BREGMAN PLUG-AND-PLAY PRIORS

Abdullah H. Al-Shabili[*,1], *Xiaojian Xu*[*,2], *Ivan Selesnick*[1], *and Ulugbek S. Kamilov*[2,3]

[1]Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, USA
[2]Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO, USA
[3]Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA

2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897933

ABSTRACT

The past few years have seen a surge of activity around integration of deep learning networks and optimization algorithms for solving inverse problems. Recent work on plug-and-play priors (PnP), regularization by denoising (RED), and deep unfolding has shown the state-of-the-art performance of such integration in a variety of applications. However, the current paradigm for designing such algorithms is inherently Euclidean, due to the usage of the quadratic norm within the projection and proximal operators. We propose to broaden this perspective by considering a non-Euclidean setting based on the more general Bregman distance. Our new Bregman Proximal Gradient Method variant of PnP (PnP-BPGM) and Bregman Steepest Descent variant of RED (RED-BSD) replace the traditional updates in PnP and RED from the quadratic norms to more general Bregman distance. We present a theoretical convergence result for PnP-BPGM and demonstrate the effectiveness of our algorithms on Poisson linear inverse problems.

Index Terms— Plug-and-play priors, inverse problems, proximal optimization, image reconstruction.

1. INTRODUCTION

The recovery of an unknown signal $\boldsymbol{x} \in \mathbb{R}^n$ from its noisy measurements $\boldsymbol{y} \in \mathbb{R}^m$ can often be formulated as an inverse problem

$$\boldsymbol{y} = \mathcal{N}(\boldsymbol{A}\,\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a measurement operator and $\mathcal{N}$ models the corruption of the measurements by noise, which could be signal dependent (e.g., Poisson noise) or signal independent (e.g., Gaussian noise). The solution of ill-posed inverse problems is often formulated as an optimization problem

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{arg\,min}}\ f(\boldsymbol{x}) + g(\boldsymbol{x}) \tag{2}$$

where f is the data-fidelity term and g is the regularizer.

The past few years have seen a surge of efforts to integrate deep learning (DL) priors into iterative algorithms [1, 2]. *Plug-and-play priors (PnP)* [3] and *regularization by denoising (RED)* [4] are two methods that integrate pre-trained DL denoisers into iterative algorithms. Deep unfolding is a related strategy based on unfolding an iterative algorithm and including trainable blocks within it [5]. Compared to the black-box DL, model-based DL methods integrate the physics-based knowledge of the measurement model. Their empirical success, has spurred a number of algorithmic extensions [6–9], as well as theoretical convergence analyses [10–13].

* Abdullah H. Al-Shabil and Xiaojian Xu contributed equally to this work. This work was supported in part by the NSF Award CCF-2043134.

Most of the current work in PnP is fundamentally based on the traditional definition of the proximal operator that relies on the squared Euclidean norm. Under this definition the proximal operator can be naturally interpreted as the Gaussian denoiser. In this paper, we seek to broaden the family of PnP algorithms to the non-Euclidean setting by building on the recent work on Bregman proximal algorithms [14–16]. Specifically, we propose to generalize the well-known PnP-PGM [6] and RED-SD [4] algorithms to their Bregman counterparts, PnP-BPGM and RED-BSD algorithms, by using the Bregman distance. We learn the corresponding artifact-removal operators by unfolding the iterations of our algorithms. Finally, we present the theoretical convergence analysis of PnP-BPGM and test our algorithms on Poisson linear inverse problems.

2. BACKGROUND

2.1. Proximal Gradient Method

PGM can be interpreted as the Majorize-Minimization (MM) method for solving the composite optimization problem in (2). Each iteration of PGM can be expressed as a minimization of a quadratic majorizer

$$\boldsymbol{x}^{k+1} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{arg\,min}} \left\{ \boldsymbol{x}^{\mathsf{T}}\nabla f(\boldsymbol{x}^k) + \frac{L}{2}\left\|\boldsymbol{x}-\boldsymbol{x}^k\right\|_2^2 + g(\boldsymbol{x}) \right\}, \tag{3}$$

where f is assumed to have a L-Lipschitz continuous gradient. Eq. (3) can also be expressed in the following form

$$\boldsymbol{z}^k = \boldsymbol{x}^k - \gamma\nabla f(\boldsymbol{x}^k) \tag{4a}$$

$$\boldsymbol{x}^{k+1} = \mathsf{prox}_{\gamma g}(\boldsymbol{z}^k) = (I + \gamma\,\partial g)^{-1}(\boldsymbol{z}^k), \tag{4b}$$

where $0 < \gamma \leqslant 1/L$ is the step size and

$$\mathsf{prox}_g(\boldsymbol{z}) \coloneqq \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{arg\,min}} \left\{ \frac{1}{2}\left\|\boldsymbol{z}-\boldsymbol{x}\right\|_2^2 + g(\boldsymbol{x}) \right\} \tag{5}$$

is the proximal operator, which is well-defined for any proper, closed, and convex function g.

2.2. Using DL Denoisers as Priors

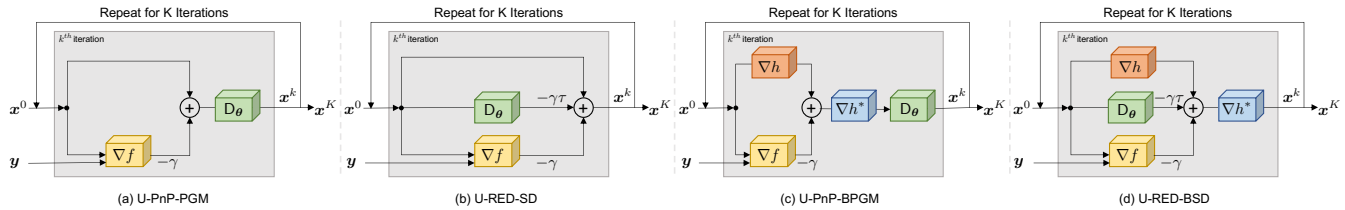
The mathematical equivalence between proximal operator (5) and Gaussian denoiser motivated denoiser-based iterative algorithms such as PnP [3]. In PnP, the proximal operator is replaced with an arbitrary Gaussian denoiser D. Unlike PnP where the explicit regularizer g is usually not known for a given denoiser, RED [4] seeks to form an explicit denoiser-based regularizer $g(\boldsymbol{x}) = \tau\,\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{x} - \mathsf{D}(\boldsymbol{x}))$, where $\tau > 0$ is the regularization parameter. When the denoiser is locally homogeneous and has a

Algorithm 1 PnP-BPGM

input: $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, and $\gamma > 0$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{x}^{k+1} = \mathsf{D}_{\boldsymbol{\theta}} \left(\nabla h^* \left(\nabla h - \gamma \nabla f \right) \right) (\boldsymbol{x}^k)$

Algorithm 2 RED-BSD

input: $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, and $\gamma > 0$
for $k = 1, 2, \ldots$ **do**
  $\boldsymbol{x}^{k+1} = \nabla h^* \left(\nabla h - \gamma \left(\nabla f + \tau \left(I - \mathsf{D}_{\boldsymbol{\theta}} \right) \right) \right) (\boldsymbol{x}^k)$



(a) U-PnP-PGM (b) U-RED-SD (c) U-PnP-BPGM (d) U-RED-BSD

Fig. 1: The proposed PnP-BPGM and RED-BSD methods replace the quadratic penalty in PnP-PGM and RED-SD by a more general Bregman distance. Both algorithms rely on data-driven regularizers obtained by training an artifact-removal operator $\mathsf{D}_{\boldsymbol{\theta}}$ via deep unfolding.

symmetric Jacobian [17], the gradient of the RED regularizer has a simple form $\nabla g(\boldsymbol{x}) = \tau \left(\boldsymbol{x} - \mathsf{D}(\boldsymbol{x}) \right)$, which enables the usage of the traditional *steepest descent (SD)* method for solving (2). By leveraging the power of state-of-the-art DL-denoisers, such as DnCNN [18], PnP/RED have achieved empirical success in many imaging applications [19, 20].

2.3. The Bregman Distance

Given a differentiable convex reference function h defined on a closed convex set $C \subset \mathbb{R}^n$, the Bregman distance $B_h : \operatorname{dom} h \times \operatorname{int} \operatorname{dom} h \to [0, \infty)$ [21] is defined by

$$B_h(\boldsymbol{x}; \boldsymbol{y}) \coloneqq h(\boldsymbol{x}) - h(\boldsymbol{y}) - \nabla h(\boldsymbol{y})^{\mathsf{T}} (\boldsymbol{x} - \boldsymbol{y}). \tag{6}$$

The Bregman distance[1] is an extension of the classical squared Euclidean distance which is recovered when $h(\boldsymbol{x}) = 1/2 \|\boldsymbol{x}\|_2^2$. Other widely-used Bregman distance functions include the KL-divergence and the Itakura–Saito (IS) distance.

3. PROPOSED METHOD

The path to Bregman-based proximal algorithm starts from observing that the quadratic majorization step in the classical PGM in eq. (3) is equivalent to the following condition:

$$\frac{L}{2} \|\boldsymbol{x}\|^2 - f(\boldsymbol{x}) \quad \text{is convex}, \tag{7}$$

where the equivalence follows from the first-order convexity inequality. To bypass the Lipschitz gradient assumption, the work in [14, 15] has proposed to generalize the condition in eq. (7) by using a possibly non-quadratic reference function h

$$L\, h(\boldsymbol{x}) - f(\boldsymbol{x}) \quad \text{is convex}. \tag{8}$$

Such functions f can be referred to as L-smooth relative to h. Then, by using the first-order convexity inequality, one can obtain a Bregman majorizer of the data-fidelity term

$$f(\boldsymbol{x}) \leqslant f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k)^{\mathsf{T}} (\boldsymbol{x} - \boldsymbol{x}^k) + L\, B_h(\boldsymbol{x}, \boldsymbol{x}^k). \tag{9}$$

[1]Note that the Bregman distance is a pesudodistance, because it does not satisfy the triangle inequality, and is generally asymmetric.

This inequality directly leads to the Bregman PGM (BPGM) method, which generalizes the classical PGM using a Bregman majorizer as

$$\boldsymbol{x}^{k+1} = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\operatorname{arg\,min}} \left\{ \boldsymbol{x}^{\mathsf{T}} \nabla f(\boldsymbol{x}^k) + L\, B_h(\boldsymbol{x}, \boldsymbol{x}^k) + g(\boldsymbol{x}) \right\}. \tag{10}$$

The BPGM method shares the same structural splitting mechanism as the classical PGM [16], which allows one to express (10) as

$$\boldsymbol{z}^k = \nabla h^* \left(\nabla h - \gamma \nabla f \right) (\boldsymbol{x}^k) \tag{11a}$$

$$\boldsymbol{x}^{k+1} = \left(\nabla h + \gamma\, \partial g \right)^{-1} \nabla h(\boldsymbol{z}^k) \tag{11b}$$

where $0 < \gamma \leqslant 1/L$ is the step size and h^* denotes the Fenchel conjugate of h. Note that the PGM is a special case of the BPGM obtained by setting $h(\boldsymbol{x}) = 1/2 \|\boldsymbol{x}\|_2^2$. The first step of the BPGM in (11a) is known as the Mirror Descent (MD) algorithm, which generalizes the classical GM. The second step is known as the *left* Bregman proximal operator (BPO) defined as

$$\operatorname{prox}_g^h(\boldsymbol{z}) \coloneqq \underset{\boldsymbol{x} \in \mathbb{R}^n}{\operatorname{arg\,min}} \left\{ B_h(\boldsymbol{x}, \boldsymbol{z}) + g(\boldsymbol{x}) \right\}. \tag{12}$$

Traditionally, the BPO is motivated from a computational perspective, e.g., Bregman projection onto the simplex with $h(\boldsymbol{x}) = \boldsymbol{x}^{\mathsf{T}} \log(\boldsymbol{x})$ is simpler than the corresponding classical proximal operator (5). Similar to the case where the classical proximal operator can be interpreted as a Gaussian denoiser, under some conditions on the reference function h, the BPO can be interpreted as an exponential family mean estimator. Moreover, selecting the reference function h provides more flexibility depending on the problem settings [14–16].

3.1. Bregman PnP and RED Algorithms

In this section, we propose two algorithms, PnP-BPGM and RED-BSD that extend existing two algorithms PnP-PGM and RED-SD, respectively. PnP-BPGM is obtained by replacing the BPO in (11b) with a DL network $\mathsf{D}_{\boldsymbol{\theta}}$

$$\boldsymbol{z}^k = \nabla h^* \left(\nabla h - \gamma \nabla f \right) (\boldsymbol{x}^k) \tag{13a}$$

$$\boldsymbol{x}^{k+1} = \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{z}^k), \tag{13b}$$

where $\boldsymbol{\theta}$ are the learnable parameters that characterize the network $\mathsf{D}_{\boldsymbol{\theta}}$. Similarly, the Bregman variant of RED-SD is obtained as

$$\boldsymbol{x}^{k+1} = \nabla h^* \left(\nabla h - \gamma \left(\nabla f + \tau \left(I - \mathsf{D}_{\boldsymbol{\theta}} \right) \right) \right) (\boldsymbol{x}^k), \tag{14}$$

Table 1: The PSNR (dB) results of different methods on the testing images with different peaks and kernels.

Method	1	2	3	4	5	6	7	8	9	10	11	12	Average
Uniform kernel, peak = 8													
Corrupted	11.70	11.13	11.74	11.59	11.61	9.56	11.81	11.80	11.82	11.63	12.04	11.91	11.53
U-Net	20.89	23.11	**21.28**	20.79	19.79	19.15	19.63	24.76	21.96	**22.70**	23.37	22.65	21.67
U-PnP-PGM	19.57	22.74	20.89	20.59	19.20	19.15	19.04	24.38	21.79	22.43	23.19	22.44	21.28
U-RED-SD	20.38	22.74	20.74	20.29	18.81	19.00	18.90	24.44	21.81	22.38	23.32	22.41	21.27
U-PnP-BPGM	**21.00**	**24.12**	21.27	20.72	20.10	19.17	***19.81***	**25.21**	**22.13**	22.65	**23.82**	22.62	**21.89**
U-RED-BSD	20.97	23.97	21.14	**20.82**	**20.25**	**19.28**	19.78	25.08	**22.13**	**22.70**	23.75	**22.66**	21.88
Uniform kernel, peak = 32													
Corrupted	16.14	16.08	16.51	16.25	15.76	13.97	15.81	17.12	16.68	16.77	17.13	16.95	16.26
U-Net	21.50	24.66	22.12	21.71	21.01	19.97	20.23	26.19	22.56	23.38	24.67	23.40	22.62
U-PnP-PGM	20.90	23.76	22.03	21.54	20.54	19.71	19.85	25.37	22.22	23.26	24.01	23.32	22.21
U-RED-SD	21.37	24.13	21.92	21.41	20.75	19.88	20.17	25.67	22.40	23.35	24.25	23.40	22.39
U-PnP-BPGM	**21.58**	25.01	22.15	**21.81**	**21.64**	**20.23**	**20.57**	26.33	22.64	**23.45**	24.90	**23.46**	**22.81**
U-RED-BSD	21.57	**25.04**	**22.17**	21.64	21.48	20.22	20.34	**26.44**	**22.65**	23.35	**24.91**	23.41	22.77
Gaussian kernel, peak = 8													
Corrupted	11.98	11.25	12.01	11.86	12.01	9.71	12.32	11.89	11.89	11.79	12.17	12.07	11.75
U-Net	21.72	24.92	22.06	21.55	22.17	20.87	21.27	25.60	22.22	22.97	24.63	23.08	22.76
U-PnP-PGM	21.01	23.97	21.70	21.41	20.74	19.72	20.32	25.18	22.17	22.72	24.17	22.87	22.16
U-RED-SD	21.18	23.30	21.88	21.26	20.65	19.79	20.15	24.86	21.96	22.97	23.69	23.03	22.06
U-PnP-BPGM	**22.30**	24.60	**22.48**	**21.78**	**22.44**	19.23	**21.92**	**26.03**	**22.48**	**23.90**	**24.49**	**23.60**	**22.94**
U-RED-BSD	22.22	**24.62**	22.17	21.72	22.27	**19.61**	21.61	25.76	22.37	23.79	24.37	**23.60**	22.84
Gaussian kernel, peak = 32													
Corrupted	17.06	16.62	17.30	17.17	17.05	14.45	17.19	17.51	17.04	17.35	17.57	17.55	16.99
U-Net	22.63	26.74	23.13	23.13	23.83	**21.69**	22.51	27.14	22.89	24.00	25.95	24.12	23.98
U-PnP-PGM	22.12	24.50	23.61	23.10	22.54	19.53	21.81	26.03	22.60	24.27	24.77	24.22	23.26
U-RED-SD	22.15	25.43	23.07	23.14	22.86	21.29	21.92	26.55	22.80	24.27	25.31	24.24	23.58
U-PnP-BPGM	**23.41**	**26.85**	**23.79**	**23.54**	24.41	20.82	**23.30**	27.86	**23.13**	**25.03**	26.03	**24.83**	**24.42**
U-RED-BSD	23.12	26.79	23.41	23.27	**24.46**	21.02	23.05	**27.88**	23.11	24.96	**26.04**	24.75	24.32

where I is an identity operator. When the assumptions for the existence of the explicit RED regularizer in [4] hold, then RED-BSD can be interpreted as the mirror descent algorithm. Note that the PnP-PGM [3] and RED-SD [4] are recovered when $h(\boldsymbol{x}) = 1/2\|\boldsymbol{x}\|^2$ and $\mathsf{D}_{\boldsymbol{\theta}}$ being a Gaussian denoiser.

Algorithm 1 and Algorithm 2 summarize the proposed PnP-BPGM and RED-BSD algorithms. In this work, the regularizer $\mathsf{D}_{\boldsymbol{\theta}}$ is implemented using the deep unfolded strategy, so we refer to the proposed algorithms as unfolded PnP-BPGM (U-PnP-BPGM) and unfolded RED-BSD (U-RED-BSD). Similarly, the unfolded version of PnP-PGM, and RED-SD are referred as U-PnP-PGM and U-RED-SD. All four different unfolding architectures are shown in Fig. 1 and will be compared in the next section.

Recent work has explored the convergence properties of various PnP/RED algorithms [10, 11, 22]. Similar results can be also established for both PnP-BPGM and RED-BSD. The following theorem presents the analysis of PnP-BPGM for a strongly convex function f and a Lipschitz continuous operator $\mathsf{D}_{\boldsymbol{\theta}}$. While these assumptions are too strong for some applications, they provide the first steps for the broader analysis of Bregman PnP/RED methods.

Theorem 1 *Assume h be μ_h-strongly convex with L_h-Lipschitz continuous gradient, and f be μ_f-strongly convex function with L_f-Lipschitz continuous gradient. Assume $\mathsf{D}_{\boldsymbol{\theta}}$ be an M-Lipschitz operator. Then, the iteration in* (13) *converges to a fixed point if*

$$M < \frac{\mu_h\,(\mu_f + L_f)}{L_h\,L_f - \mu_h\,\mu_f} \tag{15}$$

and the step size $\frac{\mu_h}{\mu_f}\left(\frac{L_h}{\mu_h} - \frac{1}{M}\right) < \gamma < \frac{\mu_h}{L_f}\left(1 + \frac{1}{M}\right)$.

4. NUMERICAL ILLUSTRATION

4.1. Poisson Linear Inverse Problem

We empirically evaluated the proposed methods on Poisson linear inverse problems. Poisson noise is a signal dependent noise whose negative log-likelihood function results in the following data-fidelity term and its gradient

$$f(\boldsymbol{x}) = \mathbf{1}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}^{\mathsf{T}}\log\left(\boldsymbol{A}\boldsymbol{x}\right) + \mathbf{1}^{\mathsf{T}}\log(\boldsymbol{y}!) \tag{16a}$$

$$\nabla f(\boldsymbol{x}) = \boldsymbol{A}^{\mathsf{T}}\left(\mathbf{1} - \boldsymbol{y} \oslash \left(\boldsymbol{A}\boldsymbol{x}\right)\right) \tag{16b}$$

where $\mathbf{1}$ is a vector of ones, and $\oslash$ denotes element-wise division.

Classical algorithms for solving Poisson linear inverse problems include the Richardson–Lucy (RL) algorithm and transform-based methods [23–28]. Several ADMM-based algorithms were proposed that handle the data fidelity via its proximal operator [29, 30]. In [14] it was showed that by using the Burg's entropy as a reference function $h(\boldsymbol{x}) = -\mathbf{1}^{\mathsf{T}}\log(\boldsymbol{x})$, one can satisfy (8) with $L \geqslant \|\boldsymbol{y}\|_1$. Therefore, using (13) we can obtain the following simple iteration for PnP-BPGM

$$\boldsymbol{z}^k = \frac{\boldsymbol{x}^k}{\mathbf{1} + \gamma\,\boldsymbol{x}^k \odot \nabla f(\boldsymbol{x}^k)} \tag{17a}$$

$$\boldsymbol{x}^{k+1} = \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{z}^k), \tag{17b}$$

where $\odot$ is the element-wise multiplication. It can be shown that the backward operator is related to inverse Gamma scale estimator.

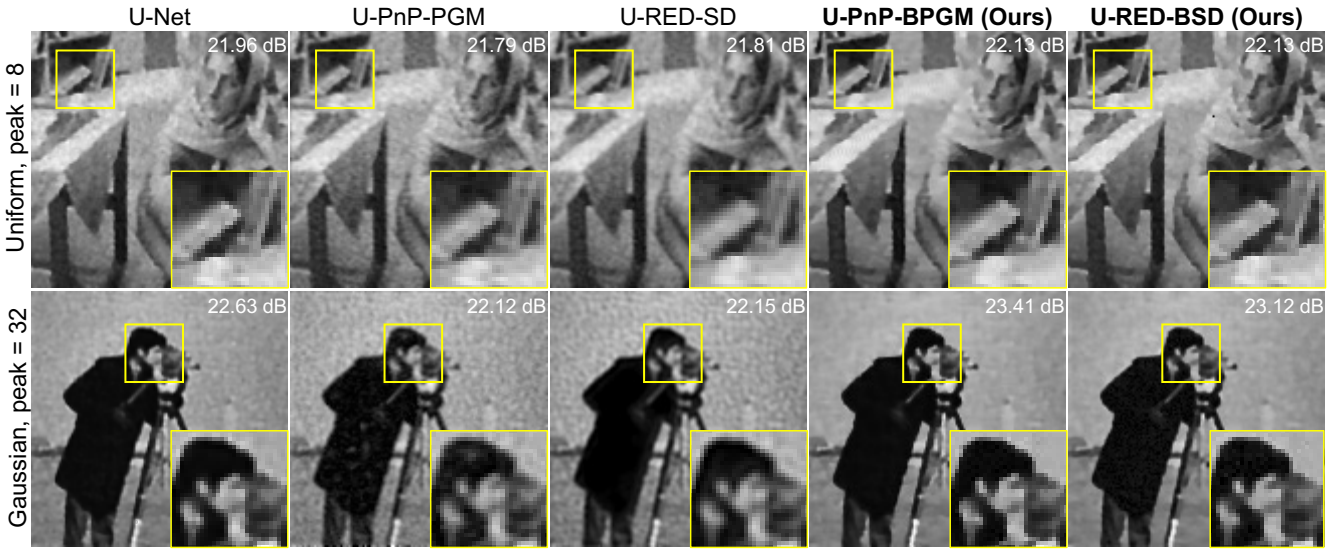


Fig. 2: Examples of image reconstruction results on *Babara* (top) and *Cameraman* (bottom) obtained by U-Net, U-PnP-PGM, U-RED-SD, U-PnP-BPGM, and U-RED-BSD. The first row is corresponding to the noise peak 8 with uniform kernel, and the second row is noiser peak 32 with Gaussian kernel. Each reconstruction is labeled with its PSNR (dB) value with respect to the Ground-truth image. Visual differences are highlighted using the rectangles drawn inside the images. Note U-PnP-BPGM and U-RED-BSD shows close performance one to another, outperforming other methods and providing the best visual results by recovering sharp edges and removing artifacts.

Similarly, RED-BSD in (14) can be simplified to

$$\boldsymbol{x}^{k+1} = \frac{\boldsymbol{x}^k}{\mathbf{1} + \gamma\, \boldsymbol{x}^k \odot (\nabla f + \tau\,(I - \mathsf{D}_{\boldsymbol{\theta}}))(\boldsymbol{x}^k)}. \tag{18}$$

4.2. Image Deblurring with Poisson Noise

We demonstrate the ability of our proposed algorithms PnP-BPGM and RED-BSD over their traditionally counterparts PGM and RED on Poisson linear inverse problems. We focus on image deblurring, where the forward model $\boldsymbol{A}$ corresponds to the blurring operator. Specifically, we follow a similar settings in [29, 30], and test our algorithms for Poisson noise with peaks 8 and 32 using two different blur kernels of size 9 by 9: (1) a Gaussian kernel with $\sigma = 1.6$, and (2) a uniform kernel, respectively. All the methods compared are trained in an unfolding fashion as illustrated in Fig. 1, where the end-to-end training seeks to compute the trainable parameters in $\mathsf{D}_{\boldsymbol{\theta}}$ by minimizing the ℓ_2 loss function between network output $\{\boldsymbol{x}^K\}$ and the ground-truth $\{\boldsymbol{x}\}$ over all training samples. We set $\boldsymbol{x}^0$ using the raw measurements $\boldsymbol{y}$ with a small white Gaussian perturbation. We unfold each algorithm with $K = 100$ iterations for stable performance, where in each iteration, the network $\mathsf{D}_{\boldsymbol{\theta}}$ is realized using a $7-$layer DnCNN [18] with shared weights across all iterations. The step-size parameter γ and the regularization parameter τ in RED and BRED are set as a learnable parameters, initialized with $\gamma = 5 \times 10^{-1}$ and $\tau = 1 \times 10^{-3}$. As a reference, we also report the image reconstruction performance of the end-to-end learning method where U-Net [31] is trained end-to-end in the usual supervised fashion using the ℓ_2-loss. All networks are trained on public dataset BSD400 for 400 epochs, using the Adam solver [32] with an initial learning rate 1×10^{-4}. We select the models that achieved the best performance on the validation dataset BSD68. At test time, Set12 dataset is used to evaluate the performance of each algorithm.

The numerical results on the test dataset Set12 with respect to two scenarios are summarized in Table 1. Test images used for the quantitative performance labeled from 1 to 12 are: *Cameraman*, *House*, *Pepper*, *Starfish*, *Butterfly*, *Plane*, *Parrot*, *Lena*, *Barbara*, *Boat*, *Artist*, *Room*. For each image, the highest PSNR in each scenario is highlighted. We observe that the performances of U-PnP-BPGM and U-RED-BSD are very close to one another, providing the best performance compared to all the other methods, outperforming U-PnP-PGM and U-RED-SD. Fig. 2 shows visual examples for two images from Set12 in two different settings, uniform kernel with peak 8 (top) and Gaussian kernel with peak 32 (bottom). Note that both U-PnP-PGM and U-RED-SD yield similar visual recovery performance with artifacts remaining in the images, U-PnP-BPGM and U-RED-BSD show much better reconstruction performance in removing artifacts and noise. The enlarged regions in the image suggest that U-PnP-BPGM and U-RED-BSD better recover the fine details and sharper edges compared to their counterparts and U-Net.

5. CONCLUSION

This paper proposes generalizing plug-and-play priors (PnP) and regularization by denoising (RED) beyond squared Euclidean distance using the Bregman distance. The proposed Bregman-based methods are motivated by the recent progress in optimization, that have the potential to better align to specific non-Euclidean geometry of the loss function. Our numerical results show the potential of the proposed methods in Poisson linear inverse problems. This work can be considered as a first step towards extending widely-used PnP/RED to problems where there is a benefit of using non-Euclidean formulations of proximal and projection operators.

6. REFERENCES

[1] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *IEEE J. on Sel. Areas in Inf. Theory*, vol. 1, no. 1, pp. 39–56, 2020.

[2] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image process." *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, 2021.

[3] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conf. on Signal and Inf. Process.*, 2013, pp. 945–948.

[4] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. on Imaging Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.

[5] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. of the 27th Int. Conf. on Machine Learning*, June 2010, pp. 399–406.

[6] U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1872–1876, 2017.

[7] S. Ono, "Primal-dual plug-and-play image restoration," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1108–1112, 2017.

[8] A. H. Al-Shabili, H. Mansour, and P. T. Boufounos, "Learning plug-and-play proximal quasi-newton denoisers," in *Int. Conf. on Acoust., Speech and Signal Process.*, 2020, pp. 8896–8900.

[9] Y. Sun, Z. Wu, X. Xu, B. Wohlberg, and U. S. Kamilov, "Scalable Plug-and-Play ADMM With Convergence Guarantees," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 849–863, 2021.

[10] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds," *Proc. 32nd Int. Conf. Inf. Process. Syst.*, pp. 9079—-9089, 2018.

[11] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *Int. Conf. on Machine Learning*. PMLR, 2019, pp. 5546–5557.

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

[13] X. Xu, J. Liu, Y. Sun, B. Wohlberg, and U. S. Kamilov, "Boosting the performance of plug-and-play priors via denoiser scaling," in *54th Asilomar Conf. on Signals, Systems, and Computers*, 2020, pp. 1305–1312.

[14] H. H. Bauschke, J. Bolte, and M. Teboulle, "A descent lemma beyond Lipschitz gradient continuity: First-order methods revisited and applications," *Math. of Operations Res.*, vol. 42, no. 2, pp. 330–348, 2017.

[15] H. Lu, R. M. Freund, and Y. Nesterov, "Relatively smooth convex optimization by first-order methods, and applications," *SIAM J. on Optim.*, vol. 28, no. 1, pp. 333–354, 2018.

[16] M. Teboulle, "A simplified view of first order methods for optim." *Math. Program.*, vol. 170, no. 1, pp. 67–96, 2018.

[17] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Trans. Comput. Imag.*, vol. 5, no. 1, pp. 52–67, 2018.

[18] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[19] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 408–423, 2016.

[20] A. Brifman, Y. Romano, and M. Elad, "Turning a denoiser into a super-resolver using plug and play priors," in *2016 IEEE Int. Conf. on Image Process. (ICIP)*, 2016, pp. 1404–1408.

[21] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. and Math. physics*, vol. 7, no. 3, pp. 200–217, 1967.

[22] Y. Sun, B. Wohlberg, and U. S. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *IEEE Trans. Comput. Imag.*, vol. 5, no. 3, pp. 395–408, 2019.

[23] Z. T. Harmany, R. F. Marcia, and R. M. Willett, "This is SPIRAL-TAP: Sparse Poisson intensity reconstruction algorithms—theory and practice," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1084–1096, 2011.

[24] P. Sarder and A. Nehorai, "Deconvolution methods for 3-D fluorescence microscopy images," *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 32–45, 2006.

[25] J.-L. Starck and F. Murtagh, *Astronomical Image and Data Analysis*. Berlin, Germany: Springer-Verlag, 2007.

[26] N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, "Richardson–Lucy algorithm with total variation regularization for 3D confocal microscope deconvolution," *Microsc. Res. and Technique*, vol. 69, no. 4, pp. 260–266, 2006.

[27] M. Makitalo and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising," *IEEE Trans. Image Process*, vol. 20, no. 1, pp. 99–109, 2010.

[28] F.-X. Dupé, J. M. Fadili, and J.-L. Starck, "A proximal iteration for deconvolving Poisson noisy images using sparse representations," *IEEE Trans. Image Process*, vol. 18, no. 2, pp. 310–321, 2009.

[29] M. A. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process*, vol. 19, no. 12, pp. 3133–3145, 2010.

[30] A. Rond, R. Giryes, and M. Elad, "Poisson inverse problems by the plug-and-play scheme," *J. of Vis. Communication and Image Representation*, vol. 41, pp. 96–108, 2016.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Med. Image Comput. and Computer-Assisted Intervention (MICCAI)*, Munich, Germany, Oct. 2015, pp. 234–241.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–13.