Computational Optimization and Applications (2023) 84:125–149
https://doi.org/10.1007/s10589-022-00392-w

Constrained and unconstrained deep image prior optimization models with automatic regularization

Pasquale Cascarano[1] · **Giorgia Franchini**[2] · **Erich Kobler**[3] · **Federica Porta**[2] · **Andrea Sebastiani**[1]

Received: 29 September 2021 / Accepted: 17 June 2022 / Published online: 27 July 2022

Abstract

Deep Image Prior (DIP) is currently among the most efficient unsupervised deep learning based methods for ill-posed inverse problems in imaging. This novel framework relies on the implicit regularization provided by representing images as the output of generative Convolutional Neural Network (CNN) architectures. So far, DIP has been shown to be an effective approach when combined with classical and novel regularizers. Unfortunately, to obtain appropriate solutions, all the models proposed up to now require an accurate estimate of the regularization parameter. To overcome this difficulty, we consider a locally adapted regularized unconstrained model whose local regularization parameters are automatically estimated for additively separable regularizers. Moreover, we propose a novel constrained formulation in analogy to Morozov's discrepancy principle which enables the application of a broader range of regularizers. Both the unconstrained and the constrained models are solved via the proximal gradient descent-ascent method. Numerical results demonstrate the robustness with respect to image content, noise levels and hyperparameters of the proposed models on both denoising and deblurring of simulated as well as real natural and medical images.

Keywords Deep image prior · Convolutional neural networks · Automatic regularization · Regularization by denoising · Gradient descent-ascent methods · Image denoising · Image deblurring

✉ Giorgia Franchini
giorgia.franchini@unimore.it

Federica Porta
federica.porta@unimore.it

[1] Department of Mathematics, University of Bologna, Bologna, Italy

[2] Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Modena, Italy

[3] Institute of Computer Graphics, University of Linz, Linz, Austria

Mathematics Subject Classification 65K10 · 68U10 · 90C06

1 Introduction

The task of image restoration aims at recovering a clean and sharp unknown image $\boldsymbol{u} \in \mathbb{R}^n$ given a blurry and/or noisy measurement $\boldsymbol{g} \in \mathbb{R}^m$.

Mathematically, the restoration process can be modelled as a linear inverse problem:

$$\text{find} \quad \boldsymbol{u} \in \mathbb{R}^n \quad s.t. \quad \boldsymbol{H}\boldsymbol{u} + \boldsymbol{\eta} = \boldsymbol{g}, \tag{1}$$

where $\boldsymbol{H} \in \mathbb{R}^{m \times n}$ is a known forward operator and $\boldsymbol{\eta} \in \mathbb{R}^m$ is the noise corrupting the data. In this work, we consider a zero-mean Additive White Gaussian Noise (AWGN) component with standard deviation $\sigma_{\boldsymbol{\eta}}$.

Linear inverse problems are well-known to be ill-posed [3], therefore finding $\boldsymbol{u}$ from (1) by simply inverting $\boldsymbol{H}$ is useless due to the lack of stability and/or uniqueness properties. The task is usually reformulated as the problem of finding an estimate $\boldsymbol{u}^*$ of the desired $\boldsymbol{u}$ as accurate as possible via a well-posed problem. In the last decades, several approaches have been proposed, ranging from classical variational regularization methods to deep learning based approaches [20, 23, 30, 37].

Variational regularization methods compute $\boldsymbol{u}^*$ as the solution of the following regularized optimization problem:

$$\boldsymbol{u}^* \in \operatorname*{argmin}_{\boldsymbol{u} \in \mathbb{R}^n} \frac{1}{2} \|\boldsymbol{H}\boldsymbol{u} - \boldsymbol{g}\|_2^2 + \lambda R(\boldsymbol{u}), \tag{2}$$

where the first and the second terms are referred to as *data fidelity* and *regularization*, respectively. The hyperparameter λ is a positive scalar typically called *regularization parameter*. More generally, the data fidelity term measures how a given image adheres to the model (1). Its definition usually depends on the type of noise affecting the acquired $\boldsymbol{g}$ and, upon AWGN assumptions, it is frequently defined as an ℓ_2-norm functional. The regularization term $R : \mathbb{R}^n \to \mathbb{R}$ reflects prior information on the desired solution, such as its regularity and/or sparsity [21], whereas the hyparameter λ weights the strength of the regularization.

Very recently, supervised deep learning based methods have shown state-of-the-art performances in the field of imaging inverse problems [32] due to their capability to learn the correlation between degraded images and their cleaned counterparts by exploiting high representative models like Deep Neural Network architectures and an outer training set of degraded-cleaned example pairs. However, in general, these supervised approaches have several issues, including the lack of generalization when not trained with enough data. Moreover, in many real applications, such as medical imaging, it is practically impossible to build a labeled dataset with both ground truth and degraded data [41].

All these reasons have motivated researchers to inspect unsupervised deep learning approaches which avoid the usage of the training sets [12, 13, 25, 26, 38]. Deep Image Prior (DIP) [38] is among the most promising methods belonging to this class. The DIP framework leverages the fact that the architecture of a deep Convolutional Neural Network (CNN) generator reproduces natural images more easily than random noise, thus inducing implicit regularization. Given a CNN generator $f : \mathbb{R}^s \times \mathbb{R}^N \to \mathbb{R}^n$ whose weights are denoted by $\boldsymbol{\theta} \in \mathbb{R}^s$ and a random input vector $z \in \mathbb{R}^N$ sampled from a uniform distribution, the DIP approach [38] looks for a set of weights $\boldsymbol{\theta}^*$, combining the following minimization problem

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^s}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{H}f(\boldsymbol{\theta},z)-\boldsymbol{g}\|_2^2 \tag{3}$$

with an early stopping procedure. More specifically, the weights $\boldsymbol{\theta}^*$ are obtained by applying standard gradient-based iterative algorithms to the problem (3) and early stopping the iterative process before overfitting the degraded image $\boldsymbol{g}$. The restored image $\boldsymbol{u}^*$ is then computed as $f(\boldsymbol{\theta}^*, z)$.

Up to now, researchers have mostly worked on a theoretical analysis of DIP [1, 10, 11] as well as on boosting its performance. Inspired by standard variational regularization methods, in [2, 7, 8, 29, 31, 39] the authors improved the DIP performance by adding an explicit penalization term R to the objective in (3). Hence, the optimization problem (3) is replaced by the following regularized one:

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^s}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{H}f(\boldsymbol{\theta},z)-\boldsymbol{g}\|_2^2 + \lambda R(f(\boldsymbol{\theta},z)). \tag{4}$$

As an example, in [2, 29, 39] R is set as the standard Total Variation (TV) [35], whereas in [31] the authors consider the RED regularizer [34]. In more details, the definition of TV comes from the assumption that natural images often admit very sparse approximations in the gradient domain. Hence, given a vectorized image $\boldsymbol{u} \in \mathbb{R}^n$, the TV regularizer is defined as follows:

$$\mathrm{TV}(\boldsymbol{u}) := \|\boldsymbol{D}\boldsymbol{u}\|_{1,2} := \sum_{i=1}^{n} \left(|(\mathbf{D_h}\boldsymbol{u})_i|^2 + |(\mathbf{D_v}\boldsymbol{u})_i|^2\right)^{1/2}, \tag{5}$$

where by $\boldsymbol{D} = (\boldsymbol{D}_h;\boldsymbol{D}_v) \in \mathbb{R}^{2n\times n}$ we denote the discrete gradient such that $\boldsymbol{D}_h \in \mathbb{R}^n$, $\boldsymbol{D}_v \in \mathbb{R}^n$ are the first order finite difference discrete operators along the horizontal and vertical axes, respectively. On the other hand, the RED regularizer [34] is based on the so called regularization by denoising principle, i.e. the capability of denoisers to induce regularization. It is defined as follows:

$$R(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T(\boldsymbol{u} - \mathsf{D}(\boldsymbol{u})), \tag{6}$$

where $\mathsf{D}(\cdot)$ is chosen as any off-the-shelf denoiser. In [34], by assuming the differentiability, local homogeneity, Jacobian symmetry and filter passivity of $\mathsf{D}(\cdot)$, the authors prove that R is convex, differentiable and, moreover, $\nabla R(\boldsymbol{u}) = \boldsymbol{u} - \mathsf{D}(\boldsymbol{u})$. Hereafter we denote by DeepRED the method proposed in [31] to solve problem (4) when R is set as the RED regularizer.

The selection of the regularization parameter λ in (4) is an essential issue that this approach inherits from the class of variational regularization methods [37, 43]. A wise choice of regularization parameter is obviously crucial for obtaining useful approximate solutions to ill-posed problems. Indeed, replacing (3) with (4) induces better regularized solutions, provided a suitable value for λ depending both on the level of degradation of the acquired image and on the considered problem. In the literature there exist various strategies for choosing the parameter λ, such as the Morozov's discrepancy principle, the generalized cross-validation (GCV) [14], the L-curve method [17], and the unbiased predictive risk estimator [28]. However, it is well-known that such strategies can present different limitations: they are not at all easy to apply for every regularizer; they can provide either over or under smoothed solutions; they may often require to solve (4) many times for different values of λ, making the overall procedure computationally expensive. For these reasons, manually tuning the regularization parameter by trial-and-error procedures is common in the regularized DIP framework [2, 7, 29, 31, 39], leading to an high demanding workload.

Contributions In this work, we provide two different DIP based optimization models which share the property of automatically balancing the effect of the regularization. First, we consider an unconstrained model as the one in (4) where the regularization term is additively separable. The strength of the regularization is pixelwise weighted by a set (one for each pixel) of local regularization parameters whose definition is based on local patterns. Following the idea of estimating the regularization parameter iteratively suggested in [16, 40], we automatically estimate the set of local regularization parameters according to the Uniform PENalty (UPEN) principle [5]. Furthermore, we propose to reformulate the standard regularized unconstrained DIP optimization problem (4) as a constrained one, whose constraints impose that the residual $\|\boldsymbol{H}f(\boldsymbol{\theta}^*, \boldsymbol{z}) - \boldsymbol{g}\|_2$ is almost equal to the standard deviation of the noise affecting the acquired data, in accordance to the discrepancy principle. As evident, this approach strictly depends on an estimation of the noise level in the corrupted image. However, in real applications, choosing a reasonable value of the noise level is usually much easier than finding a suitable value of the regularization parameter λ. Indeed, many efficient algorithms to estimate the noise level are known in the literature [19, 24] and successfully exploited in many fields [15, 36]. To consider automatically regularized DIP-based optimization models is an interesting issue in the DIP framework, since so far, to the best of our knowledge, no one working in this context has been focused on this aspect. Both the unconstrained and constrained models are solved via a modified and more efficient version of the proximal gradient descent-ascent (PGDA) method in which the computation of the gradient step is split in two blocks. Finally, we show that, upon suitable assumptions [6], some convergence results for the arising iterative schemes can be provided.

Organization of the paper In Sect. 2, we introduce both the unconstrained and the constrained models and we illustrate the resulting PGDA schemes. In Sect. 3, we present several numerical experiments on synthetic as well as real blurred and noisy natural and medical images and we compare the results with the standard DIP [38] and DeepRED [31].

2 Novel automatically regularized DIP-based optimization models

In this section, we introduce the unconstrained and the constrained optimization models to face the regularized DIP problem and we show how they can be treated within the PGDA framework.

2.1 Unconstrained model

The approaches described in [2, 29, 39] consider the unconstrained model (4) setting the regularizer as the handcrafted Total Variation with a single regularization parameter, which does not allow to adapt the regularization to the local image patterns. Conversely, we consider a flexible space variant regularizer and a set of local regularization parameters λ_i for $i = 1 \dots n$ weighting the strength of the regularization for each pixel. The resulting unconstrained model reads:

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^s}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{H}f(\boldsymbol{\theta},\boldsymbol{z}) - \boldsymbol{g}\|_2^2 + \sum_{i=1}^{n} \lambda_i R_i((\mathcal{A}f(\boldsymbol{\theta},\boldsymbol{z}))_{I_i}), \tag{7}$$

where $I_i \subset \{1,\dots,l\} = I$ such that $I_i \cap I_j = \emptyset$ for every $i,j = 1 \dots n$ with $i \neq j$ and $\bigcup_{i=1}^{n} I_i = I$, R_i are real-valued functions representing the local components of the regularizer, $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^l$ is a generic operator and l is a positive integer such that $l \geq n$. The functions R_i and the local parameters λ_i usually represent local energies defined on a neighbourhood of the i-th pixel thus forcing prior information based on local patterns. Practically, these local parameters are automatically chosen along the iterations as explained in Remark 1. Considering a vector $\boldsymbol{v}$ in $\mathbb{R}^l$, for every $i = 1, \dots n$ we denote $\boldsymbol{v}_{I_i} \in \mathbb{R}^{|I_i|}$ as the vector specified by the components of $\boldsymbol{v}$ whose indexes are in I_i. Examples of regularization terms belonging to this class are the Tikhonov-like and the Total Variation ones. For instance, in the Tikhonov-based regularizers, $\mathcal{A}$ is usually chosen as the identity or the laplacian operators, whereas $R_i : \mathbb{R} \to \mathbb{R}$ is chosen as the square function. Concerning the isotropic Total Variation, $\mathcal{A}$ represents the discrete gradient and $R_i : \mathbb{R}^2 \to \mathbb{R}$ is chosen as the ℓ_2-norm function.

By adding an auxiliary variable $\boldsymbol{v} := \mathcal{A}f(\boldsymbol{\theta},\boldsymbol{z})$, the optimization problem (7) is equivalent to the following formulation:

$$\begin{aligned} \underset{\boldsymbol{\theta}\in\mathbb{R}^s,\boldsymbol{v}\in\mathbb{R}^l}{\operatorname{argmin}} \ \&\frac{1}{2}\|\boldsymbol{H}f(\boldsymbol{\theta},\boldsymbol{z}) - \boldsymbol{g}\|_2^2 + \sum_{i=1}^{n} \lambda_i R_i(\boldsymbol{v}_{I_i}) \\ \text{s.t.} \quad \mathcal{A}f(\boldsymbol{\theta},\boldsymbol{z}) = \boldsymbol{v}. \end{aligned} \tag{8}$$

In order to solve problem (8), we introduce the corresponding augmented Lagrangian function defined as

$$
\begin{aligned}
L(\boldsymbol{\theta}, \boldsymbol{v}, \boldsymbol{\mu}_v) = \frac{1}{2}\|\boldsymbol{H}f(\boldsymbol{\theta}, z) - \boldsymbol{g}\|_2^2 + \sum_{i=1}^{n} \lambda_i R_i(\boldsymbol{v}_{I_i}) \\
+ \frac{\beta_v}{2}\|\mathcal{A}f(\boldsymbol{\theta}, z) - \boldsymbol{v}\|_2^2 + \langle \boldsymbol{\mu}_v, \mathcal{A}f(\boldsymbol{\theta}, z) - \boldsymbol{v}\rangle,
\end{aligned} \tag{9}
$$

where β_v is a positive scalar, called penalty parameter and $\boldsymbol{\mu}_v$ is the Lagrangian parameter associated with the constraint $\mathcal{A}f(\boldsymbol{\theta}, z) = \boldsymbol{v}$. Some papers [8, 31] address the minimization of the regularized DIP optimization problem (4) by seeking the saddle points of the related augmented Lagrangian function through the ADMM algorithm. However, an highly inexact version of ADMM is practically implemented since the updating step for the weights $\boldsymbol{\theta}$ is, in general, solved inexactly by applying only one iteration of a gradient-based method. For this reason, instead of ADMM, we take into account another class of methods tailored for minimax problems. In more detail, by denoting with $\boldsymbol{x} \equiv [\boldsymbol{\theta};\boldsymbol{v}]$, we handle the saddle point problem

$$
\min_{\boldsymbol{x}\in\mathbb{R}^{s+l}} \max_{\boldsymbol{\mu}_v\in\mathbb{R}^l} L(\boldsymbol{x}, \boldsymbol{\mu}_v) \tag{10}
$$

by means of the class of alternating proximal gradient descent-ascent (PGDA) methods [6, 9, 27] (see Appendix 1 for a survey of these algorithms). By introducing the notation $\mathcal{R}(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i R_i(\boldsymbol{v}_{I_i})$ and defining

$$
K(\boldsymbol{x}, \boldsymbol{\mu}_v) = \frac{1}{2}\|\boldsymbol{H}f(\boldsymbol{\theta}, z) - \boldsymbol{g}\|_2^2 + \frac{\beta_v}{2}\|\mathcal{A}f(\boldsymbol{\theta}, z) - \boldsymbol{v}\|_2^2 + \langle \boldsymbol{\mu}_v, \mathcal{A}f(\boldsymbol{\theta}, z) - \boldsymbol{v}\rangle,
$$

upon suitable initialization of the involved variables, the k-th iteration of the alternating PGDA iterative algorithm described in [6] to solve (10) reads as follows:

$$
\begin{cases}
\boldsymbol{x}^{k+1} = \mathrm{prox}_{\alpha_x \mathcal{R}}(\boldsymbol{x}^k - \alpha_{\boldsymbol{x}} \nabla_{\boldsymbol{x}} K(\boldsymbol{x}^k, \boldsymbol{\mu}_v^k)) \\
\boldsymbol{\mu}_v^{k+1} = \boldsymbol{\mu}_v^k + \alpha_{\mu_v} \nabla_{\mu_v} K(\boldsymbol{x}^{k+1}, \boldsymbol{\mu}_v^k)
\end{cases} \tag{11}
$$

where $\alpha_{\boldsymbol{x}}$ and α_{μ_v} are proper positive learning rates. By definition of proximal operator, the vector $\boldsymbol{x}^{k+1}$ in the first step of (11) can be written as in the following:

$$
\boldsymbol{x}^{k+1} = \underset{\boldsymbol{x}}{\mathrm{argmin}}\ \alpha_{\boldsymbol{x}} \mathcal{R}(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - (\boldsymbol{x}^k - \alpha_{\boldsymbol{x}} \nabla_{\boldsymbol{x}} K(\boldsymbol{x}^k, \boldsymbol{\mu}_v^k))\|_2^2.
$$

Hence, in view of the notation introduced above,

$$
\begin{aligned}
\boldsymbol{x}^{k+1} &= \underset{\boldsymbol{\theta}\in\mathbb{R}^s, \boldsymbol{v}\in\mathbb{R}^l}{\mathrm{argmin}}\ \alpha_{\boldsymbol{x}} \mathcal{R}(\boldsymbol{x}) + \frac{1}{2}\left\| \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{v} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\theta}^k - \alpha_{\boldsymbol{x}} \nabla_{\boldsymbol{\theta}} K(\boldsymbol{x}^k,\ \boldsymbol{\mu}_v^k) \\ \boldsymbol{v}^k - \alpha_{\boldsymbol{x}} \nabla_{\boldsymbol{v}} K(\boldsymbol{x}^k,\ \boldsymbol{\mu}_v^k) \end{bmatrix} \right\|_2^2 \\
&= \underset{\boldsymbol{\theta}\in\mathbb{R}^s, \boldsymbol{v}\in\mathbb{R}^l}{\mathrm{argmin}}\ \alpha_{\boldsymbol{x}} \mathcal{R}(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{\theta} - (\boldsymbol{\theta}^k - \alpha_{\boldsymbol{x}} \nabla_{\boldsymbol{\theta}} K(\boldsymbol{x}^k, \boldsymbol{\mu}_v^k))\|_2^2 \\
&\quad + \frac{1}{2}\|\boldsymbol{v} - (\boldsymbol{v}^k - \alpha_{\boldsymbol{x}} \nabla_{\boldsymbol{v}} K(\boldsymbol{x}^k, \boldsymbol{\mu}_v^k))\|_2^2.
\end{aligned} \tag{12}
$$

Due to the separability of the objective in (12) with respect to the variables $\boldsymbol{\theta}$ and $\boldsymbol{v}$ and by assuming convexity of $\mathcal{R}$ and $\alpha_{\boldsymbol{x}} = \frac{1}{\beta_v}$, iteration (11) can be rewritten as

$$\begin{cases} \boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \alpha_x \nabla_{\boldsymbol{\theta}} K(\boldsymbol{x}^k, \boldsymbol{\mu}_v^k) \\ \boldsymbol{v}^{k+1} = \underset{\boldsymbol{v}\in\mathbb{R}^l}{\operatorname{argmin}}\ \alpha_x \mathcal{R}(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{v} - (\boldsymbol{v}^k - \alpha_x \nabla_v K(\boldsymbol{x}^k, \boldsymbol{\mu}_v^k))\|_2^2 = \\ \quad = \underset{\boldsymbol{v}\in\mathbb{R}^l}{\operatorname{argmin}} \sum_{i=1}^n \lambda_i R_i(\boldsymbol{v}_{I_i}) + \frac{1}{2\alpha_x}\left\| \boldsymbol{v} - \left(\mathcal{A}f(\boldsymbol{\theta}^k, \boldsymbol{z}) + \frac{\boldsymbol{\mu}_v^k}{\beta_v} \right) \right\|_2^2 \\ \boldsymbol{\mu}_v^{k+1} = \boldsymbol{\mu}_v^k + \alpha_{\mu_v}(\mathcal{A}f(\boldsymbol{\theta}^{k+1}, \boldsymbol{z}) - \boldsymbol{v}^{k+1}). \end{cases} \tag{13}$$

Concerning the optimization problem in the second step of (13), if the proximal map of R_i can be easily computed for all $i = 1 \dots n$, then the problem can be efficiently solved in a closed form by applying the proximity operator of R_i to the n components of $\mathcal{A}f(\boldsymbol{\theta}^k, \boldsymbol{z}) + \frac{\boldsymbol{\mu}_v^k}{\beta_v}$. Such hypotheses on R_i are not so restrictive. For example both Tikhonov-like and isotropic Total Variation regularizers satisfy these assumptions since the R_i are set as the square or ℓ_2-norm functions.

Remark 1 In our implementation, we chose to vary the set of local regularization parameters λ_i along the iterations. In particular, their formulation is inspired by [5] and reads:

$$\lambda_i^k = \frac{1}{2n} \frac{\|\boldsymbol{H}f(\boldsymbol{\theta}^{k+1}, \boldsymbol{z}) - \boldsymbol{g}\|_2^2}{R_i\big((\mathcal{A}f(\boldsymbol{\theta}^{k+1}, \boldsymbol{z}))_{I_i}\big)}. \tag{14}$$

This entails that the smaller is the value of the local component function the greater is the regularization provided at pixel i. We remark that, in the experimentation, we set these parameters to a certain value when the denominator decreases below a fixed threshold.

2.1.1 Practical and theoretical details of algorithm (13)

We point out that, taking into account the separable nature of problem (12) and the optimization efficiency, in the practical implementation, we exploit the already computed value for $\boldsymbol{\theta}^{k+1}$ in the update of $\boldsymbol{v}^{k+1}$. In Sect. 3.6 we compare the behaviour of the standard alternating PGDA method (13) and that of the implemented version which employs $\boldsymbol{\theta}^{k+1}$ for the computation of $\boldsymbol{v}^{k+1}$ on one of the problem under analysis. The two versions of the alternating PGDA are comparable, even if the standard one needs higher memory requirements.

Remark 2 Under the hypotheses that the function $K(\boldsymbol{x}, \boldsymbol{y})$ is ρ-weakly convex and L-Lipschitz in the first component uniformly in the second one, and the regularizer $\mathcal{R}$ is proper, convex and lower semicontinuous and $L_{\mathcal{R}}$-Lipschitz continuous on its domain, and since $K(\boldsymbol{x}, \boldsymbol{y})$ is concave and has Lipschitz continuous gradient in the second component uniformly in the first one, then the convergence result [6, Theorem 3.7] can be invoked. Such theorem ensures that an ε-stationary point [6, Definition (6)] of (9) can be visited in a finite number of iterations depending on ε. Both the invoked theorem and definition are recalled in Appendix 1 (see Definition () and Theorem ()). We point out that the learning rates $\alpha_{\boldsymbol{x}}$ and $\alpha_{\boldsymbol{\mu}_v}$ are required to be bounded by proper constants which we do not know in practice. For this reason, we decide to fix $\alpha_{\boldsymbol{x}} = \frac{1}{\alpha_{\mu_v}} = \beta_v$. The choice of β_v is discussed in the following remark.

Remark 3 The value of the penalty parameter β_v is hand-tuned. However, in the experimental part (Sect. 3) we empirically show that the choice of this hyperparameter does not affect the performance of the method as much as the choice of the regularization parameter when dealing with model (4). In detail, we empirically demonstrate that the performance of the proposed model is not sensitive to this penalty parameter β_v for all considered test problems if chosen in a reasonable set. Moreover none of the considered value for β_v makes the algorithm divergent.

2.2 Constrained model

The starting point of this approach is again the regularized DIP optimization problem in (4). Differently from the unconstrained model described in Sect. 2.1, we here assume $R : \mathbb{R}^n \to \mathbb{R}$ is a generic regularizer. The constrained model we refer to, in the following, reads as:

$$\operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^s} R(f(\boldsymbol{\theta},\boldsymbol{z})) \quad \text{s.t.} \quad f(\boldsymbol{\theta},\boldsymbol{z}) \in D_{\sigma_\eta}, \tag{15}$$

where D_{σ_η} is defined as:

$$D_{\sigma_\eta} := \{f(\boldsymbol{\theta},\boldsymbol{z}) \in \mathbb{R}^n \mid \|\boldsymbol{H}f(\boldsymbol{\theta},\boldsymbol{z}) - \boldsymbol{g}\|_2^2 \le \tau\sigma_{\boldsymbol{\eta}}^2 m\}, \tag{16}$$

with τ being a positive scalar and $\sigma_{\boldsymbol{\eta}}$ being the standard deviation of the noise affecting $\boldsymbol{g}$. This constrained model (15) exploits the Morozov's discrepancy principle by simply extending [18, 33]. If $R\circ f$ is convex, this model is equivalent to (4) for a suitable $\lambda \ge 0$. In particular, by the KKT complementary condition, the discrepancy principle seeks a $\lambda > 0$, such that the minimizer of (15) lies on the boundary of D_{σ_η}. However, this hypothesis of convexity appears too restrictive for the DIP framework. Nevertheless, under milder assumptions, the KKT conditions ensure that a local optimum for (15) is a stationary point for (4) provided a particular $\lambda \ge 0$. Therefore, we focus on problem (15) since it allows us to avoid the dependence on the choice of the regularization parameter λ. Finally, we cannot theoretically guarantee that the solution of (15) satisfies the discrepancy principle (namely, lies on the boundary of D_{σ_η}), but in Sect. 3 we empirically verify that our approaches implicitly enforce it. We stress that our general approach (15), largely differs from model (4) proposed in the literature, since it overcomes the problem of tuning the regularization parameter provided the noise standard deviation $\sigma_{\boldsymbol{\eta}}$. In practice, it is sufficient to consider a good estimate of $\sigma_{\boldsymbol{\eta}}$ which can be computed by applying the efficient algorithms described in [19, 24].

In order to solve (15), we propose to consider the alternating PGDA method. By introducing two auxiliary variables $\boldsymbol{t} := f(\boldsymbol{\theta}, z)$ and $\boldsymbol{r} := \boldsymbol{H}f(\boldsymbol{\theta}, z) - \boldsymbol{g}$, two positive penalty parameters β_t and β_r, the augmented Lagrangian functional is defined as:

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{t}, \boldsymbol{r}; \boldsymbol{\mu}_t, \boldsymbol{\mu}_r) = R(\boldsymbol{t}) + i_{B_\delta}(\boldsymbol{r}) + \langle \boldsymbol{\mu}_t, f(\boldsymbol{\theta}, z) - \boldsymbol{t} \rangle + \frac{\beta_t}{2} \| \boldsymbol{t} - f(\boldsymbol{\theta}, z) \|^2 \\ + \langle \boldsymbol{\mu}_r, \boldsymbol{H}f(\boldsymbol{\theta}, z) - \boldsymbol{g} - \boldsymbol{r} \rangle + \frac{\beta_r}{2} \| \boldsymbol{r} - (\boldsymbol{H}f(\boldsymbol{\theta}, z) - \boldsymbol{g}) \|^2, \end{aligned} \tag{17}$$

where i_{B_δ} is the indicator function of the ball $B_\delta \subset \mathbb{R}^m$, centered in $\mathbf{0} \in \mathbb{R}^m$, of radius $\delta := \sqrt{\tau \sigma^2 m}$, and $\boldsymbol{\mu}_t$, $\boldsymbol{\mu}_r$ are the Lagrangian parameters related to the auxiliary variables. Given the notation $\boldsymbol{x} = [\boldsymbol{\theta}, \boldsymbol{t}, \boldsymbol{r}]$ and $\boldsymbol{y} = [\boldsymbol{\mu}_t, \boldsymbol{\mu}_r]$, and by setting

$$\begin{aligned} K(\boldsymbol{x}, \boldsymbol{y}) = \frac{\beta_t}{2} \| \boldsymbol{t} - f(\boldsymbol{\theta}, z) \|^2 + \frac{\beta_r}{2} \| \boldsymbol{r} - (\boldsymbol{H}f(\boldsymbol{\theta}, z) - \boldsymbol{g}) \|^2 \\ + \langle \boldsymbol{\mu}_t, f(\boldsymbol{\theta}, z) - \boldsymbol{t} \rangle + \langle \boldsymbol{\mu}_r, \boldsymbol{H}f(\boldsymbol{\theta}, z) - \boldsymbol{g} - \boldsymbol{r} \rangle \end{aligned}$$

and

$$\mathcal{R}(\boldsymbol{x}) = R(\boldsymbol{t}) + i_{B_\rho}(\boldsymbol{r}),$$

the augmented Lagrangian function (17) has the form:

$$L(\boldsymbol{\theta}, \boldsymbol{t}, \boldsymbol{r}; \boldsymbol{\mu}_t, \boldsymbol{\mu}_r) \equiv L(\boldsymbol{x}, \boldsymbol{y}) \equiv K(\boldsymbol{x}, \boldsymbol{y}) + \mathcal{R}(\boldsymbol{x}).$$

The general iteration of the alternating PGDA iterative algorithm to solve

$$\min_{\boldsymbol{x} \in \mathbb{R}^{s+n+m}} \max_{\boldsymbol{y} \in \mathbb{R}^{n+m}} L(\boldsymbol{x}, \boldsymbol{y}) \tag{18}$$

can be written as:

$$\begin{cases} \boldsymbol{x}^{k+1} = \text{prox}_{\alpha_x \mathcal{R}}(\boldsymbol{x}^k - \alpha_x \nabla_{\boldsymbol{x}} K(\boldsymbol{x}^k, \boldsymbol{y}^k)) \\ \boldsymbol{y}^{k+1} = \boldsymbol{y}^k + \alpha_y \nabla_{\boldsymbol{y}} K(\boldsymbol{x}^{k+1}, \boldsymbol{y}^k) \end{cases} \tag{19}$$

where α_x and α_y are proper positive learning rates. By following the approach employed in Sect. 2.1 for the unconstrained case, the vector $\boldsymbol{x}^{k+1}$ in the first step of (19) can be rewritten as

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \text{argmin}_{\boldsymbol{x}} \ \alpha_x \mathcal{R}(\boldsymbol{x}) + \frac{1}{2} \left\| \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{t} \\ \boldsymbol{r} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\theta}^k - \alpha_x \nabla_{\boldsymbol{\theta}} K(\boldsymbol{x}^k, \boldsymbol{y}^k) \\ \boldsymbol{t}^k - \alpha_x \nabla_{\boldsymbol{t}} K(\boldsymbol{x}^k, \boldsymbol{y}^k) \\ \boldsymbol{r}^k - \alpha_x \nabla_{\boldsymbol{r}} K(\boldsymbol{x}^k, \boldsymbol{y}^k) \end{bmatrix} \right\|_2^2 \\ &= \text{argmin}_{\boldsymbol{x}} \ \alpha_x R(\boldsymbol{t}) + i_{B_\rho}(\boldsymbol{r}) + \frac{1}{2} \| \boldsymbol{\theta} - (\boldsymbol{\theta}^k - \alpha_x \nabla_{\boldsymbol{\theta}} K(\boldsymbol{x}^k, \boldsymbol{y}^k)) \|_2^2 \\ &+ \frac{1}{2} \| \boldsymbol{t} - (\boldsymbol{t}^k - \alpha_x \nabla_{\boldsymbol{t}} K(\boldsymbol{x}^k, \boldsymbol{y}^k)) \|_2^2 \\ &+ \frac{1}{2} \| \boldsymbol{r} - (\boldsymbol{r}^k - \alpha_x \nabla_{\boldsymbol{r}} K(\boldsymbol{x}^k, \boldsymbol{y}^k)) \|_2^2. \end{aligned} \tag{20}$$

As a consequence, by selecting $\beta_t = \beta_r = \dfrac{1}{\alpha_x}$,

$$\begin{cases} \boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \alpha_x \nabla_{\boldsymbol{\theta}} K(\boldsymbol{x}^k, \boldsymbol{y}^k) \\ \boldsymbol{t}^{k+1} = \underset{\boldsymbol{t}\in\mathbb{R}^n}{\operatorname{argmin}}\ \alpha_x R(\boldsymbol{t}) + \frac{1}{2}\|\boldsymbol{t} - (\boldsymbol{t}^k - \alpha_x \nabla_{\boldsymbol{t}} K(\boldsymbol{x}^k, \boldsymbol{y}^k))\|_2^2 \\ \phantom{\boldsymbol{t}^{k+1}} = \underset{\boldsymbol{t}\in\mathbb{R}^n}{\operatorname{argmin}}\ \alpha_x R(\boldsymbol{t}) + \frac{1}{2}\left\|\boldsymbol{t} - \left(f(\boldsymbol{\theta}^k, z) + \frac{\boldsymbol{\mu}_t^k}{\beta_t}\right)\right\|_2^2 \\ \boldsymbol{r}^{k+1} = \underset{\boldsymbol{r}\in\mathbb{R}^m}{\operatorname{argmin}}\ \alpha_x i_{B_\rho}(\boldsymbol{r}) + \frac{1}{2}\|\boldsymbol{r} - (\boldsymbol{r}^k - \alpha_x \nabla_{\boldsymbol{r}} K(\boldsymbol{x}^k, \boldsymbol{y}^k))\|_2^2 \\ \phantom{\boldsymbol{r}^{k+1}} = \underset{\boldsymbol{r}\in\mathbb{R}^m}{\operatorname{argmin}}\ \alpha_x i_{B_\rho}(\boldsymbol{r}) + \frac{1}{2\alpha_x}\left\|\boldsymbol{r} - (\boldsymbol{H}f(\boldsymbol{\theta}^k, z) - \boldsymbol{g}) - \frac{\boldsymbol{\mu}_r^k}{\beta_r}\right\|_2^2 \\ \boldsymbol{\mu}_t^{k+1} = \boldsymbol{\mu}_t^k + \alpha_y(f(\boldsymbol{\theta}^{k+1}, z) - \boldsymbol{t}^{k+1}) \\ \boldsymbol{\mu}_r^{k+1} = \boldsymbol{\mu}_r^k + \alpha_y(\big(\boldsymbol{H}f(\boldsymbol{\theta}^{k+1}, z) - \boldsymbol{g}\big) - \boldsymbol{r}^{k+1}). \end{cases} \tag{21}$$

Similarly to the standard DIP framework solving (3), the first step in (21) updates the network's weights performing one back-propagation step. The update of $\boldsymbol{t}$, provided by the second step reported in (21), strictly depends on the choice of the regularizer. However, the minimization problem to find $\boldsymbol{t}^{k+1}$ is mathematically equivalent to the proximal map of $\alpha_x R$ in $f(\boldsymbol{\theta}^k, z) + \dfrac{\boldsymbol{\mu}_t^k}{\beta_t}$, therefore it can admit a closed form solution or it can be solved through either fixed point or gradient descent strategies as in [31]. The update of $\boldsymbol{r}$ is a simple projection of $\boldsymbol{H}f(\boldsymbol{\theta}^k, z) - \boldsymbol{g} + \frac{\boldsymbol{\mu}_r^k}{\beta_r}$ onto the ball B_δ. From the practical point of view, in the updating steps for $\boldsymbol{t}^{k+1}$ and $\boldsymbol{r}^{k+1}$ in (21), we exploit the already computed vector $\boldsymbol{\theta}^{k+1}$ for improving the optimization efficiency, as already discussed in Sect. 2.1.1. As for the convergence properties of the scheme (21), analogous conclusions to those of Remark 2 hold also in this case. Finally, we point out that the penalty parameters β_t and β_r are hand-tuned. However, the considerations highlighted for the penalty β_v in Remark 3 also apply to these two hyperparameters.

3 Results

In this section, we show the results of some numerical experiments carried out to highlight the main benefits of the suggested unconstrained and constrained models and to evaluate their effectiveness in solving image deblurring and denoising tasks on synthetic natural images as well as real medical ones. We perform several tests by varying the level of degradation, evaluate the performances through qualitative visual comparisons and quantitatively by PSNR and SSIM metrics. Finally, we discuss about the effectiveness of our implementation with respect to the standard PGDA.

3.1 Implementation and evaluation details

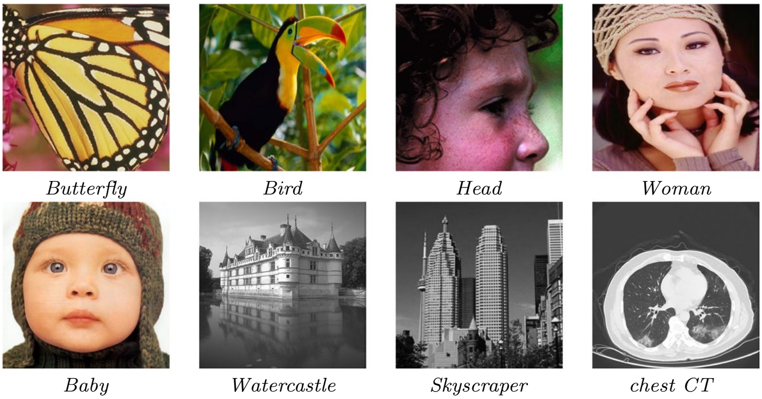
Implementation details Regarding the choice of the regularizer, both models allow a certain freedom of choice. For the unconstrained model, we consider the handcrafted space variant Total Variation and in the following we refer to it as DIP-WTV. We stress that in this case we consider the model (7) by assuming all R_i are set equal to the 2D ℓ_2-norm for $i = 1 \dots n$ and $\mathcal{A}$ is taken equal to the discrete gradient, hence $l = 2n$ and $I_i = \{i, n+i\}$ for each $i = 1, \dots, n$. Upon these assumptions the set of regularization parameters is defined according to the formula given by (14). Concerning the constrained optimization model (15), we set the regularizer R equal to the RED regularizer [34]. We refer to this approach as cDIP-RED in the following. The cDIP-RED approach requires the knowledge of standard deviation of the noise affecting the acquired image. As already mentioned, we point out that we estimate the noise standard deviation by applying the algorithm described in [19], even for the simulated tests where we do know it. The parameter τ in (15) is set equal to 1 for all the experiments. For both models and for all the experiments performed we stop the related PGDA iterative process after 5000 iterations. We remark that both the DIP-WTV and the cDIP-RED approaches are based on a modified version of the alternating PGDA scheme, as clarified is Sect. 2.1.1. In Sect. 3.6 we discuss this choice on one of the problem under analysis. As deep neural network architecture f we consider a generative CNN Encoder–Decoder architecture with skip connections by concatenation as suggested in [38], whereas the input z is taken as a random input tensor sampled from a uniform distribution. The input z is a 3D tensor having the same dimension of the unknown image and 32 channels, the number of weights $\boldsymbol{\theta}$ is about 2 millions.

In the experiments, we follow the common practice in the DIP framework [31, 38] and use the Adam [22] algorithm implemented in PyTorch with the default parameters to update $\boldsymbol{\theta}$ in (13) and (21). As typically done, we also perturb in each iteration the input z by a component sampled from a Gaussian distribution with zero mean and standard deviation equal to $\frac{1}{30}$ and we compute the final output as the average of all iterates.

Competitors We compare the proposed approaches DIP-WTV and cDIP-RED with the standard DIP [38] and the DeepRED [31] algorithms. We point out that in [31] the authors prove that DeepRED outperforms other several approaches as far as the deblurring and denoising tasks are concerned. Moreover, we underline that in their implementation[1], to enforce the regularization, the authors implement a strategy that increases the magnitude of the regularization parameter λ along the iterations when the computed solution starts overfitting the corrupted image. More precisely, when the PSNR value between the restored image and the degraded image is greater than a given threshold γ the regularization parameter is increased by adding a constant.

Test set In Fig. 1, we depict the images used in the numerical simulations. We consider a test set of five red-green-blue (RGB) natural images belonging to the

[1] https://github.com/GaryMataev/DeepRED



Butterfly *Bird* *Head* *Woman*

Baby *Watercastle* *Skyscraper* *chest CT*

Fig. 1 The test images employed in the numerical experiments. *Butterfly*: RGB image 256 × 256 pixels, *Bird*: RGB image 288 × 288 pixels, *Head*: RGB image 256 × 256 pixels, *Woman*: RGB 224 × 320 pixels, *Baby*: RGB 512 × 512 pixels, *Watercastle*: BW 320 × 480 pixels, *Skyscraper*: BW 256 × 256 pixels, *chest CT*: BW 512 × 512 pixels

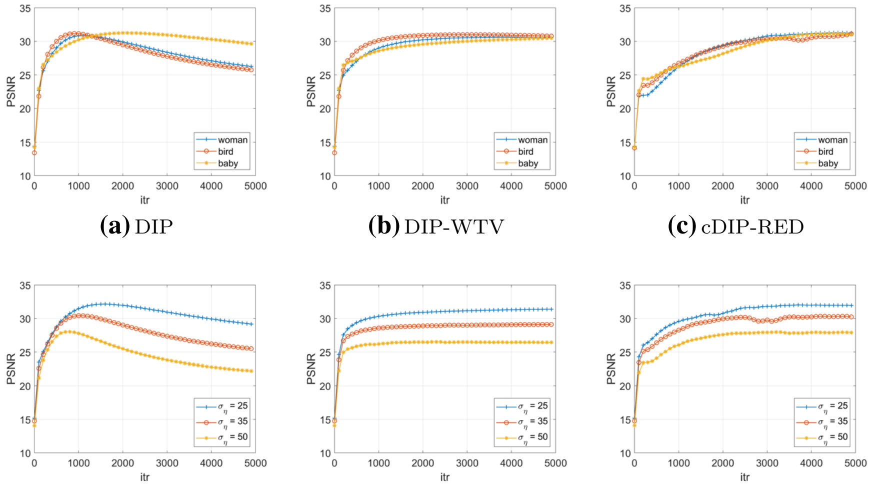
Set5 dataset [4], two black-white (BW) natural images and one chest CT image of a patient affected by COVID-19 already post-processed into a 2D image after the acquisition. We treat all the images belonging to Set5 as ground truths as well as the *watercastle* BW image. In our experiments, the simulated acquired images are created by applying the image formation model (1) to the related ground truths. In particular, to simulate blurred data we assume that $\boldsymbol{H}$ represents the discretization of a convolutional product with a Gaussian kernel of standard deviation $\sigma_{\boldsymbol{H}}$. We remark that the level of degradation of the simulated acquisition is specified by the magnitudes of σ_η and $\sigma_{\boldsymbol{H}}$. Finally, we stress that the *skyscraper* BW image and the real chest CT image are affected by artifacts. Since no ground truths are available for these images, the comparisons among the methods are carried out through visual inspection. The codes and the images used for these numerical experiments are available online[2].

3.2 Stability w. r. t. hyperparameters and empirical convergence

In this section, we describe the advantages which the models previously suggested bring over the considered competitors. In the first part, we empirically show the proposed approaches avoid the typical noise overfitting of DIP. Then, we underline how the suggested methods are more robust with respect to the choice of the hyperparameters than DeepRED. Finally, we empirically demonstrate that solutions of the proposed approaches satisfy the Morozov's discrepancy principle.

No overfitting In the first test, we highlight the sensitivity of the standard DIP algorithm with respect to the choice of the optimal number of iterations to be performed and we compare it with DIP-WTV and cDIP-RED. For all experiments in

[2] https://github.com/pcascarano/cDIP-RED-DIP-WTV



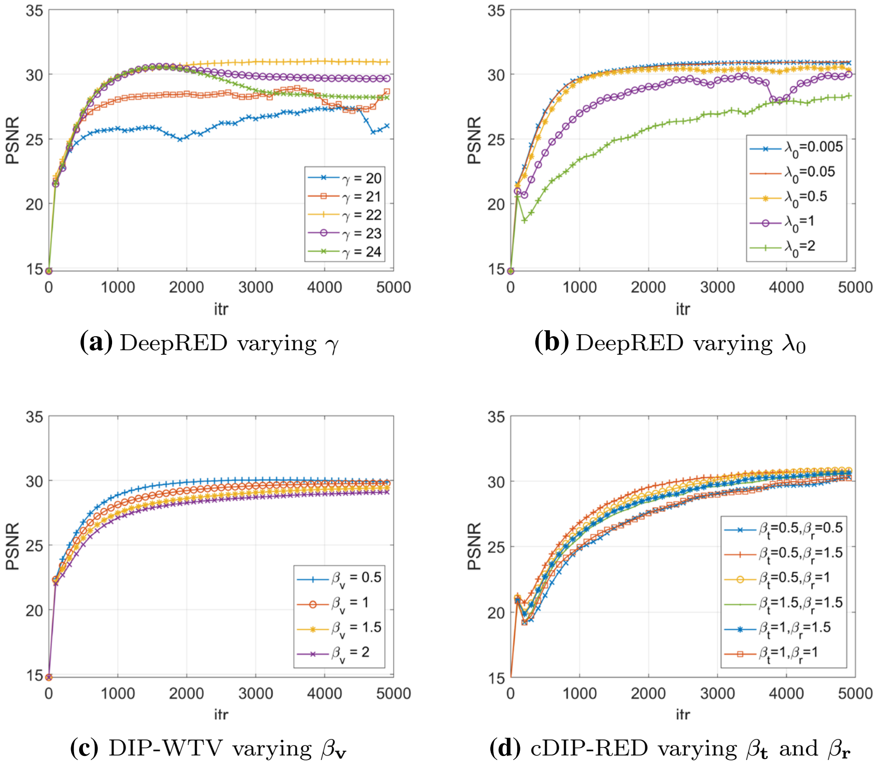
(a) DIP **(b)** DIP-WTV **(c)** cDIP-RED

(d) DIP **(e)** DIP-WTV **(f)** cDIP-RED

Fig. 2 The PSNR values achieved by DIP, DIP-WTV and cDIP-RED along the iterations. In (**a**)-(**b**)-(**c**) the DIP, DIP-WTV and cDIP-RED are tested on three different RGB images degraded setting $\sigma_\eta = 35$. In (**d**)-(**e**)-(**f**) DIP, DIP-WTV and cDIP-RED, respectively, are tested on the *butterfly* RGB image corrupted with different noise levels

this section we set the penalty parameters $\beta_v = 1$, $\beta_t = 0.5$ and $\beta_r = 1$. We consider the *woman*, *bird*, and *baby* images and we simulate the noisy acquisitions by corrupting the ground truths with an AWGN component of standard deviation $\sigma_\eta = 35$. Then, we apply DIP, DIP-WTV, and cDIP-RED and in the upper panel of Fig. 2 we depict the behaviour of the PSNR metric along the performed iterations. In order to analyze the relation between the noise level of the simulated acquisition and the optimal number of iterations to be performed by DIP, DIP-WTV and cDIP-RED, in lower panel of Fig. 2, we report the behavior of the PSNR metric along the iterations while the level of corruption changes. In particular, we consider the *butterfly* test images and we corrupt it by AWGN with $\sigma_\eta = 25, 35, 50$. As a general comment, Fig. 2 shows that standard DIP starts overfitting the corrupted image along the iterative process. Moreover, for the DIP approach, this test highlights that the number of iterations to reach the best PSNR strongly depends on the image considered (Fig. 2a), and on the level of corruption (Fig. 2d). Conversely, the DIP-WTV (Fig. 2b and e) and cDIP-RED (Fig. 2c and f) schemes do not overfit the corrupted data while the PSNR does not decrease.

No regularization parameter is required The DeepRED algorithm overcomes the problem of overfitting by adding the RED regularizer to the objective minimized by the standard DIP, provided a proper value for the regularization parameter λ. In this section, we highlight the sensitivity of the DeepRED algorithm with respect to the choice of the hyparameters defining the sequence of the regularization parameters, namely the threshold γ and the starting value of the regularization parameter λ_0. In Fig. 3a, we show the behaviour of the PSNR for different values of the threshold γ. The degraded image is obtained by corrupting



(a) DeepRED varying γ

(b) DeepRED varying λ_0

(c) DIP-WTV varying $\beta_{\mathbf{v}}$

(d) cDIP-RED varying $\beta_{\mathbf{t}}$ and $\beta_{\mathbf{r}}$

Fig. 3 The PSNR values achieved by DIP, DIP-WTV and cDIP-RED along the iterations for the *butterfly* RGB image with $\sigma_\eta = 35$

the *butterfly* image with an AWGN component setting $\sigma_\eta = 35$. The parameter λ_0 is fixed equal to 0.005 while the increasing factor equals 0.03. For high values of the threshold the regularization contribution is too weak thus the PSNR starts decreasing, which means DeepRED starts overfitting the degraded image. For low values of the threshold, the regularization parameter starts becoming high thus too much regularization is enforced. The best compromise for this *butterfly* test problem is $\gamma = 22$. However, in our experiments we observe that this value depends once again on both the image and the noise level considered and cannot be fixed a priori. In Fig. 3b, we report the PSNR behaviour obtained by fixing $\gamma = 22$ and by changing the starting value of the regularization parameter λ_0. We observe the output of the DeepRED in 5000 iterates largely depends on the choice of this hyperparameter. We stress that DeepRED is implemented in the ADMM framework which requires the tuning of the penalty parameter. In our experiments, the DeepRED penalty parameter is set equal to 0.5 as suggested by the authors in [31].

The main feature of DIP-WTV and cDIP-RED is that the introduced regularization has no parameters to be estimated. In the case of DIP-WTV, the space

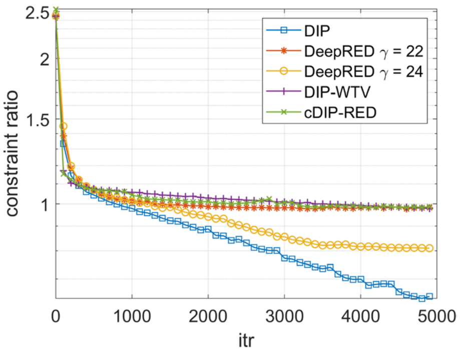


Fig. 4 The constraint ratio's trend along the iterations obtained by applying DIP, DeepRED (with $\gamma = 22$ and $\gamma = 24$), DIP-WTV and cDIP-RED to the *butterfly* RGB image corrupted by AWGN with $\sigma_\eta = 35$

variant regularization parameters are automatically estimated along the iterations, whereas the constrained formulation of cDIP-RED allows to automatically estimate the strength of the regularization by the Morozov's discrepancy principle.

Stability w.r.t. penalty parameters β_v, β_t and β_r We remark that for the DIP-WTV and cDIP-RED approaches we just need to fix the penalties β_v, β_t and β_r. However, in order to prove the stability of these methodologies with respect to the choice of these parameters, in Fig. 3c and d we depict the PSNR behaviour provided by DIP-WTV and cDIP-RED on the previous test image by setting different values for β_v, β_t and β_r. We stress that the range for the penalties for DIP-WTV and cDIP-RED has been deduced by the values suggested in [31] for their ADMM implementation.

From these figures we can conclude that for the DIP-WTV and the cDIP-RED methods the penalty parameters affect the convergence speed, but the PSNR behaviour of both the approaches is stable along the iterations and no noise-overfitting is present for any of the configurations considered. Moreover, we also observed that these different configurations provide comparable restorations in terms of visual quality in 5000 iterations. We observe that to maximize the performances of the cDIP-RED method we should set $\beta_t < \beta_r$. This is due to the fact that a bigger value of β_r provides more consistency with the initial data. Finally, we stress that the PSNR behaviour reported in Fig. 3c and d for the particular *butterfly* test problem are common to all the other tests performed.

All these considerations allow us to state that DIP-WTV and cDIP-RED are more robust than DIP and DeepRED with respect to the choice of the hyperparameters values. Moreover, independently on the penalties parameters setting, if compared to the standard DIP, we can stop DIP-WTV and cDIP-RED being confident that these methods do not overfit noise.

Satisfying the Morozov's discrepancy principle In Fig. 4, we consider once again the denoising test on the *butterfly* image described previously. We analyze the behaviour of the constraint ratio $\|f(\theta^{(k)}, z) - \boldsymbol{g}\|/\delta$ as a function of the iterations number. We remark that a constraint ratio equal to 1 entails the corresponding iterate is almost at the boundary of D_{σ_η} defined in (16). We observe that DIP and DeepRED (setting $\gamma = 24$) slowly overfit the simulated noisy acquisition and converge to an interior point of D_{σ_η}. On the other hand, DeepRED (with $\gamma = 22$), DIP-WTV and cDIP-RED converge to a solution which lies on the boundary of D_{σ_η} and hence

Table 1 PSNR mean values for the Set5 for two level of noise. In blue we highlight the best PSNR value

	σ_η	Noisy	DIP	DeepRED	DIP-WTV	cDIP-RED
PSNR	25	25.46	32.29	32.91	32.48	32.95
	50	19.89	27.87	28.15	27.98	28.34

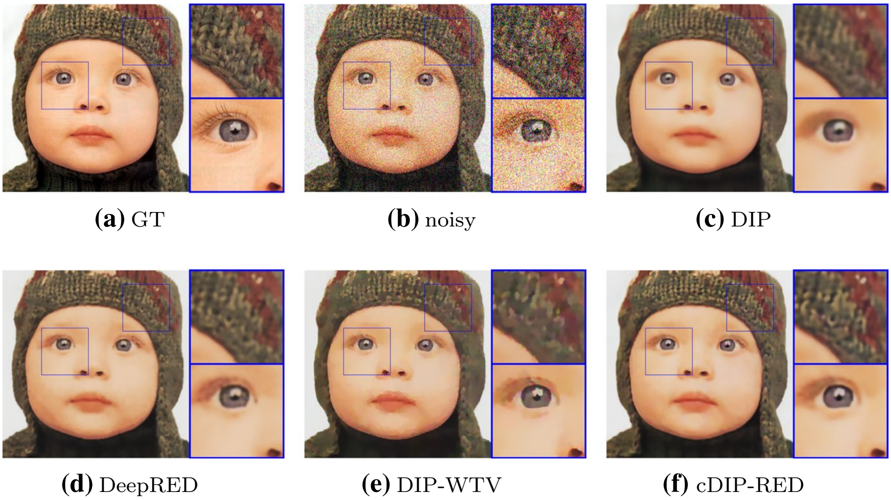


Fig. 5 Restored images for the *baby* test problem setting $\sigma_\eta = 50$. The PSNR values are: Noisy: 19.84 dB, DIP: 27.85 dB, DeepRED: 28.32 dB, DIP-WTV: 28.26 dB, cDIP-RED: 28.43 dB

implicitly satisfy the discrepancy principle. We stress that we empirically observe the same behaviour for all the other experiments performed. As a general comment, this test confirms once again how the performances of DeepRED largely depend on the choice of the hyperparameter γ defining the strength of the regularization. Moreover, we empirically show a more robust convergent behaviour of DIP-WTV and cDIP-RED avoiding costly parameter tuning.

3.3 Denoising task

We validate DIP-WTV and cDIP-RED by comparing them with DIP and DeepRED on the Set5 [4] dataset for the denoising task. The starting noisy images are created by corrupting the ground truth images with an AWGN component of standard deviation equals to 25 and 50. We remark once again that for the cDIP-RED approach we estimate the noise standard deviation even if we know its value. The performances are evaluated by means of the PSNR metric and, in addition, by a visual comparison. In particular, Figs. 5 and 6 report the restored *baby* and *butterfly* images starting from the data with the highest level of corruption considered. In Table 1 we report the mean values of the PSNR metric on Set5. For the DIP algorithm we have selected for each image the number of iteration maximizing the PSNR value. For

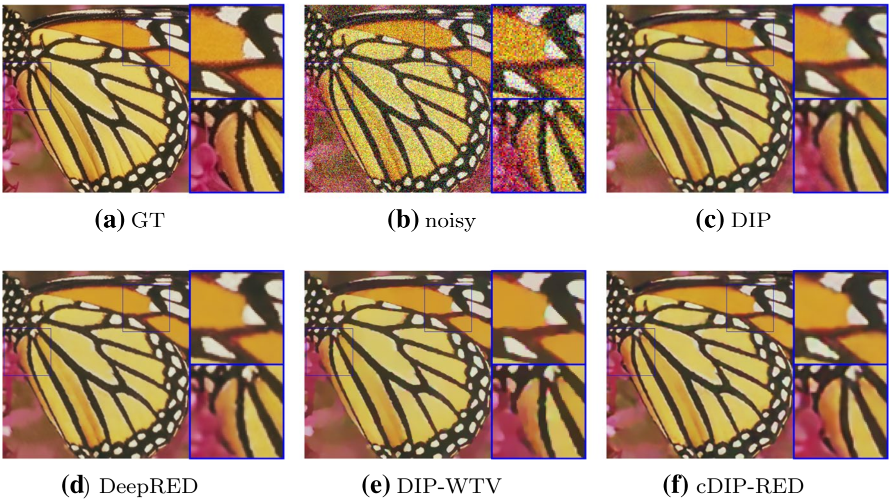

(**a**) GT
(**b**) noisy
(**c**) DIP
(**d**) DeepRED
(**e**) DIP-WTV
(**f**) cDIP-RED

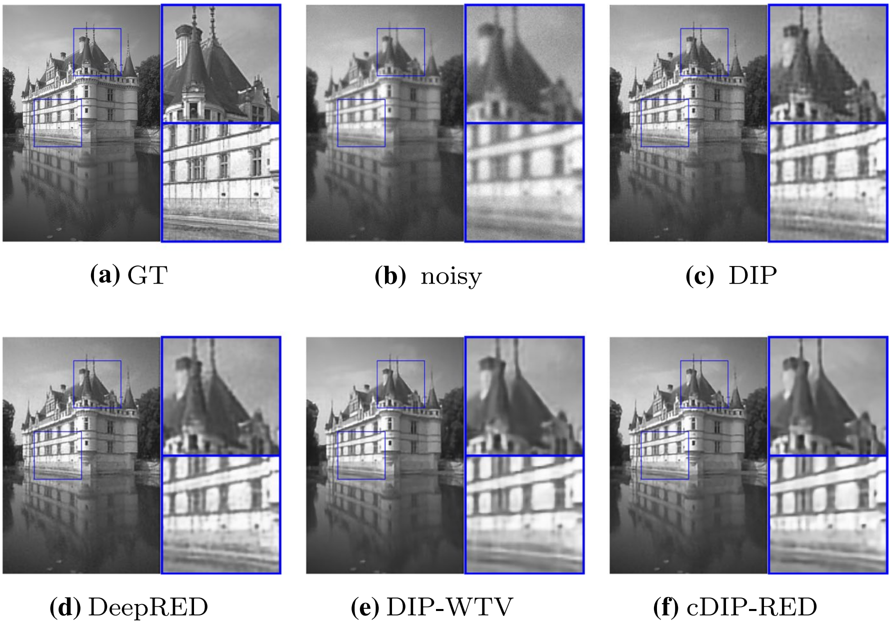
Fig. 6 Restored images for the *butterfly* test problem setting $\sigma_\eta = 50$. The PSNR values are: Noisy: 19.88 dB, DIP: 27.81 dB, DeepRED: 28.13 dB, DIP-WTV: 28.01 dB, cDIP-RED: 28.69 dB

DeepRED we set the ADMM penalty equal to 0.5, whereas we have selected the threshold γ and the starting regularization parameter λ_0 in order to maximize the PSNR for each image. For DIP-WTV and cDIP-RED we set for all the images $\beta_v = 1$ and $\beta_t = 0.5$ and $\beta_r = 1$, respectively. For DeepRED, DIP-WTV and cDIP-RED the restored images have been obtained performing 5000 iterations.

The results reported in Table 1 show that cDIP-RED outperforms DIP and provides slightly better performances with respect to DeepRED in terms of PSNR metric. We remark that cDIP-RED does not require any hand-tuning of the regularization parameter. Concerning DIP-WTV, we observe that it provides better performances than DIP. Moreover, we stress that it has shown more robustness to the choice of the hyperparameters with respect to the DeepRED and it has the lowest number of hyperparameters to be set. Unfortunately, the handcrafted Total Variation regularizer is not as effective as RED regularization for natural images, which manifests in lower PSNR scores for DIP-WTV. In Figs. 5 and 6, we report the simulated noisy acquisitions of the *baby* and *butterfly* images setting $\sigma_\eta = 50$ and the restored images obtained by DIP, DeepRED, DIP-WTV and cDIP-RED. Moreover, in the captions, we highlight the PSNR values. As a general comment, the DIP algorithm struggles to recover the image texture. The cDIP-RED restorations look sharper

Table 2 PSNR and SSIM mean values for the Set5 considering Gaussian blur with $\sigma_H = 2$ and the noise-level $\sigma_\eta = 10$. In blue we highlight the best PSNR and SSIM values

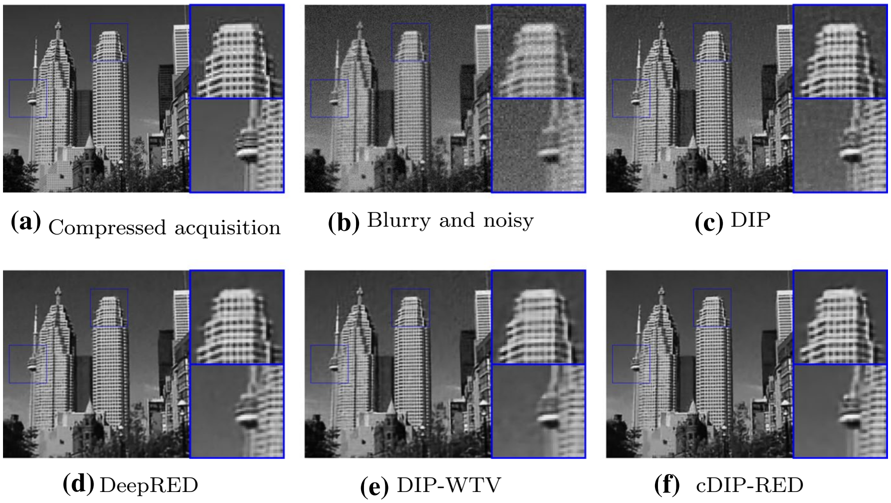
	Blurred	DIP	DeepRED	DIP-WTV	cDIP-RED
PSNR	25.93	30.08	30.81	30.56	30.90
SSIM	0.81	0.91	0.92	0.92	0.93



(a) GT (b) noisy (c) DIP

(d) DeepRED (e) DIP-WTV (f) cDIP-RED

Fig. 7 Restored images for the *watercastle* test problem with noise level 5 and blur 1.6. The PSNR and SSIM values are: Noisy: 22.87 dB–0.76, DIP: 25.81 dB–0.87, DeepRED: 26.23 dB– 0.89, DIP-WTV: 25.87 dB–0.88, cDIP-RED: 26.28 dB–0.89



(a) Compressed acquisition (b) Blurry and noisy (c) DIP

(d) DeepRED (e) DIP-WTV (f) cDIP-RED

Fig. 8 Restored images for the *skyscraper* test problem with noise level 10 and blur 0.8

and more faithful to the ground truth than the ones obtained by DeepRED and DIP-WTV as underlined by the close-ups.

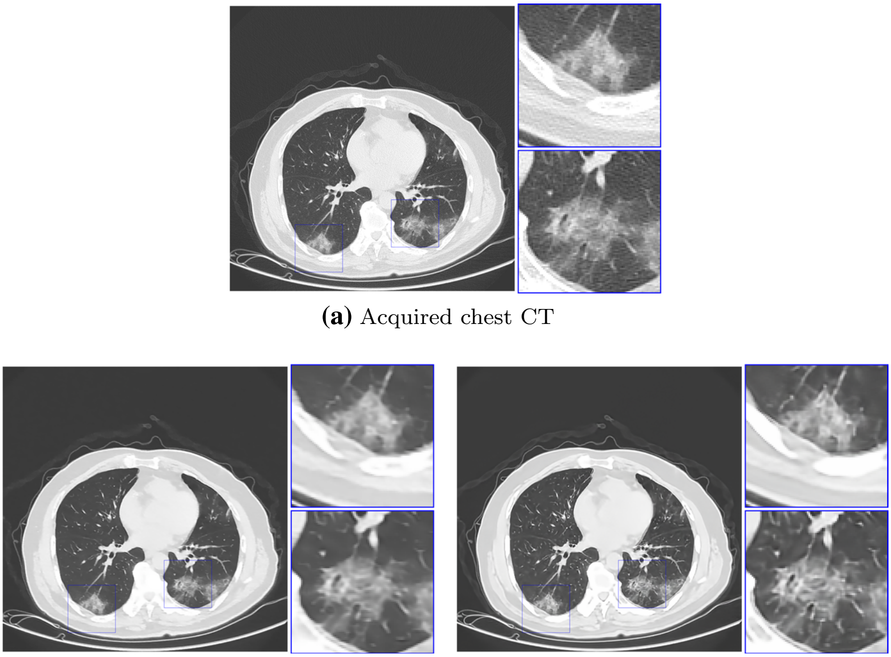
3.4 Deblurring task

In this section, we compare DIP-WTV and cDIP-RED with DIP and DeepRED on the Set5 [4] dataset for the deblurring task. The starting degraded images are constructed by setting the standard deviation of the noise $\sigma_\eta = 10$ and the standard deviation of the Gaussian blur $\sigma_H = 2$. The performances have been evaluated by means of the PSNR and SSIM metrics. In Table 2, we report the mean values of the PSNR and SSIM metrics. Moreover, we consider the *skyscraper* and the *watercastle* images and we add blur and noise by setting $\sigma_\eta = 10$ and $\sigma_H = 0.8$ for the first image, $\sigma_\eta = 5$ and $\sigma_H = 1.6$ for the second. The simulated degraded acquisitions are drawn in Figs. 7b and 8b, respectively. In Figs. 8 and 7, we report the results obtained by applying DIP, DeepRED, DIP-WTV and cDIP-RED and in the caption we report the PSNR and SSIM metrics. For the DIP and DeepRED we set all the hyperparameters in order to maximize the PSNR. For DIP-WTV and cDIP-RED we set for all the tests $\beta_v = 1.5$ and $\beta_t = 1.5$ and $\beta_r = 2$, respectively. For DeepRED, DIP-WTV, and cDIP-RED the restored images have been obtained performing 5000 iterations. From Table 2 we observe again that DeepRED and cDIP-RED reach comparable performances on Set5. However, we stress that, differently from DeepRED, the cDIP-RED scheme does not require to fix the regularization parameter. Moreover, DIP-WTV outperforms the standard DIP. For the *watercastle* image, DeepRED, and cDIP-RED reach similar performances in terms of PSNR and SSIM metrics, however the DeepRED restoration looks noisier than the one provided by cDIP-RED. Finally, DIP-WTV always performs better than the standard DIP.

Concerning the *skyscraper* we do not have a ground truth available, therefore we can compare the results only through visual inspection. Indeed, it is clear from Fig. 8a that the *skyscraper* image is affected by jpeg-compression artifacts. In order to simulate a more realistic acquisition, we further corrupt this compressed image with blur and noise (Fig. 7b). The close-ups in Fig. 8 highlight that the output cDIP-RED suppress the artifacts and outperforms the restorations provided by DIP, DeepRED and DIP-WTV in terms of visual quality. In particular, cDIP-RED can retrieve better the details and remove the artifacts and the noise.

3.5 Artifact removal for a chest CT image

Finally, we show how our methods can be effective for retrieving one real medical chest CT image of a patient affected by COVID-19 [42]. In Fig. 9a we report the acquired data together with the close-ups of two details (inflammation zones) in the lungs backside where are visible the effects of the interstitial pneumonia caused by COVID-19 disease. From these panels the standard artifacts related to the discrete angles sampling typical of the CT application are clearly visible. In Fig. 9b and c, we show the restored images provided by our DIP-WTV and cDIP-RED approaches, respectively. Generally, all finer structures, such as the inflammation details, alveoli and bronchioles, are sufficiently well retrieved, as highlighted by the close-ups. Finally, it is evident that the restoration provided by cDIP-RED looks sharper than the one restored by DIP-WTV.



(a) Acquired chest CT

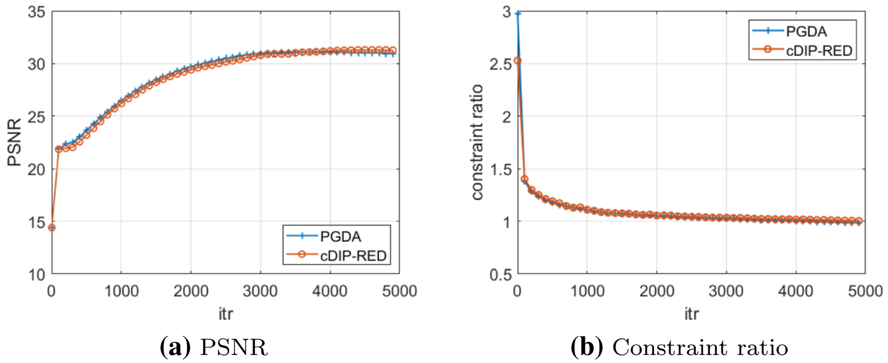
(b) DIP-WTV

(c) cDIP-RED

Fig. 9 Reconstructed images for the real CT problem. In (**a**) we report the acquired data, in (**b**–**c**) we report the restored images obtained by DIP-WTV and cDIP-RED, respectively

3.6 Standard and modified versions of the alternating PGDA method

As already mentioned, both the unconstrained and the constrained models developed in this work have been solved by means of a modified version of the alternating PGDA algorithm. In this section we compare the performance of the standard PGDA and that of the employed modified variant on a denoising problem, by a way of example. Particularly, in this analysis we focus on the sole cDIP-RED approach, since analogous considerations can be deduced for DIP-WTV. In the comparison we consider cDIP-RED exploiting the modified PGDA scheme, and its counterpart which uses the standard PGDA algorithm. In this section, the latter approach is simply denoted by PGDA. We consider the same parameters for cDIP-RED and PGDA. In Fig. 10a we report the PSNR values obtained along the iterations by applying PGDA and cDIP-RED on the denoising problem related to the *woman* image corrupted with AWGN with standard deviation $\sigma_\eta = 35$. It is evident that the achieved values are comparable and the differences between the two methods are very limited. The same considerations can be derived from the behaviour of the constraint ratio generated by the two approaches under consideration and depicted in Fig. 10b: the two curves are almost indistinguishable. Finally, we remark that exploiting the already computed values of to perform the successive variables updates, avoids storing the intermediate values and, for this reason, allows a lower memory requirement.



(a) PSNR **(b)** Constraint ratio

Fig. 10 The PSNR values and the constraint ratio's trend achieved by PGDA and cDIP-RED along the iterations for the *woman* RGB image with $\sigma_\eta = 35$

4 Conclusion

In this paper, we propose a constrained and an unconstrained DIP optimization models which automatically estimate the strength of the regularization. The unconstrained one uses a space variant handcrafted regularizer whose local regularization parameters are adaptively defined along the optimization process, whereas the constrained model is tailored for a generic regularizer and implicitly forces solutions satisfying the discrepancy principle. Particularly, we used the space variant Total Variation and the RED regularizer in the implementation for the unconstrained and the constrained models, respectively. The main strengths of the developed frameworks are threefold: it is not required to set proper values for the regularization parameter, the schemes implemented are more robust with respect to the selection of the hyperparameters than other state-of-the-art DIP-based methods, and both schemes avoid the typical overfitting behaviour of the DIP framework. The numerical experiments on image denoising and deblurring show comparable results of the developed approaches with respect to state-of-the-art strategies with the great advantage of avoiding costly parameter tuning. Finally, since in the literature highly inexact version of ADMM have been used to solve the regularized DIP models, framing the problem in the PGDA setting opens new possibilities to the theoretical convergence analysis and a more faithful match between theory and practical implementation.

A: The alternating proximal gradient descent-ascent method

The alternating proximal gradient descent-ascent (PGDA) method has been proposed [6, 9, 27] to face a saddle point problem of the form

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \max_{\boldsymbol{y}\in\mathbb{R}^n} \; \Phi(\boldsymbol{x},\boldsymbol{y}) + \mathcal{R}(\boldsymbol{x}) - h(\boldsymbol{y}), \tag{22}$$

where the hypotheses on the coupling function $\Phi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$, and the regularizers $\mathcal{R} : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ and $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are stated in the following assumptions. We consider the approach developed in [6].

Assumption 1 The coupling function Φ is

(i) ρ-weakly convex in the first component uniformly in the second one, i.e.

$$\Phi(\cdot, \boldsymbol{y}) + \frac{\rho}{2}||\cdot||^2 \textit{ is convex for all } \boldsymbol{y} \in \mathbb{R}^n$$

(ii) concave and $\nabla\Phi$ is $L_{\nabla\Phi}$-Lipschitz continuous in the second component uniformly in the first one, i.e.

$$||\nabla\Phi(\boldsymbol{x}, \boldsymbol{y}) - \nabla\Phi(\boldsymbol{x}, \boldsymbol{y}')|| \le L_{\nabla\Phi}||\boldsymbol{y} - \boldsymbol{y}'||, \ \forall\, \boldsymbol{x} \in \mathbb{R}^d, \forall\, \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^n.$$

Assumption 2 The function Φ is L-Lipschitz in the first component uniformly over dom h in the second one, i.e.

$$||\Phi(\boldsymbol{x}, \boldsymbol{y}) - \Phi(\boldsymbol{x}', \boldsymbol{y})|| \le L||\boldsymbol{x} - \boldsymbol{x}'||, \ \forall\, \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d, \boldsymbol{y} \in \text{dom } h.$$

Assumption 3 The regularizers $\mathcal{R}$ and h are proper, convex and lower semicontinuous.

(i) Additionally, $\mathcal{R}$ is $L_{\mathcal{R}}$-Lipschitz continuous on its domain, which is assumed to be open.
(ii) Furthermore, h has bounded domain dom h such that the diameter of dom h is bounded by C_h.

It is worth to remark that both the problems (10) and (18) can be cast as in (22) with $\Phi(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{x}, \boldsymbol{y})$, where, for the first problem, $\boldsymbol{y} = \boldsymbol{\mu}_v$. The regularizer h is not present neither in the formulation (10) nor in the (18) one.

Before introducing the PGDA scheme, we need to clarify which is the notion of solution for problem (22) we mean. Indeed, if the minimax problem is not convex-concave, the notion of saddle point is too strong. As done in [6], we focus on the stationarity of the so called *max function* given by

$$\varphi(\boldsymbol{x}) = \max_{y \in \mathbb{R}^n} \Phi(\boldsymbol{x}, \boldsymbol{y}) - h(\boldsymbol{y}), \quad \text{where } \varphi : \mathbb{R}^d \to \mathbb{R},$$

which can be proved to be a ρ-weakly convex function for some $\rho \ge 0$. To focus on the max function makes sense in our framework: indeed, for the problem (10), the relevant variable, corresponding to the needed weights, is the primal one, i.e., $\boldsymbol{x}$. In order to define optimality in terms of the max function we need to define the regularized max function:

$$m(\boldsymbol{x}) = \varphi(\boldsymbol{x}) + \mathcal{R}(\boldsymbol{x}), \quad \text{where } m : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}. \tag{23}$$

In the PGDA framework, the max function is in general nonsmooth, which makes it nonobvious how to define near stationarity. For this reason, a smooth approximation of m, known as Moreau envelope m_δ, parametrized by a positive scalar δ, has been introduced. In more detail, for the proper, ρ-weakly convex and lower semicontinuous function m, the Moreau envelope of m with the parameter $\delta \in (0, \rho^{-1})$ is the function from $\mathbb{R}^d \to \mathbb{R}$ defined by

$$m_\delta(\boldsymbol{x}) := \inf_{z \in \mathbb{R}^d} \left\{ m(\boldsymbol{x}) + \frac{1}{2\delta} ||z - \boldsymbol{x}||^2 \right\}.$$

The Moreau envelope allows to naturally define a notion of near stationarity even for nonsmooth and ρ-weakly convex functions.

Definition 1 For an $\varepsilon > 0$ and a $\delta \in (0, \rho^{-1})$, a point $\boldsymbol{x}$ is ε-stationary for m if $||\nabla m_\delta(\boldsymbol{x})|| \leq \varepsilon$.

Now the algorithm and its convergence properties can be introduced. For initial values $(\boldsymbol{x}^0, \boldsymbol{y}^0) \in \operatorname{dom} \mathcal{R} \times \operatorname{dom} h$, the alternating PGDA method reads as

$$\begin{cases} \boldsymbol{x}^{k+1} = \operatorname{prox}_{\eta_x \mathcal{R}} \big(\boldsymbol{x}^k - \eta_x \nabla_x \Phi(\boldsymbol{x}^k, \boldsymbol{y}^k) \big) \\ \boldsymbol{y}^{k+1} = \operatorname{prox}_{\eta_y h} \big(\boldsymbol{y}^k + \eta_y \nabla_y \Phi(\boldsymbol{x}^{k+1}, \boldsymbol{y}^k) \big) \end{cases} \tag{24}$$

where η_x and η_y are proper positive learning rates.

Theorem 1 *Let Assumptions 1, 2 and 3 hold true and the function m is lower bounded. The iterates generated by the algorithm (24) with $\eta_x = \mathcal{O}(\varepsilon^4) < \frac{1}{2\rho}$, $\eta_y = \frac{1}{L_{\nabla\Phi}}$ and $\delta = \frac{1}{2\rho}$ visit an ε-stationary point in at most $K = \mathcal{O}(\varepsilon^{-6})$*

References

1. Arridge, S., Maass, P., Öktem, O., Schönlieb, C.B.: Solving inverse problems using data-driven models. Acta Numer. **28**, 1–174 (2019)
2. Baguer, D.O., Leuschner, J., Schmidt, M.: Computed tomography reconstruction using Deep Image Prior and learned reconstruction methods. Inverse Prob. **36**(9), 094004 (2020). https://doi.org/10.1088/1361-6420/aba415
3. Bertero, M., Boccacci, P.: Introduction to inverse problems in imaging. CRC press (1998)
4. Bevilacqua, M., Roumy, A., Guillemot, C., line Alberi Morel, M.: Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In: Proceedings of the British Machine Vision Conference, pp. 135.1–135.10. BMVA Press (2012). https://doi.org/10.5244/C.26.135
5. Bortolotti, V., Brown, R., Fantazzini, P., Landi, G., Zama, F.: Uniform Penalty inversion of two-dimensional NMR relaxation data. Inverse Prob. **33**(1), 015003 (2016)
6. Boţ, R.I., Böhm, A.: Alternating proximal-gradient steps for (stochastic) nonconvex-concave minimax problems. arXiv preprint arXiv:2007.13605 (2020)
7. Cascarano, P., Comes, M.C., Mencattini, A., Parrini, M.C., Piccolomini, E.L., Martinelli, E.: Recursive deep prior video: a super resolution algorithm for time-lapse microscopy of organ-on-chip experiments. Medical Image Analysis p. 102124 (2021)

8. Cascarano, P., Sebastiani, A., Comes, M.C., Franchini, G., Porta, F.: Combining weighted total variation and deep image prior for natural and medical image restoration via admm. In: 2021 21st International Conference on Computational Science and Its Applications (ICCSA), pp. 39–46 (2021). https://doi.org/10.1109/ICCSA54496.2021.00016
9. Chen, Z., Zhou, Y., Xu, T., Liang, Y.: Proximal gradient descent-ascent: variable convergence under Kł Geometry. In: International Conference on Learning Representations (2021)
10. Cheng, Z., Gadelha, M., Maji, S., Sheldon, D.: A Bayesian perspective on the Deep Image Prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5443–5451 (2019)
11. Dittmer, S., Kluth, T., Maass, P., Baguer, D.O.: Regularization by architecture: a deep prior approach for inverse problems. J. Math. Imag. Vis. **62**(3), 456–470 (2020)
12. Gan, W., Eldeniz, C., Liu, J., Chen, S., An, H., Kamilov, U.S.: Image Reconstruction for MRI using Deep CNN Priors Trained without Groundtruth. In: 2020 54th Asilomar Conference on Signals, Systems, and Computers, pp. 475–479. IEEE (2020)
13. Gandelsman, Y., Shocher, A., Irani, M.: "Double-DIP": unsupervised image decomposition via coupled deep-image-priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11026–11035 (2019)
14. Golub, G.H., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics **21**(2), 215–223 (1979)
15. Goyal, B., Dogra, A., Agrawal, S., Sohi, B., Sharma, A.: Image denoising review: from classical to state-of-the-art approaches. Inform. Fusion **55**, 220–244 (2020)
16. Grasmair, M.: Locally adaptive total variation regularization. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 331–342. Springer (2009)
17. Hansen, P.C.: Analysis of discrete ill-posed problems by means of the L-curve. SIAM Rev. **34**(4), 561–580 (1992)
18. He, C., Hu, C., Zhang, W., Shi, B.: A fast adaptive parameter estimation for total variation image restoration. IEEE Trans. Image Process. **23**(12), 4954–4967 (2014)
19. Immerkaer, J.: Fast noise variance estimation. Comput. Vis. Image Underst. **64**(2), 300–302 (1996)
20. Jin, K.H., McCann, M.T., Froustey, E., Unser, M.: Deep convolutional neural network for inverse problems in imaging. IEEE Trans. Image Process. **26**(9), 4509–4522 (2017)
21. Karl, W.C.: Regularization in image restoration and reconstruction. In: Handbook of Image and Video Processing, pp. 183–V. Elsevier (2005)
22. Kingma, D.P., Ba, J.: ADAM: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Kobler, E., Effland, A., Kunisch, K., Pock, T.: Total deep variation for linear inverse problems. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
24. Kokil, P., Pratap, T.: Additive white gaussian noise level estimation for natural images using linear scale-space features. Circ. Syst. Signal Process. **40**(1), 353–374 (2021)
25. Krull, A., Buchholz, T.O., Jug, F.: Noise2void-learning denoising from single noisy images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2129–2137 (2019)
26. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2Noise: learning image restoration without clean data. In: International Conference on Machine Learning, pp. 2965–2974. PMLR (2018)
27. Lin, T., Jin, C., Jordan, M.I.: On Gradient Descent Ascent for Nonconvex-Concave Minimax problems. arXiv preprint arXiv:1906.00331 (2021)
28. Lin, Y., Wohlberg, B., Guo, H.: UPRE method for total variation parameter selection. Signal Process. **90**(8), 2546–2551 (2010)
29. Liu, J., Sun, Y., Xu, X., Kamilov, U.S.: Image restoration using Total Variation regularized Deep Image Prior. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7715–7719. IEEE (2019)
30. Lucas, A., Iliadis, M., Molina, R., Katsaggelos, A.K.: Using deep neural networks for inverse problems in imaging: beyond analytical methods. IEEE Signal Process. Mag. **35**(1), 20–36 (2018)
31. Mataev, G., Milanfar, P., Elad, M.: DeepRED: Deep Image Prior powered by RED. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 1–10 (2019)
32. McCann, M.T., Jin, K.H., Unser, M.: Convolutional neural networks for inverse problems in imaging: a review. IEEE Signal Process. Mag. **34**(6), 85–95 (2017)

33. Ng, M.K., Weiss, P., Yuan, X.: Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. SIAM J. Sci. Comput. **32**(5), 2710–2736 (2010)
34. Romano, Y., Elad, M., Milanfar, P.: The little engine that could: regularization by denoising (red). SIAM J. Imag. Sci. **10**(4), 1804–1844 (2017)
35. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**(1–4), 259–268 (1992)
36. Sagheer, S.V.M., George, S.N.: A review on medical image denoising algorithms. Biomed. Signal Process. Control **61**, 102036 (2020)
37. Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., Lenzen, F.: Variational methods in imaging. Applied mathematical sciences ; v. 167. Springer, New York (2009)
38. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep Image Prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9446–9454 (2018)
39. Van Veen, D., Jalal, A., Soltanolkotabi, M., Price, E., Vishwanath, S., Dimakis, A.G.: Compressed sensing with deep image prior and learned regularization. arXiv preprint arXiv:1806.06438 (2018)
40. Wen, Y.W., Yip, A.M.: Adaptive parameter selection for total variation image deconvolution. Numer. Math. Theor. Meth. Appl **2**(4), 427–438 (2009)
41. Willemink, M.J., Koszek, W.A., Hardell, C., Wu, J., Fleischmann, D., Harvey, H., Folio, L.R., Summers, R.M., Rubin, D.L., Lungren, M.P.: Preparing medical imaging data for machine learning. Radiology **295**(1), 4–15 (2020)
42. Yan, T., Wong, P.K., Ren, H., Wang, H., Wang, J., Li, Y.: Automatic distinction between COVID-19 and common pneumonia using multi-scale convolutional neural network on chest CT scans. Chaos, Solit. Fract. **140**, 110153 (2020)
43. Zanni, L., Benfenati, A., Bertero, M., Ruggiero, V.: Numerical methods for parameter estimation in poisson data inversion. J. Math. Imag. Vis. **52**(3), 397–413 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.