Dual Forward-Backward Unfolded Network for Flexible Plug-and-Play

Audrey Repetti*†, Matthieu Terris†, Yves Wiaux†, and Jean-Christophe Pesquet‡
*Department of Actuarial Mathematics & Statistics, Heriot-Watt University, Edinburgh EH14 4AS, UK
†Institute of Sensors, Signals and Systems, Heriot-Watt University, Edinburgh EH14 4AS, UK
‡ Université Paris-Saclay, CentraleSupélec, Inria, Center for Visual Computing, 91190 Gif sur Yvette, France

***Abstract*—Proximal methods have been extensively used to find maximum a posteriori (MAP) estimates of unknown images from degraded measurement. Recently, they have been mixed with neural networks (NN) to further improve the reconstruction quality. Two approaches can be distinguished: unfolded NNs, implementing a given iteration number of an optimisation algorithm, and plug-and-play (PnP) algorithms, incorporating NNs in existing optimisation algorithms. Unfolded NNs usually incorporate the measurement operator in the learning process, which can be prohibitive for applications with non-fixed measurement operators. PnP do not have this drawback, but involved NNs still depend on the underlying statistical models (e.g., higher noise level on the measurements requires stronger denoisers). In this work, we propose a PnP algorithm based on forward-backward (FB) iterations, where the learned denoiser is an unfolded NN based on dual-FB iterations. This NN is built to mimic a Gaussian denoiser from a MAP viewpoint. This allows us to introduce a regularisation parameter in the model to tune the regularization strength, similarly to standard variational approaches. This has the advantage of making the learned NN more adaptive to a variety of inverse problem statistical models, without requiring to train the NN for different noise levels.**

***Index Terms*—Dual forward-backward, unfolded network, plug-and-play algorithm, inverse problem, image restoration.**

I. Introduction

Image restoration problems can often be formulated as inverse problems, where the objective is to estimate an unknown image $\overline{x} \in \mathbb{R}^N$ from degraded measurements $y \in \mathbb{R}^M$. In this context, the model is given by

$$y = H\overline{x} + w, \tag{1}$$

where $H \colon \mathbb{R}^N \to \mathbb{R}^M$ is a linear measurement operator and $w \in \mathbb{R}^M$ is a realization of an additive white Gaussian noise, with standard deviation $\sigma > 0$. This problem is often solved by defining the estimate $\widehat{x} \in \mathbb{R}^N$ of $\overline{x}$ as

$$\widehat{x} \in \underset{x \in \mathbb{R}^N}{\operatorname{Argmin}} \; \frac{1}{2}\|Hx - y\|^2 + \lambda g(Lx), \tag{2}$$

where $g \colon \mathbb{R}^P \quad \to]-\infty, +\infty]$ is a proper lower-semicontinuous convex regularization function incorporating *a priori* information on $\overline{x}$, L is a linear operator from $\mathbb{R}^N$ to $\mathbb{R}^P$, and $\rho > 0$ is a parameter tuning the regularization strength.

Problem (2) can be solved efficiently using optimisation algorithms [1], [2]. When g is nonsmooth and proximable (e.g., ℓ_1 norm composed with an orthonormal basis), one of the simplest schemes to solve (2) is the forward-backward (FB) algorithm [3]. It alternates between a gradient step on the least-squares criterion and a proximity step on g. When g has a more sophisticated form (e.g. total variation norm or hybrid regularization), one can choose to use more advanced algorithms such as primal-dual methods [2], [4], [5], or to use a dedicated algorithm to compute sub-iteratively the associated proximal operator within the FB algorithm. For the second option, a common approach is to use the dual-FB algorithm [6], designed to compute proximity operators.

Recently, performances of optimisation algorithms (for reconstruction quality) have been improved by coupling them with neural network (NN) models. The two main approaches are (i) unfolded NNs, unrolling optimisation algorithmscoupled with learnable operators over a fixed number of iterations, and (ii) plug-and-play (PnP) algorithms, injecting denoising NNs in optimisation algorithms. Both the two approaches have shown their efficiency over a number of applications (see, e.g., [7]–[9]). For inverse problems, and particularly imaging problems, PnP algorithms may be preferred over unfolded NNs, as the involved NN is independent from the measurement operator. Indeed, for unfolded NNs, the measurement operator is included in the layers, hence in the training process, which makes the trained network measurement-specific. This drawback is not observed for PnP methods, where NNs are denoising operators (for regularizing the solution) not interfering with the measurement operator. Nevertheless the involved NN still depends on the underlying statistical models, i.e., the noise standard deviation σ. To circumvent this difficulty, some networks can be trained using variable training noise. Recently, DRUNet [9], [10] was proposed to introduce the noise level as an extra parameter and further improve its performance.

PnP algorithms were first proposed in [11], where the authors replaced the proximity operator of the prior g with a denoiser. If early methods relied on variational denoisers, such as BM3D [12], later progress of NNs in denoising tasks have allowed to boost reconstruction capabilities of PnP algorithms. Recently, much attention has been drawn to convergence and stability properties of PnP algorithms. We can draw two main approaches. The first one consists in enforcing regularity constraint on the NN of interest, in order to mimic the behavior of resolvent operators and links with a monotone inclusion problem [13]–[15]. The second approach consists in exploiting Tweedie's formula, and relating the DNN with the gradient of a prior function g [10], [16], [17].

In this work, we propose a PnP algorithm, based on FB iterations, where the proximity operator has been replaced by an unfolded dual-FB network, dubbed DFBNet. The architecture of DFBNet mimics the structure of a Gaussian denoiser from a maximum *a posteriori* (MAP) perspective. Inspired by standard variational approaches, we include a regularization parameter in the model. This allows us to control in a flexible manner the strength of the regularizer employed in the PnP algorithm.

The remainder of this paper is organized as follows: The proposed approach is described in Section II. Simulation results are provided in Section III. Finally, our conclusions are given in Section IV.

II. Proposed method

The architecture of the proposed DFBNet is grounded on the structure of the dual-FB algorithm, to mimic a Gaussian MAP denoising operator.

A. Dual-FB for denoising problem

Let $\overline{x} \in \mathbb{R}^N$ be an original unknown image, and $z = \overline{x} + n$ be a noisy observation of $\overline{x}$, where $n \in \mathbb{R}^N$ is a realization of a random normal i.i.d. variable with zero-mean and standard deviation $v > 0$. Then we aim to build a NN G such that $\mathrm{G}(z) \approx x^*$, where $x^* \in \mathbb{R}^N$ is a good estimate of $\overline{x}$. A possible estimate corresponds to the penalized least squares estimate of $\overline{x}$, defined as

$$x^* = \mathrm{prox}_{\eta g \circ L}(z) = \underset{x \in \mathcal{C}}{\mathrm{argmin}}\ \frac{1}{2}\|x - z\|^2 + \eta g(Lx), \tag{3}$$

where $L \colon \mathbb{R}^N \to \mathbb{R}^P$ represents some data transform, $\eta > 0$ is the associated regularization parameter, and $\mathcal{C} \subset \mathbb{R}^N$ is a nonempty, closed, convex set. Function g constitutes a regularization term for the denoising problem. A standard strategy is to promote the sparsity in the transformed space by setting $g = \| \cdot \|_1$.

Let us define, for every $u \in \mathbb{R}^P$ and $\gamma \in]0, +\infty[$,

$$T_{\eta,\gamma,L}(u) = \left(\mathrm{Id} - \mathrm{prox}_{\eta\gamma^{-1}g}\right)\left(u + L\,\mathrm{proj}_{\mathcal{C}}\left(z - \gamma L^* u\right)\right) \tag{4}$$

where $\mathrm{proj}_{\mathcal{C}}$ denotes the projection operator onto the set $\mathcal{C}$, and $\mathrm{prox}_{\eta\gamma^{-1}g}$ denotes the proximity operator of $\eta\gamma^{-1}g$. When $g = \| \cdot \|_1$, $\mathrm{Id} - \mathrm{prox}_{\eta\gamma^{-1}g}$ reduces to the projection onto the hypercube $[-\eta\gamma^{-1}, \eta\gamma^{-1}]^N$. The dual-FB algorithm [6], [18] for solving (3) then reads

$$\begin{array}{l} u_0 \in \mathbb{R}^P, \\ \text{For } k = 0, 1, \dots \\ \left\lfloor\ u_{k+1} = T_{\eta,\gamma_k,L}(u_k). \right. \end{array} \tag{5}$$

where, for every $k \in \mathbb{N}$, $\gamma_k \in]0, 2\|L\|^{-2}[$.

B. DFBNet architecture

The proposed DFBNet with $K \in \mathbb{N}^*$ layers and regularization parameter η unrolls the dual-FB iterations of Algorithm 5. One important features of this algorithm is that it learns different linear operators L for each layer, so adding much flexibility to the standard form of DFB.

Let $\boldsymbol{L}^K = (L_k)_{1 \leqslant k \leqslant K}$ where, for every $k \in \{1, \dots, K\}$, L_k is a linear operator from $\mathbb{R}^N$ to $\mathbb{R}^P$ and $\beta_k = 1/\|L_k\|^2$. DFBNet is expressed, for every $u \in \mathbb{R}^P$, as

$$G^K_{\eta,\boldsymbol{\gamma}^K,\boldsymbol{L}^K}(u, z) = T_{\eta,\gamma_K,L_K} \circ \cdots \circ T_{\eta,\gamma_1,L_1}(u), \tag{6}$$

where $\boldsymbol{\gamma}^K = (\gamma_k)_{1 \leqslant k \leqslant K}$ and, for every $k \in \{1, \dots, K\}$, $\gamma_k \in [\epsilon, 2\beta_k - \epsilon]$ with $\epsilon > 0$.

The asymptotic behaviour of the proposed network, i.e. when the number of layers is very large, can be directly deduced from [18, Thm. 3.7].

Proposition 1 *For every $k \in \mathbb{N}$, let $L_k = L \colon \mathbb{R}^N \to \mathbb{R}^P$. Then we have*

$$x^* = \lim_{K \to +\infty} \mathrm{proj}_{\mathcal{C}}\left(z - \gamma_{K+1} L^* G^K_{\eta,\boldsymbol{\gamma}^K,\boldsymbol{L}^K}(u_0, z)\right), \tag{7}$$

for any initial choice for $u_0 \in \mathbb{R}^P$.

Motivated by this result we define, for every $u_0 \in \mathbb{R}^P$,

$$\begin{aligned} &\overline{G}^K_{\eta,\boldsymbol{\gamma}^{K+1},\boldsymbol{L}^{K+1}}(u_0, z) \\ &\qquad = \mathrm{proj}_{\mathcal{C}}\left(z - \gamma_{K+1} L^*_{K+1} G^K_{\eta,\boldsymbol{\gamma}^K,\boldsymbol{L}^K}(u_0, z)\right), \end{aligned} \tag{8}$$

where $L_{K+1} \colon \mathbb{R}^N \to \mathbb{R}^P$, and $\gamma_{K+1} \in [\epsilon, 2\beta_{K+1} - \epsilon]$ with $\epsilon > 0$ and $\beta_{K+1} = 1/\|L_{K+1}\|^2$.

Note that, when all layers have the same learned linear operator, then the proposed approach could be interpreted as a dictionary learning approach, while the general case with multiple operators shows similarities with deep dictionary learning [19].

C. PnP-FB algorithm

To find an estimate of $\overline{x}$ from the degraded measurements y given in (1), we propose to use a PnP algorithm based on FB iterations, where the proximity operator has been replaced by DFBNet described in Section II-B. The proposed iterative approach is then given by

$$\begin{array}{l} x_0 \in \mathbb{R}^N, u_0 = L_{K+1} x_0 \\ \text{For } \ell = 0, 1, \dots \\ \left\lfloor \begin{array}{l} z_\ell = x_\ell - \mu_\ell H^* \left(H x_\ell - y\right), \\ u_{\ell+1} = G^K_{\mu_\ell \lambda,\boldsymbol{\gamma}^K,\boldsymbol{L}^K}(u_\ell, z_\ell), \\ x_{\ell+1} = \mathrm{proj}_{\mathcal{C}}\left(z_\ell - \gamma_{K+1} L^*_{K+1} u_{\ell+1}\right), \end{array} \right. \end{array} \tag{9}$$

where $\lambda > 0$ and, for every $\ell \in \mathbb{N}$, $\mu_\ell \in [\epsilon, 2/\|H\|^2 - \epsilon]$, with $\epsilon > 0$. The parameter λ is used to control the strength of DFNet, as regularization parameters are used by traditional variational methods. A layer of the DFBNet architecture is detailed in Figure 1.

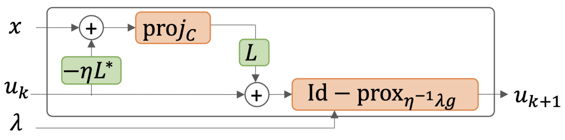

Fig. 1. Illustration of one layer of the DFBNet.

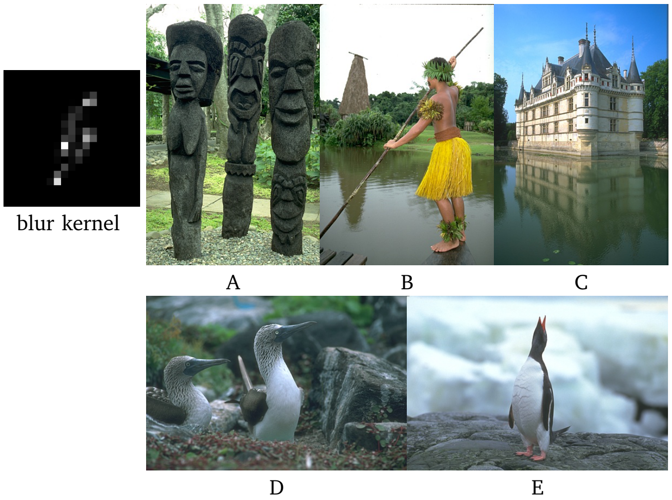

Fig. 2. Left: Convolution kernel. Right: Original test images $\overline{x}$ for the inverse problem (1).

At iteration $\ell \in \mathbb{N}$ in Algorithm (9), DFBNet is initialized with warm restart, i.e., with the dual variable u_ℓ obtained at iteration $\ell - 1$ (as output of DFNet). This is motivated by the practical fact that when a dual-FB algorithm is used within a FB algorithm, only a small number of iterations is necessary to obtain the desired proximity operator, as soon as a good input dual variable is provided to the dual-FB algorithm (i.e., after few global iterations of the FB algorithm). This is related to the monotonic behaviour of the dual-FB algorithm on the dual variable.

Using Proposition 1 and [3], one can deduce the asymptotic behaviour of algorithm (9) in the particular case when a unique linear operator is used for all the layers of DFBNet, and a large number of layers is considered.

Proposition 2 *Assume that Problem 2 has a solution. For every $k \in \{0, \ldots, K+1\}$, let $L_k = L \colon \mathbb{R}^N \to \mathbb{R}^P$, and assume that $K \to +\infty$. Then $(x_\ell)_{\ell \in \mathbb{N}}$ converges to a solution $\widehat{x}$ to Problem* (2).

III. Application to image deblurring

A. Simulation setting

We consider an inverse imaging problem of the form (1), with RGB images of size $N = 3 \times \overline{N}$, where $H \in \mathbb{R}^{N \times N}$ is a blurring operator with convolution kernel displayed in Fig. 2, and $\sigma \in \{0.03, 0.05, 0.08\}$. We evaluate the results on five RGB images from BSD68 [20] shown in Fig. 2.

	A	B	C	D	E	av. time (s.)
$\sigma = 0.03$	12.78	19.29	16.06	16.24	22.54	
DFBNet	16.33	24.81	22.2	20.41	28.34	33.2
DRUNet	16.17	23.86	21.3	20.53	28.84	15.8
BM3D	17.38	25.52	22.28	21.86	29.45	833.8
DnCNN	13.05	18.36	15.55	13.03	18.44	13.9
[15] $(\upsilon = 0.021)$	17.36	26.14	23.41	21.95	29.91	57.5
TGV	15.63	22.34	19.24	19.21	25.97	226.8
$\sigma = 0.05$	12.12	18.11	15.12	14.58	20.60	
DFBNet	14.97	23.54	20.65	19.31	27.47	24.2
DRUNet	14.94	23.59	20.15	19.64	27.18	11.6
BM3D	15.86	23.57	20.62	20.02	28.00	440.4
DnCNN	9.96	13.56	11.35	9.11	14.16	19.88
[15] $(\upsilon = 0.035)$	16.29	24.45	21.57	20.22	28.19	22.6
TGV	14.64	21.62	18.40	18.39	25.59	135.9
$\sigma = 0.08$	10.82	16.11	13.44	12.10	17.85	
DFBNet	14.49	22.75	19.68	18.71	26.93	32.7
DRUNet	14.63	23.13	19.65	18.92	27.37	13.8
BM3D	14.93	22.90	19.86	18.95	26.88	667.5
DnCNN	6.21	9.72	8.14	5.29	10.91	18.8
[15] $(\upsilon = 0.042)$	13.72	21.24	18.67	16.22	24.72	33.0
TGV	13. 92	20.88	17.61	17.06	24.39	145.4

TABLE I

SNR (dB) results obtained with the different methods, on the five test images, for three different noise levels $\sigma \in \{0.03, 0.05, 0.08\}$. For each noise level, the first row provides the input SNR of y. The last column gives the average computing time per algorithm over the five images.

We compare our method with other PnP-FB algorithms, where the denoising operator corresponds to either DRUNet [9], BM3D [12], DnCNN without batch norms (see, e.g., [21]), or DnCNN without batch norms trained to be firmly nonexpansive for convergence guaranties [15]. Note that all these denoisers, apart from the DnCNN from [15], can be applied to different statistical models, without needing to train different NNs. We also compare with a variational approach for RGB images using FB algorithm with the proximity operator of the total generalized variation (TGV) penalisation [22].

B. Training procedure

Let $\overline{\boldsymbol{x}} = (\overline{x}_i)_{1 \leqslant i \leqslant I}$ be a dataset of I images with values in $[0,1]$. For every $i \in \{1, \ldots, I\}$, we build a noisy observation of $\overline{x}_i$, given by $z_i = \overline{x}_i + n_i$, where $n_i \in \mathbb{R}^N$ is a realization of a random normal i.i.d. variable with zero-mean and standard deviation υ_i. We propose to train the DFBNet as a denoiser. To this aim, we

$$\underset{\boldsymbol{\theta}^{K+1}}{\text{minimize}} \sum_{i=1}^{I} \|\overline{G}^{K}_{\lambda_i, \boldsymbol{\gamma}^{K+1}, \boldsymbol{L}^{K+1}}(L_{K+1} z_i, z_i) - z_i\|_1, \quad (10)$$

where $\boldsymbol{\theta}^{K+1} = (\theta_k)_{1 \leqslant k \leqslant K+1}$ are the learnable parameters of the convolutional operators $\boldsymbol{L}^{K+1}$, and the definition of $\overline{G}^{K}_{\lambda_i, \boldsymbol{\gamma}^{K+1}, \boldsymbol{L}^{K+1}}$ is given in (8). We choose $\mathcal{C} = [0,1]^N$, $\lambda_i = \upsilon_i$, and, for every $k \in \{0, \ldots, K\}$, $\gamma_k = 1.8/\|L_k\|^2$.

We use Adam [23] to solve (10) on pairs of ground truth/noisy patches of size 50×50 built on the $50,000$ test images from ImageNet. We consider $K = 20$ layers,

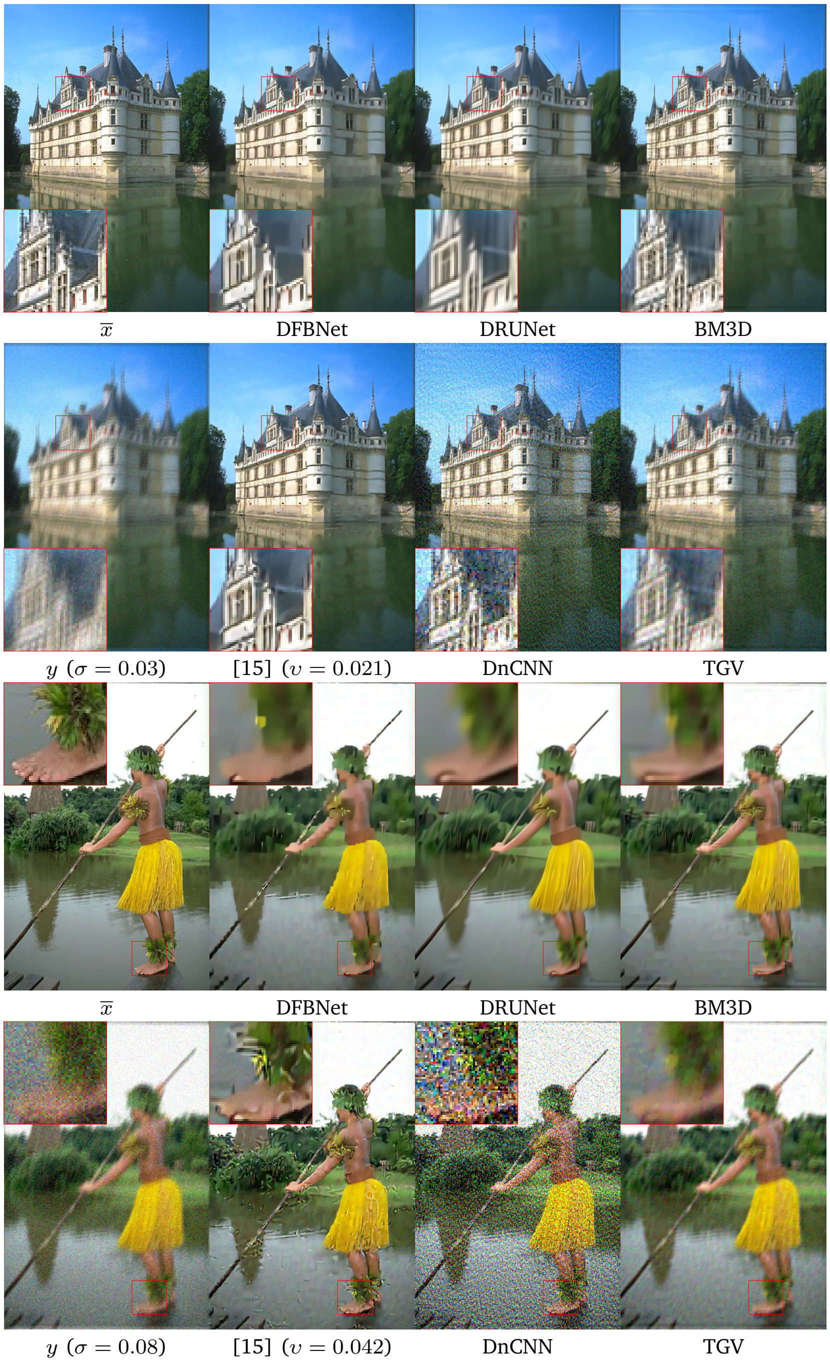

Fig. 3. Results obtained from FB iterations with different denoisers, for two images and two noise levels ($\sigma = 0.03$ and $\sigma = 0.08$). SNR values for these results are given in Table I, for image C with $\sigma = 0.03$ (top) and image B with $\sigma = 0.08$ (bottom).

$P = 64 \times \overline{N}$, batches of size 100, the learning rate is set to 10^{-3} and is divided by 2 every 10^4 iterations.

The standard DnCNN and DFBNet are trained on variable noise, considering random values of $\upsilon_i \sim \mathcal{U}([0, 0.05])$ during the training procedure. The DnCNN from [15] is trained using three fixed noise levels $\upsilon \in \{0.021, 0.035, 0.042\}$, to match the three noise levels in the inverse problem (1) with $\sigma \in \{0.03, 0.05, 0.08\}$, respectively. These values are observed to be the best to approximate the noise level of the input of the network in the PnP algorithm (see Section II-B). For DRUNet, we use the trained NN provided by the authors [9] on GitHub.

C. Simulation results

We run the FB algorithm for the settings and denoisers described in Section III-A. The proposed DFBNet as well as DRUNet, BM3D, and the proximity operator of TGV require to choose a regularization parameter. For DFBNet, DRUNet and BM3D, this parameter should correspond to the noise level of the input noisy variable (i.e., in FB iterations, it is the variable obtained after a gradient step). This is determined empirically by taking the parameters providing the best Signal to Noise Ratio (SNR) on all the images for all methods, per noise level σ. For the considered $\sigma \in \{0.03, 0.05, 0.08\}$, we obtain $\lambda \in \{0.021, 0.035, 0.042\}$, respectively. For the proximity operator of TGV, the parameters have been manually chosen to obtain the best SNR. The two other NNs (DnCNN and [15]) do not have regularization parameters.

The stopping criterion is adapted to the considered method, depending whether it benefits from convergence guaranties or not. The standard FB with TGV regularization and the PnP from [15] both offer strong convergence guarantees. For both these two methods, the algorithm is stopped when $\|x_\ell - x_{\ell-1}\| < 10^{-4}\|x_\ell\|$ holds, where $(x_\ell)_{\ell\in\mathbb{N}}$ is the sequence generated by the either the two algorithms. The three other PnP-FB (grounded on DFBNet, DRUNet and BM3D) benefit from weaker theoretical guarantees. We then propose to stop these algorithms as soon as a non-monotonic behaviour on the generated sequence $(x_\ell)_{\ell\in\mathbb{N}}$ is observed, i.e., if $\|x_{\ell+1} - x_\ell\| > \|x_\ell - x_{\ell-1}\|$.

The SNR results for the different methods, in the different settings are given in Table I. We also provide the average time (in s.) necessary to reach the solution obtained from the different methods. Results delivered by the different approaches for two settings are also provided in Fig. 3 (im. C with $\sigma = 0.03$, and im. B with $\sigma = 0.08$). We observe that DFBNet provides similar results to DRUNet. BM3D generally gives slightly better SNR values, at the price of high computational time (more than 20 times larger). The convergent DnCNN [15] provides better results for lower noise levels. However, unlike the proposed approach, [15] necessitates to train different networks when the noise level changes. In addition, training [15] is much heavier than DFBNet, due to the regularization used to obtain a firmly non-expansive network, and additional pretraining. Visual inspection on the top images of Fig. 3 shows that, even if the SNR value is lower with DFBNet than [15], the accuracy of the detail estimate is similar. Both the classical DnCNN and proximal approaches are outperformed by the other methods.

IV. Conclusion

In this work we have proposed to pair an unfolded dual FB, dubbed DFBNet, with a FB-PnP algorithm. The proposed approach can be adapted to different statistical models, thanks to the use of a regularization parameter, included in the network, and inspired from traditional variational approaches. We have shown through simulation on an image deblurring problem that DFBNet is competitive with state-of-the-art denoisers for PnP algorithms, while being adjustable to the noise level.

References

[1] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212. Springer, 2011.

[2] N. Komodakis and J.-C. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Signal Process. Mag*, vol. 32, no. 6, pp. 31–54, 2015.

[3] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[4] L. Condat, "A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013.

[5] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Advances in Computational Mathematics*, vol. 38, no. 3, pp. 667–681, 2013.

[6] P. L. Combettes, D. Dũng, and B. C. Vũ, "Proximity for sums of composite functions," *J. Math. Anal. Appl.*, vol. 380, no. 2, pp. 680–688, Aug. 2011.

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

[8] R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter, "Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery," *IEEE Signal Process. Mag*, vol. 37, no. 1, pp. 105–116, 2020.

[9] 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.*, 2021.

[10] S. Hurault, A. Leclaire, and N. Papadakis, "Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization," *arXiv preprint arXiv:2201.13256*, 2022.

[11] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conf. on Signal and Inf. Processing*. IEEE, 2013, pp. 945–948.

[12] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process*, vol. 16, no. 8, pp. 2080–2095, 2007.

[13] M. Terris, A. Repetti, J.-C. Pesquet, and Y. Wiaux, "Building firmly nonexpansive convolutional neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8658–8662.

[14] R. Cohen, M. Elad, and P. Milanfar, "Regularization by denoising via fixed-point projection (red-pro)," *SIAM Journal on Imaging Sciences*, vol. 14, no. 3, pp. 1374–1406, 2021.

[15] J.-C. Pesquet, A. Repetti, M. Terris, and Y. Wiaux, "Learning maximally monotone operators for image recovery," *SIAM Journal on Imaging Sciences*, vol. 14, no. 3, pp. 1206–1237, 2021.

[16] X. Xu, Y. Sun, J. Liu, B. Wohlberg, and U. S. Kamilov, "Provable convergence of plug-and-play priors with mmse denoisers," *IEEE Signal Process. Lett.*, vol. 27, pp. 1280–1284, 2020.

[17] R. Laumont, V. De Bortoli, A. Almansa, J. Delon, A. Durmus, and M. Pereyra, "On maximum-a-posteriori estimation with plug & play priors and stochastic gradient descent," *arXiv preprint arXiv:2201.06133*, 2022.

[18] P. L. Combettes, D. Dũng, and B. C. Vũ, "Dualization of signal recovery problems," *Set-Valued and Variational Analysis*, vol. 18, pp. 373–404, Dec. 2010.

[19] Wen Tang, Emilie Chouzenoux, Jean-Christophe Pesquet, and Hamid Krim, "Deep transform and metric learning network: Wedding deep dictionary learning and neural networks," *arXiv preprint arXiv:2002.07898*, 2020.

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, July 2001, vol. 2, pp. 416–423.

[21] T. Meinhardt, M. Moller, C. Hazirbas, and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proc. of the IEEE International Conference on Computer Vision (ICCV17)*, 2017, pp. 1781–1790.

[22] Kristian Bredies, Karl Kunisch, and Thomas Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.

[23] K. Zhang, W. Zuo, and L. Zhang, "Deep plug-and-play super-resolution for arbitrary blur kernels," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit.*, 2019, pp. 1671–1681.