCNN for $\sigma = 25$ and with the gradient-step denoiser for $\sigma = 5$ in [23], which indicates the appropriateness of the usage.

[6]We computed $\|\mathbf{H}_g\|$ with a precise power method (300 iterations).

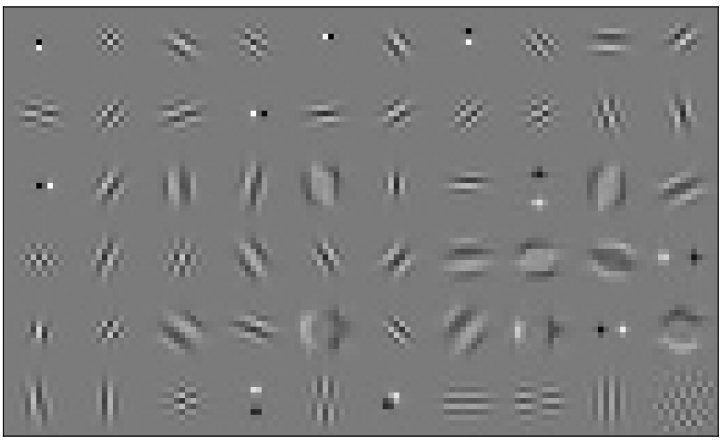

Figure 3.3. *Impulse response of the filters in the learned WCRR-NN.*

3.5. Interpretation as sparsity prior. The filters and profile functions learned for our WCRR-NNs are shown in Figures 3.3 and 3.5, respectively.

Filters. The impulse responses of the filters in $\mathbf{W}$ present patterns akin to wavelets and Gabor filters, in that they come in various modulations, orientations, and scales. In addition, the kernel $\mathbf{K}$ corresponding to the convolution $\mathbf{W}^T\mathbf{W}$ is very close to the 2D discrete Kronecker impulse, meaning that $\mathbf{W}$ is almost a Parseval frame ($\mathbf{W}^T\mathbf{W} \simeq \mathbf{I}$). A key difference, however, is that $\mathbf{K}$ is zero-mean. We also observed that more filters than in the convex setting of CRR-NNs are needed to reach the maximal performance. The payoff is that the filters are now able to capture more complicated patterns.

Profile functions. The learned profiles ψ_i are shared among the filters and then individually rescaled with the α_i, so that the ψ_i have the same shape. Hence, only their prototype ψ is discussed here. The latter converges to a quasi-convex function (i.e., sub-level sets are intervals) even without our explicitly imposing this constraint. Moreover, ψ fully exploits the 1-weak convexity of the regularizer R in the sense that $\min_t \psi''(t) = -1$. Hence, this is an active constraint since R would not satisfy it by default. Overall, ψ closely resembles the minimax concave penalty function [30].

To extend our model, we also experimented with learning a different ψ_i for each filter. This led to less interpretable profiles (not necessarily quasi-convex and with some oscillations), while it only offered a negligible gain in performance: less than 0.05dB on the denoising experiment for noise levels $\sigma \in \{5/255, 15/255, 25/255\}$.

Noise-dependent scaling. The only part of R that depends on the noise level σ is the profiles ψ_i, which depend on σ through the $\alpha_i(\sigma)$. As can be seen in Figure 3.4, the $1/\alpha_i$ are on average linear functions of σ. Loosely speaking, most profiles will roughly have the form $\psi_i(t, \sigma) \simeq \sigma^2 \psi(t/\sigma)$. To verify that such a simple dependence of R on σ is sufficient, 3 WCRR-NNs were trained to denoise at a single noise level ($\sigma \in \{5/255, 15/255, 25/255\}$). As these models do not outperform the multi-noise-level WCRR-NN on BSD68, the simple rescaling of the profiles appears to suffice.

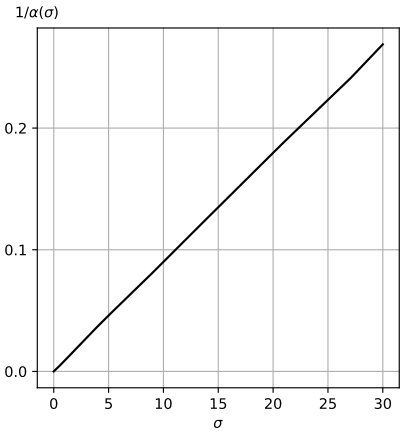

Figure 3.4. *Plot of* $1/\alpha(\sigma)$ *vs.* σ*, where* $\alpha(\sigma) = \sum_{i=1}^{N_C} \alpha_i(\sigma)$ *encodes the average behavior across the channels.*

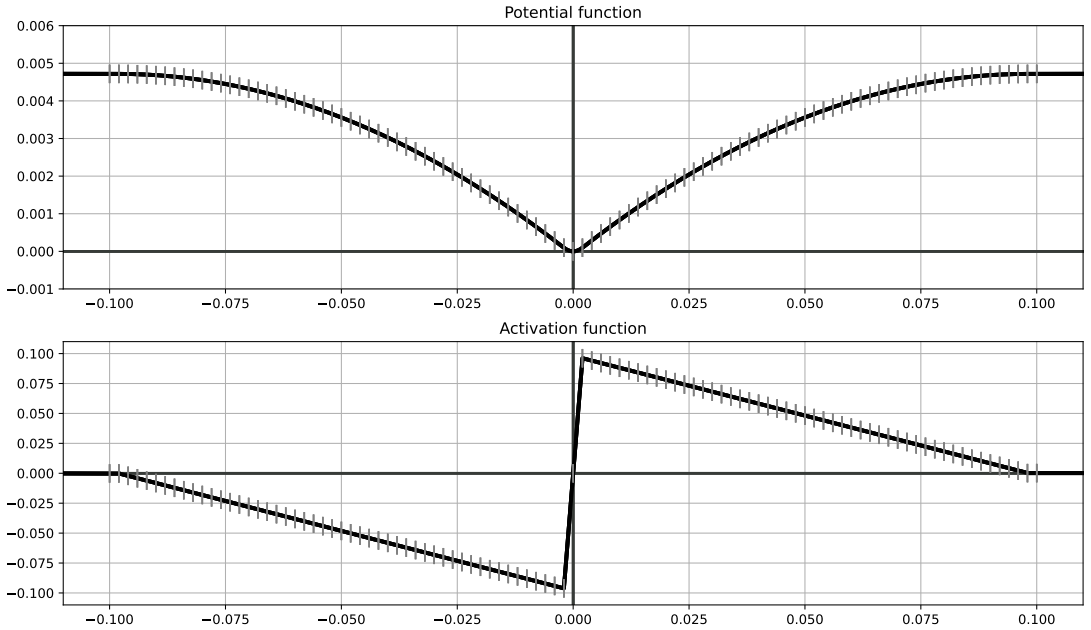

Figure 3.5. *Potential function* ψ *and activation function* $\varphi = \psi'$ *of the learned WCRR-NN. These functions are splines of degrees* 2 *and* 1*, respectively. The vertical markers indicate the control points of the splines.*

Signal-processing perspective. The regularizer R is trained to promote natural images. The corresponding gradient-descent step[7] $\mathbf{x} \mapsto \mathbf{x} - \nabla R(\mathbf{x})/\mathrm{Lip}(\nabla R) = \mathbf{x} - \mathbf{W}^T \boldsymbol{\varphi}(\mathbf{W}\mathbf{x})/\|\varphi'\|_\infty$, which should increase the regularity of images, is therefore expected to remove features considered as noise in natural images. In turn, we then expect that $\mathbf{x} \mapsto \mathbf{W}^T \boldsymbol{\varphi}(\mathbf{W}\mathbf{x})/\|\varphi'\|_\infty$ extracts some noise. Due to its shape (see Figure 3.5), the function $\varphi/\|\varphi'\|_\infty$ preserves the small responses $\mathbf{W}\mathbf{x}$ to the filters (it is almost the identity for small inputs) and cuts the large ones (it is almost the zero function for large inputs). Hence, one reconstructs the

[7] In our setting with no biases and where φ has a maximum slope at the origin, it can be shown that $\mathrm{Lip}(\mathbf{W}^T \boldsymbol{\varphi}(\mathbf{W}\cdot)) = \|\varphi'\|_\infty$.

estimated noise $\mathbf{W}^T \boldsymbol{\varphi}(\mathbf{W}\mathbf{x})/\|\varphi'\|_\infty$ by essentially removing the components of $\mathbf{x}$ that exhibit a significant correlation with the kernels. This allows for a more efficient noise extraction than done by the monotonic clipping function learned in the convex regularization framework of CRR-NNs; see [19, Figures 5 and 6]. While the monotonic clipping also preserves the small inputs, it is unable to fully remove the large responses because of the monotonicity constraint stemming from the convexity of the underlying potential.

In addition to the above perspective, we can make a link with wavelet- or framelet-like denoising [14, 29, 9, 6, 47]. Indeed, given that $\mathbf{W}^T\mathbf{W} \simeq \mathbf{I}$, the gradient-descent step can be approximated as $\mathbf{x} \mapsto \mathbf{x} - \mathbf{W}^T \boldsymbol{\varphi}(\mathbf{W}\mathbf{x})/\|\varphi'\|_\infty \simeq \mathbf{W}^T\phi(\mathbf{W}\mathbf{x})$ with $\phi = \mathrm{Id} - \varphi/\|\varphi'\|_\infty$. Since ϕ is zero around the origin and is the identity for sufficiently large inputs, it qualitatively stands between the soft- and hard-thresholding functions that have been key components for wavelet and framelet denoising for years. Finally, note that framelet-denoising models are themselves closely related to proximal operators [21].

4. Extension to generic inverse problems.

We now use the regularizer $R_{\boldsymbol{\theta}(\sigma)}$ trained in section 3 to solve inverse problems based on the variational formulation (1.2). Here, the key challenge is the possible non-convexity of the objective, which prevents us from minimizing (1.2) globally. It is, however, possible to search for critical points. These are still of particular interest, especially because the regularizer has a simple structure with an *almost* convex energy landscape.

Proximal vs. gradient methods. The standard PnP frameworks rely on proximal-based methods with an explicit denoising step. The motivation there is that the denoising step is typically efficient to perform, while neither the regularizer (if it exists) is explicitly known, nor is its gradient. In our setting, on the contrary, it is very efficient to evaluate the regularizer and its gradient, and hence AGD methods [41], which are applicable to general non-convex problems [17], are better suited. In our setting, AGD is also known to attain optimal convergence rates among first-order methods. In what follows, we recall the main features of AGD and show how to leverage the knowledge of the weak-convexity modulus of the objective.

4.1. Accelerated gradient descent.

To solve the inverse problem, we minimize the regularized objective

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}\|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda R_{\boldsymbol{\theta}(\sigma)}(\mathbf{x}), \tag{4.1}$$

where $R_{\boldsymbol{\theta}(\sigma)}$ is the 1-weakly convex regularizer from section 3.4 and $\lambda > 0$ is a regularization parameter. Since the objective is differentiable, we can rely on gradient-based methods to find critical points of $\mathcal{J}$ as a convenient alternative to proximal algorithms. To reduce the reconstruction time, we propose an AGD variant in Algorithm 4.1, which is tailored to λ-weakly convex functionals $\mathcal{J}$ with L-Lipschitz-continuous gradient.

From (3.5), we infer that $\boldsymbol{\nabla}\mathcal{J}$ is L-Lipschitz-continuous with $L \leq \|\mathbf{H}\|^2 + \lambda \max(\mu, 1)$, which implies for $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$ the standard upper estimate

$$\mathcal{J}(\mathbf{x}_1) \leq \mathcal{J}(\mathbf{x}_2) + \boldsymbol{\nabla}\mathcal{J}(\mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2) + \frac{L}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2. \tag{4.2}$$

As $R_{\boldsymbol{\theta}(\sigma)}$ is 1-weakly convex, $\mathcal{J}$ is λ-weakly convex. Hence, the subgradient inequality for convex functions leads for $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$ to the lower estimate

$$\mathcal{J}(\mathbf{x}_1) \geq \mathcal{J}(\mathbf{x}_2) + \boldsymbol{\nabla}\mathcal{J}(\mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2) - \frac{\lambda}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2. \tag{4.3}$$

Algorithm 4.1. Safeguarded AGD for λ-weakly convex $\mathcal{J}$ with L-Lipschitz gradient.

Input: initialization $\mathbf{x}_0 \in \mathbb{R}^d$, tolerance $\epsilon > 0$, $a > 1$
Set $t_0 = t_1 = 1$, $k = 1$, $\mathbf{z}_0 = \mathbf{x}_0$, $\mathbf{x}_1 = \mathbf{x}_0$
while $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|/\|\mathbf{x}_{k-1}\| > \epsilon$ or $k = 1$ **do**
  $\mathbf{z}_k = \mathbf{x}_k + \frac{t_{k-1}-1}{t_k}(\mathbf{x}_k - \mathbf{x}_{k-1})$
  $\text{crit} = \boldsymbol{\nabla}\mathcal{J}(\mathbf{z}_k)^T(\mathbf{z}_k - \mathbf{z}_{k-1}) + \frac{a\lambda}{2}\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2$
  if crit > 0 **then**
    $\mathbf{z}_k = \mathbf{x}_k$
    $t_k = 1$
  $\mathbf{x}_{k+1} = \mathbf{z}_k - \frac{1}{L}\boldsymbol{\nabla}\mathcal{J}(\mathbf{z}_k)$
  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
  $k \leftarrow k + 1$
Output: Approximate solution $\mathbf{x}_k$

Given some initialization $\mathbf{x}_0 = \mathbf{x}_{-1} = \mathbf{z}_0 \in \mathbb{R}^d$ and a sequence of Nesterov momentum parameters $\{\beta_k\}_{k\in\mathbb{N}} \subset [0,1]$, the standard AGD [41] update steps read

$$\mathbf{z}_k = \mathbf{x}_k + \beta_k(\mathbf{x}_k - \mathbf{x}_{k-1}), \tag{4.4}$$

$$\mathbf{x}_{k+1} = \mathbf{z}_k - \frac{1}{L}\boldsymbol{\nabla}\mathcal{J}(\mathbf{z}_k). \tag{4.5}$$

The combination of (4.5) and (4.2) yields the decrease estimate

$$\mathcal{J}(\mathbf{x}_{k+1}) - \mathcal{J}(\mathbf{z}_k) \leq -\frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{z}_k\|^2. \tag{4.6}$$

However, the update (4.4) does not necessarily guarantee the decrease of $\{\mathcal{J}(\mathbf{z}_k)\}_{k\in\mathbb{N}}$. Hence, for a predefined $a > 1$, we propose to check the condition

$$\boldsymbol{\nabla}\mathcal{J}(\mathbf{z}_k)^T(\mathbf{z}_k - \mathbf{z}_{k-1}) + \frac{a\lambda}{2}\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2 \leq 0 \tag{4.7}$$

after having tentatively performed (4.4). If (4.7) is violated, we perform the plain gradient update $\mathbf{z}_k = \mathbf{x}_k$ (instead of (4.4)) and apply a restart technique, as proposed in [46]. In this case, we get from (4.6) at the previous step that

$$\mathcal{J}(\mathbf{z}_k) - \mathcal{J}(\mathbf{z}_{k-1}) \leq -\frac{L}{2}\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2. \tag{4.8}$$

Otherwise, the incorporation of (4.3) implies that

$$\mathcal{J}(\mathbf{z}_k) - \mathcal{J}(\mathbf{z}_{k-1}) \leq \boldsymbol{\nabla}\mathcal{J}(\mathbf{z}_k)^T(\mathbf{z}_k - \mathbf{z}_{k-1}) + \frac{\lambda}{2}\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2 \leq -\frac{(a-1)\lambda}{2}\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2. \tag{4.9}$$

To sum up, the acceleration steps are kept only if they lead to a sufficient decrease of the objective. Otherwise, the plain gradient-descent step guarantees this decrease.

Remark 4.1. The gradient-based condition (4.7) is more restrictive than the objective-based condition (4.8). However, (4.7) is computationally cheaper to verify as it only involves

inner products of already computed quantities. In practice, we observed that (4.7) is rarely violated.

Remark 4.2. The parameter a in (4.7) must be greater than the weak-convexity modulus of $R_{\boldsymbol{\theta}}$ to ensure convergence. At this point, a precise estimate of this modulus—which we know to be bounded by one in our setting—weakens the condition (4.7) and typically yields faster convergence. On the contrary, the reliance on a loose bound leads to frequent restarts, at the detriment of acceleration.

Regarding Algorithm 4.1, we now derive a convergence result in Theorem 4.3 using [45, Theorem 3.7], which itself extends the seminal work [3] to the inertia setting. Note that the objective (4.1) is semialgebraic since the profile function ψ is piecewise polynomial. Hence, it satisfies the required (quite technical) KL property; see also [3].

Theorem 4.3. *Assume that $\mathcal{J}$ satisfies the KL property and is bounded from below. If the sequence $(\mathbf{z}_k)_{k\in\mathbb{N}}$ generated by Algorithm 4.1 (without the stopping criterion) is bounded, then it converges to a critical point $\hat{\mathbf{z}}$ of $\mathcal{J}$. Moreover, the sequence $(\mathbf{z}_k)_{k\in\mathbb{N}}$ has finite length, in the sense that*

$$\sum_k \|\mathbf{z}_{k+1} - \mathbf{z}_k\| < \infty. \tag{4.10}$$

Proof. According to [45, Theorem 3.7], we need to check that

(H1) there exists $a > 0$ such that $\mathcal{J}(\mathbf{z}_k) + a\|\mathbf{z}_k - \mathbf{z}_{k-1}\|^2 \leq \mathcal{J}(\mathbf{z}_{k-1})$ for all $k \in \mathbb{N}$;
(H2) there exists $b > 0$ such that $\|\nabla\mathcal{J}(\mathbf{z}_k)\| \leq 2b(\|\mathbf{z}_k - \mathbf{z}_{k-1}\| + \|\mathbf{z}_{k+1} - \mathbf{z}_k\|)$ for all $k \in \mathbb{N}$;
(H3) there exists a subsequence $(\mathbf{z}_{k_j})_{j\in\mathbb{N}}$ such that $\mathbf{z}_{k_j} \to \mathbf{z}$ and $\mathcal{J}(\mathbf{z}_{k_j}) \to \mathcal{J}(\mathbf{z})$.

These three conditions are needed in order to conclude that the iterates $(\mathbf{z}_k)_{k\in\mathbb{N}}$ satisfy (4.10) and converge to a critical point $\hat{\mathbf{z}}$ of $\mathcal{J}$. We have already verified (H1) in (4.8) and (4.9). For (H2), we first note that if (4.7) is violated, then it directly holds that

$$\|\nabla\mathcal{J}(\mathbf{z}_k)\| = L\|\mathbf{z}_{k+1} - \mathbf{z}_k\|. \tag{4.11}$$

Otherwise, $\|\nabla\mathcal{J}(\mathbf{z}_k)\| = L\|\mathbf{x}_{k+1} - \mathbf{z}_k\|$, and it follows that

$$\begin{aligned}\|\nabla\mathcal{J}(\mathbf{z}_k)\| &\leq L\|\mathbf{z}_{k+1} - \mathbf{z}_k\| + \beta_k L\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \\ &\leq L\|\mathbf{z}_{k+1} - \mathbf{z}_k\| + L\left\|\mathbf{z}_k - \frac{1}{L}\nabla\mathcal{J}(\mathbf{z}_k) - \mathbf{z}_{k-1} + \frac{1}{L}\nabla\mathcal{J}(\mathbf{z}_{k-1})\right\| \\ &\leq L\|\mathbf{z}_{k+1} - \mathbf{z}_k\| + 2L\|\mathbf{z}_k - \mathbf{z}_{k-1}\|. \end{aligned} \tag{4.12}$$

Since we assume that the sequence $(\mathbf{z}_k)_{k\in\mathbb{N}}$ is bounded and since $\mathcal{J}$ is continuous, also (H3) holds and the result follows from [45, Theorem 3.7]. ∎

Remark 4.4. To ensure that $(\mathbf{z}_k)_{k\in\mathbb{N}}$ remains bounded, one can simply add a regularization term $\kappa\|\mathbf{x}\|^2$ to $\mathcal{J}$, where $\kappa > 0$ can be arbitrarily small. Then $\mathcal{J}$ becomes coercive because the profile ψ has linear extensions (see Lemma 4.5). Therefore, $(\mathbf{z}_k)_{k\in\mathbb{N}}$ must remain bounded; otherwise $(\mathcal{J}(\mathbf{z}_k))_{k\in\mathbb{N}}$ could not be decreasing (see (4.8)). Empirically, however, this "trick" was found to be unnecessary as the iterates would remain bounded in all settings explored.

Lemma 4.5. *Let R be a ridge regularizer of the form (3.2), where the profiles ψ_j are continuous, even, and have affine extensions.*[8] *Then*

$$\mathbf{x} \mapsto \|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2 + R(\mathbf{x}) + \kappa\|\mathbf{x}\|_2^2 \tag{4.13}$$

is coercive for any $\kappa > 0$.

Proof. By assumption all ψ_j are affine on $[t_0, +\infty)$ with slope $u_j \in \mathbb{R}$. Hence, it holds for $|t| > t_0$ that $\psi_j(t) = \psi_j(|t|) = u_j(|t| - t_0) + \psi_j(t_0)$. Next, we define $v_j = \min_{|t| \le t_0}(\psi_j(t) - u_j(|t| - t_0)) \le \psi_j(t_0)$, which is well-defined since ψ_j are continuous. By definition of v_j, it holds for any $t \in \mathbb{R}$ that $\psi_j(t) \ge u_j(|t| - t_0) + v_j$, and the objective in (4.13) is lower bounded by

$$\mathbf{x} \mapsto \|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2 + \kappa\|\mathbf{x}\|_2^2 + \sum_{j=1}^{d \times N_C} u_j(|\mathbf{w}_j^T \mathbf{x}| - t_0) + v_j, \tag{4.14}$$

which is coercive for any $\kappa > 0$. ∎

Remark 4.6. For $\sqrt{\lambda_{\min}(\mathbf{H}^T\mathbf{H})} \ge \lambda$, the objective (4.1) is $(\sqrt{\lambda_{\min}(\mathbf{H}^T\mathbf{H})} - \lambda)$-strongly convex and, hence, convex. Then Algorithm 4.1 is guaranteed to converge to a global minimum of the objective. Otherwise, some results on convergence to local minima come into play, including with convergence rates [3, 44].

As the problem is potentially non-convex, the initialization of the algorithm may influence the final reconstruction. However, we did not observe such a dependence in our experimental settings. Therefore, we opted for *the common zero-initialization*. A more sophisticated strategy may take as initial configuration the reconstruction of a trustworthy convex variational model such as that in [19]. Then Algorithm 4.1 would be used to refine the reconstruction.

4.2. Experiments. The WCRR-NN model trained in section 3 to perform denoising on the BSD500 dataset is now deployed to solve two image-reconstruction problems using safeguarded AGD. For each setup, λ and σ are tuned over a validation set to maximize the peak signal-to-noise ratio (PSNR) with the coarse-to-fine routine from [19], and then used for evaluation.

MRI. The ground truth consists of fully sampled knee images with size (320×320) from the fastMRI dataset [26]. The corresponding MRI measurements are a subsampled version of the 2D Fourier transforms (k-space). This subsampling is performed with a Cartesian mask that has two parameters: the acceleration $M_{\text{acc}} = 4$ and the center fraction $M_{\text{cf}} = 0.08$. All the $\lfloor 320 M_{\text{cf}} \rfloor$ columns in the center of the k-space (low frequencies) are retained in full, while columns in the other region of the k-space are uniformly sampled. More precisely, we are left with $\lfloor 320/M_{\text{acc}} \rfloor$ selected columns. Lastly, both the real and imaginary parts of the measurements are corrupted by Gaussian noise with standard deviation $\sigma_{\mathbf{n}} = 10^{-4}$. For validation and testing, we picked 10 and 99 images, respectively, all normalized to have a maximum value of one.

CT. To provide a comparison with adversarial regularization (AR) [35] and its convex counterpart ACR [38] (see more details in section 4.2.1), we include the sparse-view CT experiment proposed in [38]. Its data consist of human abdominal CT scans for 10 patients,

[8] In the sense that there exists $t_0 \in \mathbb{R}$ such that ψ_j is affine on $(-\infty, -t_0]$ and on $[t_0, +\infty)$.

Table 4.1
PSNR and SSIM values for MRI and CT reconstruction experiments.

Metric	PSNR	SSIM
Zero-fill	27.92	0.711
TV [5]	32.03	0.7922
CRR-NN [19]	33.14	0.842
WCRR-NN	34.55	0.858
Prox-DRUNet [24]	35.09	0.864

(a) MRI

Metric	PSNR	SSIM	Param.
TV	31.57	0.852	1
ACR [38]	32.17	0.868	$6 \cdot 10^5$
CRR-NN	32.87	0.862	$5 \cdot 10^3$
AR [35]	33.62	0.875	$2 \cdot 10^7$
WCRR-NN	34.06	0.895	$1 \cdot 10^4$
Prox-DRUNet	34.20	0.901	$2 \cdot 10^7$

(b) CT

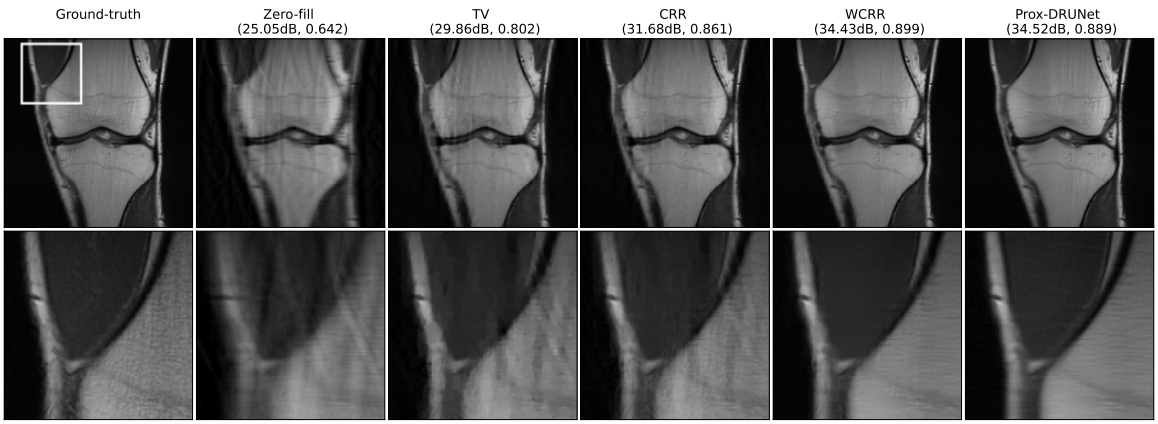

Figure 4.1. *Reconstructed images for the MRI experiment. The reported metrics are PSNR and SSIM.*

publicly available as part of the low-dose CT Grand Challenge [37]. For validation, 6 images are taken uniformly from the first patient of the training set used by [38]. To benchmark all methods, we use the same set as [38], made of 128 slices with size (512×512) from a single patient, all normalized to have a maximum value of one. The CT measurements are simulated using a parallel-beam acquisition geometry with 200 angles and 400 detectors. These measurements are corrupted by Gaussian noise with standard deviation $\sigma_{\mathbf{n}} = 2.0$.

4.2.1. Comparison and discussion. The PSNR and structural similarity index measure (SSIM) values on the test sets are reported together with the parameter numbers in Table 4.1. The hyperparameters of each method are tuned to maximize the average PSNR over the validation sets with the coarse-to-fine method described in [19]. We observe that WCRR-NNs outperform the other energy-based methods and are close to the PnP approach. For both problems, reconstructions are provided in Figures 4.1 and 4.2, and examples of convergence curves for SAGD are given in Figures 4.3 and 4.4.

Overall, the results illustrate the universality and efficiency of our method. In the following, we briefly comment on the competing methods used in our evaluation.

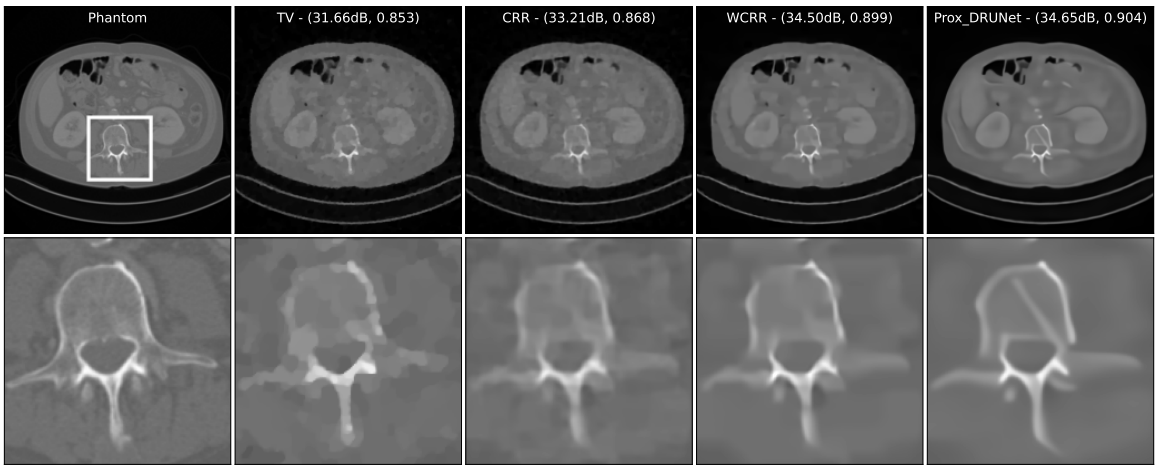

Figure 4.2. *Reconstructions for the sparse-view CT experiment. The reported metrics are PSNR and SSIM.*

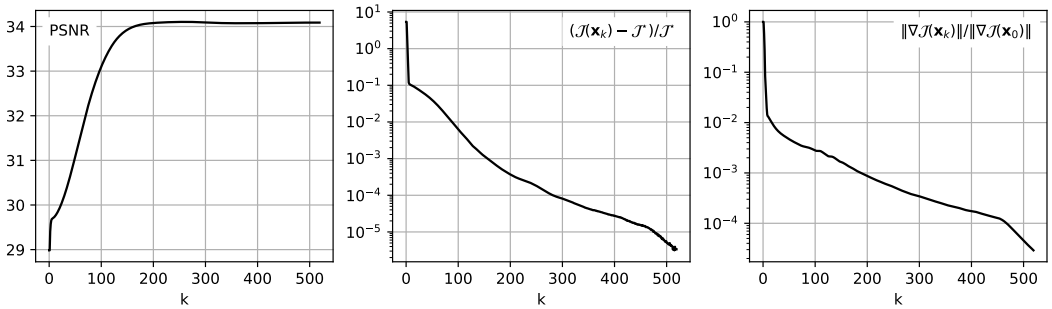

Figure 4.3. *Example of convergence curves (MRI).*

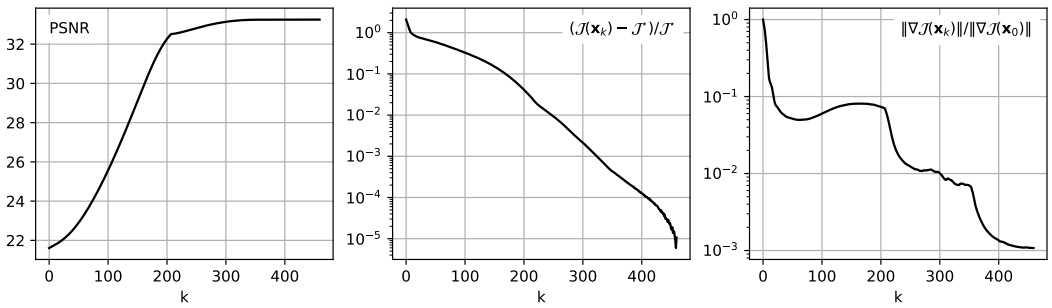

Figure 4.4. *Example of convergence curves (CT).*

Convex models. The TV and CRR-NN reconstructions serve as references for convex methods. They are computed via the FISTA algorithm [5] with a nonnegativity constraint. Similar to denoising, we observe that the move from convex to weakly convex regularization leads to significant improvements in quality. In the MRI experiment, the aliasing artifacts introduced by CRR-NN, and even more by TV, are suppressed by the weakly convex regularizer. In the

CT experiment, the TV reconstruction includes staircasing artifacts, and CRR-NN has a slight tendency to blur the edges. On the contrary, our weakly convex regularizer is able to produce sharp edges without blur, but sometimes at the cost of oversmoothing some background details. Note that convex models are still better understood from a theoretical perspective because convergence to global optima can be guaranteed. Hence, they might be favorable in certain settings.

Adversarial regularization. As references for explicit regularization approaches, we provide a comparison with the convex ACR [38, 40] framework and its non-convex counterpart AR [35]. Bypassing a gradient-based parameterization, these models parameterize the regularizer R directly and train it in an adversarial manner. As the regularizers of [38, 40, 35] are tailored to a specific inverse problem, we can only provide a comparison for their CT experiment. Even though ACR and AR have significantly more parameters than (W)CRR-NN, they perform less well. The numerical results present favorable evidence regarding the effectiveness of the parameterization used for WCRR-NNs. Note, however, that drawing a definitive conclusion on the parameterization only is delicate since AR and ACR rely on a different training procedure.

Plug-and-play. Our approach bears some resemblance with PnP methods since R is learned on a generic denoising task. Hence, it is natural to compare WCRR-NNs with a deep CNN version of this approach. Among countless variations, the recently proposed framework [24], which we refer to as Prox-DRUNet, is the closest to ours in terms of theoretical guarantees and existence of an underlying regularizer (see section 3.4 for a discussion). We use the pretrained DRUNet-based proximal denoiser from [24] within the PnP-PGD (proximal gradient descent) for CT and the PnP-DRS (Douglas–Rachford splitting) for MRI.[9] This approach, which may be considered the state of the art in energy-related PnP, yields slightly better PSNR and SSIM than our method. Note that Prox-DRUNet involves 3 orders of magnitude more parameters and days of training.

In the MRI experiment, both Prox-DRUNet and WCRR-NN are able to avoid the aliasing artifacts typically generated by the methods that rely on a convex regularizer. In the CT experiment, a visual inspection of the reconstructions reveals that quality metrics are only part of the story. While the output of Prox-DRUNet always looked remarkably realistic, it was more prone to hallucination/artifact exaggeration, especially for hard problems such as the CT experiment. In that respect, the Prox-DRUNet reconstruction in Figure 4.2 is particularly telling: It includes an elongated structure that is not present in the ground truth, nor in any other reconstruction. While such *enhanced* images are desirable in many settings and lead to state-of-the-art denoising performance, they raise major concerns for sensitive applications, including medical imaging. Regarding the theoretical convergence guarantees of PnP-PGD, the necessary Lipschitz constraint is only enforced by regularization during training. Unfortunately, it is infeasible to verify if it is met after training. In practice, it indeed seems to be not fully met [24].

5. Conclusion.

In this paper, we proposed a method for the learning of a 1-weakly convex regularizer that leads to a convex denoising functional. To the best of our knowledge, this is the first instance of convex non-convex schemes that surpasses BM3D for the denoising of

[9] PnP-DRS is well suited to settings where the proximal operator of the data term can be efficiently computed, which includes MRI but not CT.

natural images. A key feature of our method is that the architecture deployed to parameterize the regularizer is shallow. Thereby, the role of each parameter is transparent: Parameters are adjusted to produce a sparsity-promoting prior. Although the regularization of inverse problems with the learned regularizer does not necessarily lead to a convex objective, gradient-based optimization methods are empirically effective and produce high-quality reconstructions. In the future, a better understanding of WCRR-NNs might help to boost the performance of lightweight and robust data-driven image-reconstruction models even further. This includes the dependence of the learned regularizer on the modality and/or on the image domain used during training. It is indeed expected, for instance, that a fine-tuning of the regularizer with modality-specific prior knowledge will improve the quality of the reconstruction.

Acknowledgments. The authors are thankful to Pakshal Bohra and Stanislas Ducotterd for helpful discussions. Finally, the authors want to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] J. Abe, M. Yamagishi, and I. Yamada, *Linearly involved generalized Moreau enhanced models and their proximal splitting algorithm under overall convexity condition*, Inverse Problems, 36 (2020), 035012, https://doi.org/10.1088/1361-6420/ab551e.

[2] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, *Contour detection and hierarchical image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 898–916.

[3] H. Attouch, J. Bolte, and B. F. Svaiter, *Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods*, Math. Program., 137 (2013), pp. 91–129, https://doi.org/10.1007/s10107-011-0484-9.

[4] S. Bai, J. Z. Kolter, and V. Koltun, *Deep equilibrium models*, in Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, 2019, pp. 1–12.

[5] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[6] T. Blu and F. Luisier, *The SURE-LET approach to image denoising*, IEEE Trans. Image Process., 16 (2007), pp. 2778–2786, https://doi.org/10.1109/TIP.2007.906002.

[7] P. Bohra, J. Campos, H. Gupta, S. Aziznejad, and M. Unser, *Learning activation functions in deep (spline) neural networks*, IEEE Open J. Signal Process., 1 (2020), pp. 295–309, https://doi.org/10.1109/OJSP.2020.3039379.

[8] P. Bohra, D. Perdios, A. Goujon, S. Emery, and M. Unser, *Learning Lipschitz-controlled activation functions in neural networks for Plug-and-Play image reconstruction methods*, in NeurIPS 2021 Workshop on Deep Learning and Inverse Problems, 2021, pp. 1–9.

[9] S. Chang, B. Yu, and M. Vetterli, *Adaptive wavelet thresholding for image denoising and compression*, IEEE Trans. Image Process., 9 (2000), pp. 1532–1546, https://doi.org/10.1109/83.862633.

[10] Y. Chen and T. Pock, *Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration*, IEEE Trans. Pattern Anal. Mach. Intell., 39 (2016), pp. 1256–1272.

[11] Y. Chen, R. Ranftl, and T. Pock, *Insights into analysis operator learning: From patch-based sparse models to higher order MRFs*, IEEE Trans. Image Process., 23 (2014), pp. 1060–1072, https://doi.org/10.1109/TIP.2014.2299065.

[12] R. Cohen, Y. Blau, D. Freedman, and E. Rivlin, *It has potential: Gradient-driven denoisers for convergent solutions to inverse problems*, in Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, 2021, pp. 18152–18164.

[13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising by sparse 3-D transform-domain collaborative filtering*, IEEE Trans. Image Process., 16 (2007), pp. 2080–2095, https://doi.org/10.1109/TIP.2007.901238.

[14] D. Donoho, *De-noising by soft-thresholding*, IEEE Trans. Inform. Theory, 41 (1995), pp. 613–627, https://doi.org/10.1109/18.382009.

[15] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.
[16] S. Ducotterd, A. Goujon, P. Bohra, D. Perdios, S. Neumayer, and M. Unser, *Improving Lipschitz-constrained Neural Networks by Learning Activation Functions*, preprint, https://arxiv.org/abs/2210.16222, 2022.
[17] S. Ghadimi and G. Lan, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Math. Program., 156 (2016), pp. 59–99, https://doi.org/10.1007/s10107-015-0871-8.
[18] A. Goujon, A. Etemadi, and M. Unser, *The Role of Depth, Width, and Activation Complexity in the Number of Linear Regions of Neural Networks*, https://arxiv.org/abs/2206.08615, 2022.
[19] A. Goujon, S. Neumayer, P. Bohra, S. Ducotterd, and M. Unser, *A neural-network-based convex regularizer for inverse problems*, IEEE Trans. Comput. Imaging, 9 (2023), pp. 781–795, https://doi.org/10.1109/TCI.2023.3306100.
[20] R. Gribonval and M. Nikolova, *A characterization of proximity operators*, J. Math. Imaging Vision, 62 (2020), pp. 773–789, https://doi.org/10.1007/s10851-020-00951-y.
[21] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl, *Parseval proximal neural networks*, J. Fourier Anal. Appl., 26 (2020), 59.
[22] S. Hurault, A. Chambolle, A. Leclaire, and N. Papadakis, *A relaxed proximal gradient descent algorithm for convergent plug-and-play with proximal denoiser*, in Scale Space and Variational Methods in Computer Vision, Springer, 2023, pp. 379–392.
[23] S. Hurault, A. Leclaire, and N. Papadakis, *Gradient step denoiser for convergent Plug-and-Play*, in International Conference on Learning Representations, 2022, pp. 1–30, https://openreview.net/forum?id=fPhKeld3Okz.
[24] S. Hurault, A. Leclaire, and N. Papadakis, *Proximal denoiser for convergent Plug-and-Play optimization with nonconvex regularization*, in 39th International Conference on Machine Learning, Proc. Mach. Learn. Res. 162, PMLR, 2022, pp. 9483–9505.
[25] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, in 3rd International Conference on Learning Representations, ICLR, 2015, http://arxiv.org/abs/1412.6980.
[26] F. Knoll, J. Zbontar, A. Sriram, M. J. Muckley, M. Bruno, A. Defazio, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, Z. Zhang, M. Drozdzalv, A. Romero, M. Rabbat, P. Vincent, J. Pinkerton, D. Wang, N. Yakubova, E. Owens, C. L. Zitnick, M. P. Recht, D. K. Sodickson, and Y. W. Lui, *fastMRI: A publicly available raw k-space and DICOM dataset of knee images for accelerated MR image reconstruction using machine learning*, Radiol. Artif. Intell., 2 (2020), e190007.
[27] E. Kobler, A. Effland, K. Kunisch, and T. Pock, *Total deep variation for linear inverse problems*, in Conference on Computer Vision and Pattern Recognition, IEEE, 2020, pp. 7546–7555.
[28] E. Kobler, T. Klatzer, K. Hammernik, and T. Pock, *Variational networks: Connecting variational methods and deep learning*, in Pattern Recognition, Springer, 2017, pp. 281–293.
[29] M. Lang, H. Guo, J. Odegard, C. Burrus, and R. Wells, *Noise reduction using an undecimated discrete wavelet transform*, IEEE Signal Process. Lett., 3 (1996), pp. 10–12, https://doi.org/10.1109/97.475823.
[30] A. Lanza, S. Morigi, I. W. Selesnick, and F. Sgallari, *Sparsity-inducing nonconvex nonseparable regularization for convex image processing*, SIAM J. Imaging Sci., 12 (2019), pp. 1099–1134, https://doi.org/10.1137/18M1199149.
[31] A. Lanza, S. Morigi, I. W. Selesnick, and F. Sgallari, *Convex Non-Convex Variational Models*, Springer, Cham, 2021, pp. 1–57, https://doi.org/10.1007/978-3-030-03009-4_61-1.
[32] J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, and B. Recht, *First-order methods almost always avoid strict saddle points*, Math. Program., 176 (2019), pp. 311–337, https://doi.org/10.1007/s10107-019-01374-3.
[33] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier, *NETT: Solving inverse problems with deep neural networks*, Inverse Problems, 36 (2020), 065005, https://doi.org/10.1088/1361-6420/ab6d57.
[34] J. Li, J. Li, Z. Xie, and J. Zou, *Plug-and-Play ADMM for MRI reconstruction with convex nonconvex sparse regularization*, IEEE Access, 9 (2021), pp. 148315–148324, https://doi.org/10.1109/ACCESS.2021.3124600.
[35] S. Lunz, O. Öktem, and C.-B. Schönlieb, *Adversarial regularizers in inverse problems*, in Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, 2018, pp. 8516–8525.

[36] M. T. McCann and M. Unser, *Biomedical image reconstruction: From the foundations to deep neural networks*, Found. Trends Signal Process., 13 (2019), pp. 283–359, https://doi.org/10.1561/2000000101.
[37] C. McCollough, *TU-FG-207A-04: Overview of the low dose CT grand challenge*, Med. Phys., 43 (2016), pp. 3759–3760, https://doi.org/10.1118/1.4957556.
[38] S. Mukherjee, S. Dittmer, Z. Shumaylov, S. Lunz, O. Öktem, and C.-B. Schönlieb, *Learned Convex Regularizers for Inverse Problems*, arXiv:2008.02839, 2021.
[39] S. Mukherjee, A. Hauptmann, O. Öktem, M. Pereyra, and C.-B. Schönlieb, *Learned reconstruction methods with convergence guarantees: A survey of concepts and applications*, IEEE Signal Process. Mag., 40 (2023), pp. 164–182, https://doi.org/10.1109/MSP.2022.3207451.
[40] S. Mukherjee, C.-B. Schönlieb, and M. Burger, *Learning convex regularizers satisfying the variational source condition for inverse problems*, in NeurIPS Workshop on Deep Learning and Inverse Problems, 2021, pp. 1–5.
[41] Y. E. Nesterov, *A method of solving a convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk, 269 (1983), pp. 543–547.
[42] S. Neumayer, A. Goujon, P. Bohra, and M. Unser, *Approximation of Lipschitz functions using deep spline neural networks*, SIAM J. Math. Data Sci., 5 (2023), pp. 306–322, https://doi.org/10.1137/22M1504573.
[43] M. Nikolova, *Energy minimization methods*, in Handbook of Mathematical Methods in Imaging, Springer, New York, 2015, pp. 157–204.
[44] P. Ochs, *Local convergence of the heavy-ball method and iPiano for non-convex optimization*, J. Optim. Theory Appl., 177 (2018), pp. 153–180, https://doi.org/10.1007/s10957-018-1272-y.
[45] P. Ochs, Y. Chen, T. Brox, and T. Pock, *iPiano: Inertial proximal algorithm for nonconvex optimization*, SIAM J. Imaging Sci., 7 (2014), pp. 1388–1419, https://doi.org/10.1137/130942954.
[46] B. O'Donoghue and E. Candès, *Adaptive restart for accelerated gradient schemes*, Found. Comput. Math., 15 (2015), pp. 715–732, https://doi.org/10.1007/s10208-013-9150-3.
[47] A. Parekh and I. W. Selesnick, *Convex denoising using non-convex tight frame regularization*, IEEE Signal Process. Lett., 22 (2015), pp. 1786–1790, https://doi.org/10.1109/LSP.2015.2432095.
[48] A. Ribes and F. Schmitt, *Linear inverse problems in imaging*, IEEE Signal Process. Mag., 25 (2008), pp. 84–99.
[49] S. Roth and M. J. Black, *Fields of experts*, Int. J. Comput. Vision, 82 (2009), pp. 205–229.
[50] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[51] G. Scrivanti, E. Chouzenoux, and J.-C. Pesquet, *A CNC approach for directional total variation*, in 30th European Signal Processing Conference, IEEE, 2022, pp. 488–492, https://doi.org/10.23919/EUSIPCO55093.2022.9909763.
[52] H. Sedghi, V. Gupta, and P. M. Long, *The singular values of convolutional layers*, in International Conference on Learning Representations, 2019, pp. 1–12, https://openreview.net/forum?id=rJevYoA9Fm.
[53] B. Tan, Y. Li, H. Zhao, X. Li, and S. Ding, *A novel dictionary learning method for sparse representation with nonconvex regularizations*, Neurocomput., 417 (2020), pp. 128–141, https://doi.org/10.1016/j.neucom.2020.07.085.
[54] A. N. Tikhonov, *Solution of incorrectly formulated problems and the regularization method*, Soviet Math., 4 (1963), pp. 1035–1038.
[55] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, *Plug-and-play image restoration with deep denoiser prior*, IEEE Trans. Pattern Anal. Mach. Intell., 44 (2022), pp. 6360–6376, https://doi.org/10.1109/TPAMI.2021.3088914.
[56] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, *Loss functions for image restoration with neural networks*, IEEE Trans. Comput. Imaging, 3 (2017), pp. 47–57, https://doi.org/10.1109/TCI.2016.2644865.
[57] J. Zou, M. Shen, Y. Zhang, H. Li, G. Liu, and S. Ding, *Total variation denoising with non-convex regularizers*, IEEE Access, 7 (2019), pp. 4422–4431, https://doi.org/10.1109/ACCESS.2018.2888944.