Training Methods for Regularizing Gradients on Multi-Task Image Restoration Problems*

Samuel Willingham[1,2][a], Mårten Sjöström[2][b] and Christine Guillemot[1][c]
[1]*Inria Rennes, Rennes, France*
[2]*Mid Sweden University, Sundsvall, Sweden*

Keywords: Inverse Problems, Computer Vision, Image Restoration, Deep Equilibrium Models, Deep Priors.

Abstract: Inverse problems refer to the task of reconstructing a clean signal from a degraded observation. In imaging, this pertains to restoration problems like denoising, super-resolution or in-painting. Because inverse problems are often ill-posed, regularization based on prior information is needed. Plug-and-play (pnp) approaches take a general approach to regularization and plug a deep denoiser into an iterative solver for inverse problems. However, considering the inverse problems at hand in training could improve reconstruction performance at test-time. Deep equilibrium models allow for the training of multi-task priors on the reconstruction error via an estimate of the iterative method's fixed-point (FP). This paper investigates the intersection of pnp and DEQ models for the training of a regularizing gradient (RG) and derives an upper bound for the reconstruction loss of a gradient-descent (GD) procedure. Based on this upper bound, two procedures for the training of RGs are proposed and compared: One optimizes the upper bound directly, the other trains a deep equilibrium GD (DEQGD) procedure and uses the bound for regularization. The resulting regularized RG (RERG) produces consistently good reconstructions across different inverse problems, while the other RGs tend to have some inverse problems on which they provide inferior reconstructions.

1 INTRODUCTION

This paper considers image enhancement and restoration through the lens of inverse problems. Inverse problems refer to a large subset of signal processing applications, where one attempts to recover a signal from a flawed or noisy observation. In imaging, this refers to the reconstruction of degraded images - e.g. images with missing pixels, or low-resolution images. Because inverse problems can be ill-posed, some kind of regularization that imposes prior assumptions on the search-space is necessary.

One approach to solving the regularization problem for inverse problems are the pnp methods (Venkatakrishnan et al., 2013; Pesquet et al., 2021; Chan et al., 2016; Le Pendu and Guillemot, 2023; Zhang et al., 2021). These methods use pre-trained priors in iterative algorithms like the alternating direction method of multipliers (ADMM) (Chan et al., 2016), the forward-backward algorithm (FB) (Pesquet et al., 2021), or GD. ADMM and FB use a regularizing proximal operator (PO) that represents the prior. This PO can be replaced with a deep Gaussian denoiser (Venkatakrishnan et al., 2013; Chan et al., 2016), leading to good reconstruction performance. Apart from methods based on POs, there are other approaches like regularization by denoising (Romano et al., 2017), or the pnp Regularizing Gradient (pnpReG) (Fermanian et al., 2023), which trains a RG that is used to regularize a pnp GD procedure. Overall, pnp approaches are very general in application, but because they are not trained on the inverse problems at hand, performance can be further improved (Willingham et al., 2023).

To train a full iterative scheme directly on the inverse problem at hand, deep equilibrium (DEQ) models (Gilton et al., 2021; Bai et al., 2019; Fung et al., 2022; Winston and Kolter, 2020; Ling et al., 2022) can be used to train an entire iterative method via its FP. This can be done in a Jacobian-free manner (Fung et al., 2022), which is effectively one step of back-

[a] https://orcid.org/0009-0005-1954-1143
[b] https://orcid.org/0000-0003-3751-6089
[c] https://orcid.org/0000-0003-1604-967X

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956770.

Willingham, S., Sjöström, M. and Guillemot, C.
Training Methods for Regularizing Gradients on Multi-Task Image Restoration Problems.
DOI: 10.5220/0012368100003660
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 3: VISAPP*, pages 145-153
ISBN: 978-989-758-679-8; ISSN: 2184-4321

propagation on the iterative method, starting at the FP. When applied to a range of inverse problems, multi-task (MT) DEQ models (Willingham et al., 2023) allow for the training of MT priors on a large range of inverse problems by leveraging the reconstruction error over a range of inverse problems, leading to strong MT reconstruction performance.

However, DEQ models heavily depend on finding a good approximation of the FP. This can be quite difficult, as convergence is not always guaranteed and even if the algorithm does converge, this can take a large amount of iterations or lead to bad estimates of the FP. As a result, this can lead to sub-optimal parameter updates.

The contributions of this paper are:

- This paper investigates and compares different approaches of training a RG for a range of inverse problems.
- We show that the difference between the GD and the FB reconstruction errors is bounded from above. This upper bound led us to propose a method for training the RG using a FB optimization method. The resulting RG is called RG1.
- We also propose a procedure that uses the upper bound to regularize the training of a multi-task DEQGD procedure. This regularized RG (RERG) leads to strong reconstruction performance on a range of inverse problems by taking into account the FB and the GD reconstruction-errors. In testing, RERG displayed performance close to whichever was better (RG1 or DEQGD) on any given inverse problem.
- We compare four different RGs on a range of inverse problems and discuss the differences.

2 RELATED WORKS AND THEORY

2.1 Inverse Problems and Maximum a Posteriori Estimation

We consider the image formation model

$$\boldsymbol{y} = A\hat{\boldsymbol{x}} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{y}$ denotes the degraded observation, $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ is the ground truth image, $A : \mathbb{R}^d \to \mathbb{R}^{d'}$ denotes the degradation operation and $\boldsymbol{\varepsilon} \in \mathbb{R}^{d'}$ denotes additive white Gaussian noise (AWGN) with standard deviation $\sigma \geq 0$, and $d, d' \in \mathbb{N}$.

In this paper, we consider maximum a posteriori (MAP) estimation (Venkatakrishnan et al., 2013; Zhang et al., 2021; Le Pendu and Guillemot, 2023; Fermanian et al., 2023), meaning the search for an $\boldsymbol{x} \in \mathbb{R}^d$, that maximizes $p(\boldsymbol{x} \mid \boldsymbol{y})$; i.e. the $\boldsymbol{x}$ that is the most likely to have caused observation $\boldsymbol{y}$. Assuming uniqueness, this leads to

$$\hat{\boldsymbol{x}}_{\text{MAP}} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, p(\boldsymbol{x} \mid \boldsymbol{y}) \tag{2}$$

$$= \underset{\boldsymbol{x}}{\operatorname{argmin}} - \log p(\boldsymbol{y} \mid \boldsymbol{x}) - \log p(\boldsymbol{x}) \tag{3}$$

$$= \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sigma^2 R(\boldsymbol{x}), \tag{4}$$

where we set $R(\boldsymbol{x}) := -\log p(\boldsymbol{x})$ and $\|\cdot\|_2$ denotes the 2-norm. The expression with the norm is called the data-term. R is called the regularizer and represents the prior distribution.

The MAP estimation problem can be solved using iterative algorithms, like the ADMM, FB or GD algorithm.

2.2 Plug-and-Play Regularizing Gradient

PnpReG (Fermanian et al., 2023) deals with the regularization problem in equation (4) by linking the gradient of the regularizer with its proximal operator

$$\operatorname{prox}_{\sigma^2 R}(\boldsymbol{z}) := \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \sigma^2 R(\boldsymbol{x}). \tag{5}$$

As shown in (Fermanian et al., 2023), it holds that for all $\boldsymbol{z} \in \mathbb{R}^d$ and $\sigma \geq 0$

$$\sigma^2 \frac{\delta R(\boldsymbol{x})}{\delta \boldsymbol{x}}\bigg|_{\boldsymbol{x}=\operatorname{prox}_{\sigma^2 R}(z)} = \boldsymbol{z} - \operatorname{prox}_{\sigma^2 R}(\boldsymbol{z}). \tag{6}$$

Thus, the loss

$$L_\sigma^{\text{link}}(\boldsymbol{z}) = \|\sigma^2 G(P_{\sigma^2}(\boldsymbol{z})) - (\boldsymbol{z} - P_{\sigma^2}(\boldsymbol{z}))\|_2^2 \tag{7}$$

is used to train the gradient G of the regularizer corresponding to the PO defined by a Gaussian denoiser. This uses the approximation $P_{\sigma^2}(\boldsymbol{x}) \approx \operatorname{prox}_{\sigma^2 R}(\boldsymbol{z})$, where $P_{\sigma^2}(\boldsymbol{x})$ is trained as a deep-denoiser. The overall training loss is

$$L_{\text{pnpReG}} = \delta\|P_{\sigma^2}(\tilde{\boldsymbol{z}}) - \hat{\boldsymbol{x}}\|_1 + \lambda L_\sigma^{\text{link}}(\tilde{\boldsymbol{z}}), \tag{8}$$

where $\|\cdot\|_1$ is the L1 norm, $\lambda > 0$, $\tilde{\boldsymbol{z}} = \hat{\boldsymbol{x}} + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon}$ being AWGN with standard deviation $\sigma_0 > 0$. Furthermore, δ is equal to 1 if $\sigma = \sigma_0$ and δ is equal to zero, otherwise. This leads to a RG that is trained jointly with a denoiser and produces strong reconstruction results when used as regularization in a GD algorithm. Pnp approaches, however, do not take the reconstruction error into account and could potentially be improved by considering the inverse problems at hand.

2.3 Deep Equilibrium Models

DEQ models (Gilton et al., 2021; Bai et al., 2019; Fung et al., 2022; Winston and Kolter, 2020; Ling et al., 2022) allow for the training of a whole iterative procedure h_θ via the corresponding FP $\boldsymbol{x}_h$ and the resulting reconstruction error. In (Bai et al., 2019), this is done by considering $h_\theta(\boldsymbol{x}_h) = \boldsymbol{x}_h$ and thus

$$\frac{\delta \boldsymbol{x}_h}{\delta\theta} = \frac{\delta h_\theta(\boldsymbol{x})}{\delta \boldsymbol{x}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}_h} \frac{\delta \boldsymbol{x}_h}{\delta\theta} + \frac{\delta h_\theta(\boldsymbol{x})}{\delta\theta}\bigg|_{\boldsymbol{x}=\boldsymbol{x}_h}. \tag{9}$$

Rearranging this and plugging $\delta\boldsymbol{x}_h/\delta\theta$ into the derivative of loss $l : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with respect to θ, yields

$$\frac{\delta l(\hat{\boldsymbol{x}}, \boldsymbol{x}_h)}{\delta\theta} = \frac{\delta l(\hat{\boldsymbol{x}}, \boldsymbol{x})}{\delta \boldsymbol{x}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}_h} \frac{\delta \boldsymbol{x}_h}{\delta\theta} \tag{10}$$

$$= \frac{\delta l(\hat{\boldsymbol{x}}, \boldsymbol{x})}{\delta \boldsymbol{x}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}_h} J^{-1} \frac{\delta h_\theta(\boldsymbol{x})}{\delta\theta}\bigg|_{\boldsymbol{x}=\boldsymbol{x}_h}, \tag{11}$$

where the Jacobian $J := \mathrm{id} - \frac{\delta h_\theta(\boldsymbol{x})}{\delta \boldsymbol{x}}\Big|_{\boldsymbol{x}=\boldsymbol{x}_h}$ exists and is invertible (Fung et al., 2022). Assuming the inverse of J to be equal to the identity still yields a direction of descent (Fung et al., 2022). This is called Jacobian-free back-propagation and can be likened to a single step of traditional back-propagation, using the FP $\boldsymbol{x}_h$ as input.

The following section will introduce important definitions that we shall use to expand existing theory and propose different training methods for RGs.

3 DEFINITIONS

As it is our intent to train and compare RGs to be used in a GD algorithm, we shall first define the necessary functions and expressions.

We define the two iterative algorithms to solve the MAP estimation problem in 4. One is the FB algorithm (Pesquet et al., 2021) that was also used for the multi-task DEQ (MTDEQ) prior (Willingham et al., 2023), where for each ground truth image $\hat{\boldsymbol{x}}$, we pick a degradation $A : \mathbb{R}^d \to \mathbb{R}^{d'}$ and noise-level σ at random, generating noise $\boldsymbol{\varepsilon}$, leading to a degraded observation $\boldsymbol{y} = A(\hat{\boldsymbol{x}}) + \boldsymbol{\varepsilon}$. For $\boldsymbol{z} \in \mathbb{R}^d$. One iteration of the FB algorithm takes the form

$$f_{\sigma,A,\theta}(\boldsymbol{z}, \boldsymbol{y}) := P_{\theta,\eta\sigma^2}\left(\boldsymbol{z} - \eta D(\boldsymbol{z}, \boldsymbol{y})\right), \tag{12}$$

where P_{θ,σ^2} is a regularizer in form of a DRUNet (as in (Le Pendu and Guillemot, 2023) and (Zhang et al., 2021)) with parameters θ used on noise-level $\sigma \geq 0$ with step-size $\eta \in \mathbb{R}$. For $\boldsymbol{z} \in \mathbb{R}^d$, we define

$$D(\boldsymbol{z}, \boldsymbol{y}) := \frac{\delta \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|_2^2}{\delta \boldsymbol{x}}\bigg|_{\boldsymbol{x}=\boldsymbol{z}} \tag{13}$$

as the derivative of the data-term.

Similarly, for $\boldsymbol{z} \in \mathbb{R}^d$, one iteration of the gradient-descent algorithm is defined as

$$g_{\sigma,A,\theta}(\boldsymbol{z}, \boldsymbol{y}) := \boldsymbol{z} - \eta\left(D(\boldsymbol{z}, \boldsymbol{y}) + \sigma^2 G_\theta(\boldsymbol{z}, \boldsymbol{y})\right), \tag{14}$$

where G_θ is the network representing the gradient of the regularizer in (4), which we also refer to as RG. We use the same 3-channel DRU-net architecture that is used in (Fermanian et al., 2023). $\boldsymbol{x}_{g,\sigma,A,\theta}(\boldsymbol{y})$ and $\boldsymbol{x}_{f,\sigma,A,\theta}(\boldsymbol{y})$ are the FPs of $g_{\sigma,A,\theta}(\cdot, \boldsymbol{y})$ and $f_{\sigma,A,\theta}(\cdot, \boldsymbol{y})$, respectively. For clarity of notation, the indices σ, A and θ as well as the argument $\boldsymbol{y}$ will be omitted, when this is allowed by the context.

For ease of notation, we define the reconstruction errors

$$L^{\mathrm{FB}}_{\sigma,A,\theta} := \|f(\boldsymbol{x}_f) - \hat{\boldsymbol{x}}\|_2^2 \tag{15}$$

$$L^{\mathrm{GD}}_{\sigma,A,\theta} := \|g(\boldsymbol{x}_g) - \hat{\boldsymbol{x}}\|_2^2 \tag{16}$$

where L^{FB} denotes the reconstruction error of the FB algorithm and L^{GD} is the reconstruction error of the GD algorithm.

Remark 1. *Note that it is our intention to find parameters θ and thus, RG G_θ, such that L^{GD} is small. This is the entity that is evaluated when computing peak signal-to-noise ratio (PSNR) on the reconstruction.*

Definition 1 (Lipschitz continuity (Bauschke et al., 2017))**.** *We call a function $h : \mathbb{R}^n \to \mathbb{R}^n$, with $n \in N$, $\mathcal{L}$-**Lipschitz continuous** with relation to the metric induced by $\|\cdot\|_2$ and with **Lipschitz constant** $\mathcal{L} \geq 0$ if and only if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n$ it holds that*

$$\|h(\boldsymbol{x}_1) - h(\boldsymbol{x}_2)\|_2 \leq \mathcal{L}\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2. \tag{17}$$

*If there exists an $\mathcal{L} < 1$ that permits this condition, we call h a **contraction**.*

Using these definitions, the following sections will introduce an upper bound for the GD reconstruction error, propose training approaches based on said upper bound and compare the resulting reconstruction performance.

4 METHOD

4.1 Derivations

This section leverages the introduced definitions from section 3 to derive a relationship between the FB reconstruction error and the GD reconstruction error that leads to a new approach to realizing the goal in remark 1.

Theorem 1. *If GD procedure g is a contraction with relation to $\|\cdot\|_2$ (with Lipschitz constant $\mathcal{L}_g < 1$), then there exists an $\tilde{\boldsymbol{x}} \in \mathbb{R}^d$, such that*

$$\|\boldsymbol{x}_g - \hat{\boldsymbol{x}}\|_2 \le \frac{1}{1-\mathcal{L}_g}\sqrt{L^{link}_{\eta\sigma^2}(\tilde{\boldsymbol{x}})} + \|\boldsymbol{x}_f - \hat{\boldsymbol{x}}\|_2. \quad (18)$$

Proof. By plugging $\tilde{\boldsymbol{x}} = \boldsymbol{x}_f - \eta D(\boldsymbol{x}_f)$ into L^{link}, we get that for all $\eta > 0$ and $\sigma \ge 0$, it holds that

$$L^{\text{link}}_{\eta\sigma^2}(\tilde{\boldsymbol{x}}) = \|\eta\sigma^2 G(\boldsymbol{x}_f) - (\boldsymbol{x}_f - \eta D(\boldsymbol{x}_f) - \boldsymbol{x}_f)\|_2^2 \quad (19)$$

$$= \|\eta\sigma^2 G(\boldsymbol{x}_f) + \eta D(\boldsymbol{x}_f)\|_2^2. \quad (20)$$

This leads to

$$\begin{aligned}\|\boldsymbol{x}_f - \boldsymbol{x}_g\|_2 =& \|\boldsymbol{x}_f - \eta(D(\boldsymbol{x}_f) + \sigma^2 G(\boldsymbol{x}_f)) & (21)\\ &+ \eta(D(\boldsymbol{x}_f) + \sigma^2 G(\boldsymbol{x}_f)) - \boldsymbol{x}_g\|_2 & (22)\\ \le& \|g(\boldsymbol{x}_f) - \boldsymbol{x}_g\|_2 + \sqrt{L^{\text{link}}_{\eta\sigma^2}(\tilde{\boldsymbol{x}})} & (23)\\ \le& \mathcal{L}_g\|\boldsymbol{x}_f - \boldsymbol{x}_g\|_2 + \sqrt{L^{\text{link}}_{\eta\sigma^2}(\tilde{\boldsymbol{x}})}, & (24)\end{aligned}$$

resulting in the statement via the triangle inequality. □

Remark 2. *Furthermore, because for all $a, b \in \mathbb{R}$, it holds that $a^2 + b^2 \ge 2ab$, we get*

$$\frac{1}{2}\|\boldsymbol{x}_g - \hat{\boldsymbol{x}}\|_2^2 \le \frac{1}{(1-\mathcal{L}_g)^2} L^{link}_{\eta\sigma^2}(\tilde{\boldsymbol{x}}) + \|\boldsymbol{x}_f - \hat{\boldsymbol{x}}\|_2^2, \quad (25)$$

allowing us to limit the GD reconstruction error (i.e. a RG to be used in a GD algorithm) by using the two loss-terms on the right as training objectives.

Based on remark 2, we train a GD procedure by training a PO and RG to minimize the bound in (25).

Remark 3. *It immediately follows that for*

$$L^{bound} := \frac{2}{(1-\mathcal{L}_g)^2} L^{link}_{\eta\sigma^2}(\tilde{\boldsymbol{x}}) + 2\|\boldsymbol{x}_f - \hat{\boldsymbol{x}}\|_2^2, \quad (26)$$

we also get that $\|\boldsymbol{x}_g - \hat{\boldsymbol{x}}\|_2^2$ is bounded by a convex combination of itself and L^{bound}, or more generally

$$\|\boldsymbol{x}_g - \hat{\boldsymbol{x}}\|_2^2 \le \lambda L^{bound} + \zeta\|\boldsymbol{x}_g - \hat{\boldsymbol{x}}\|_2^2, \quad (27)$$

for all $\lambda, \zeta > 0$ and with $\lambda + \zeta \ge 1$.

This allows us to constrain the hypothesis-space for the training of a deep equilibrium regularizing gradient, adding regularization for the DEQ training of a GD algorithm, leading to the RERG, for which the training objective will be defined in the next section.

```
Data: Ground truth image set D, $k \in \{1,2,3\}$, set
      of noise-levels $\Sigma$ and set of degradations $\mathcal{A}$;
Result: Parameters $\Theta$ for a regularizing gradient
        and the corresponding proximal operator;
for a number of epochs do
    for all $\hat{\boldsymbol{x}}$ from D do
        $(\sigma, A) \leftarrow$ random choice from $\Sigma \times \mathcal{A}$;
        $\boldsymbol{y} \leftarrow A\hat{\boldsymbol{x}} + \boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(0,\sigma)$;
        Find FPs $\boldsymbol{x}_f$ of $f_{\sigma,A,\theta}$ and $\boldsymbol{x}_g$ of $g_{\sigma,A,\theta}$ ;
        Calculate $L^k$ corresponding to the
          algorithm trained;
        Update parameters $\theta$ via loss $L^k$ ;
    end
end
```

Algorithm 1: Training algorithm for the RGs. $k = 1$ leads to RG1, $k = 2$ gives the DEQGD and $k = 3$ leads to RERG.

4.2 Algorithm

Based on the definitions in section 3 and the derivations in section 4.1, we introduce the training pipeline described in algorithm 1. This pipeline is quite similar to the one used in (Willingham et al., 2023), with only one inverse problem considered at each iteration and with different training objectives. We use this algorithm for the training of three different RGs, via the training objectives

$$L^1 := 0.1 L^{\text{link}}_{\sigma^2}(\boldsymbol{x}_f - \eta D(\boldsymbol{x}_f)) + L^{\text{FB}} \quad (28)$$

$$L^2 := L^{\text{GD}} \quad (29)$$

$$L^3 := L^1 + 0.1 L^{\text{GD}} \quad (30)$$

where 0.1 is an experimentally chosen hyper-parameter, which could likely be further optimized to improve reconstruction performance at test time. The weighting in L^3 could also be modified to increase the weight of L^{GD} or L^1, respectively.

Note that for the derivative of $L^{\text{link}}_{\sigma^2}(\boldsymbol{x}_f - \eta D(\boldsymbol{x}_f))$ used for the update of the parameters in algorithm 1, we use Jacobian-free back-propagation (Fung et al., 2022), i.e. the assumption that the approximation

$$\frac{\delta L^{\text{link}}_{\sigma^2}(\boldsymbol{x}_f - \eta D(\boldsymbol{x}_f))}{\delta\theta} \approx \left.\frac{\delta L^{\text{link}}_{\sigma^2}(\boldsymbol{x})}{\delta\theta}\right|_{\boldsymbol{x} = \boldsymbol{x}_f - \eta D(\boldsymbol{x}_f)} \quad (31)$$

yields a direction of descent.

The use of algorithm 1 with objective L^1 is different from pnpReG in three ways:

- We replace the denoising-loss (i.e. the first summand in (8)) with the FB reconstruction loss, training the PO on the resulting FB reconstruction-error rather than a Gaussian denoising problem.
- The algorithm evaluates L^{link} at $\tilde{\boldsymbol{x}}$ from Theorem 1, which depends on the equilibrium point of the

FB algorithm, instead of evaluating L^{link} on images only perturbed by AWGN, as is done in the pnpReG method of (Fermanian et al., 2023). If we consider only L^{link} at $\tilde{\boldsymbol{x}}$ from Theorem 1, it would train a PO and RG in a way such that the resulting FB and GD algorithms have the same FP. Looking at equation (20), this measures how "non-fixed" $\boldsymbol{x}_f$ is when put into the corresponding GD procedure, since the part with L^{link} is equal to zero if $\boldsymbol{x}_f$ is a FP of the GD procedure.

- We only consider the case where the σ used for L^{link} is identical with the standard deviation of the AWGN.

Similar to (Willingham et al., 2023), we evaluate the neural nets at images that actually appear when iteratively solving the MAP estimation problem from (4) for a given inverse problem, rather than on an assumed Gaussian perturbation.

Furthermore, we can use the upper bound to regularize the training of a DEQGD procedure by considering the reconstruction error of the GD procedure as well as regularizing the RG by tying it to a PO via the bound given in (25). This constrains the hypothesis-space by using two loss-terms that can both be used to approach the training objective outlined in remark 1.

The following sections will highlight how the resulting methods were trained and tested in order to compare them to pnpReG (Fermanian et al., 2023), MTDEQ (Willingham et al., 2023) and Gaussian denoiser plugged into the pnp ADMM algorithm (results from (Fermanian et al., 2023)).

5 EXPERIMENTS

Using the theory and algorithm introduced in the previous sections, this section discusses the details of training and the experiments made to compare the trained RGs.

For the experiments we have trained three different RGs using the pipeline outlined in algorithm 1:

- RG1, which is the prior that uses loss L^1 i.e. directly attempts to minimize the upper bound,
- A DEQGD, which uses loss L^2, and
- RERG, which uses loss L^3, combining both RG1 and a DEQGD.

5.1 Step-Sizes and Degradations

We use a step-size of $\eta_{\text{GD}} = 0.05$ for the GD procedure, and similar to (Fermanian et al., 2023), we use the adam optimizer (Kingma and Ba, 2014) to find the FP of the GD procedure from (14) for training. The FB procedure uses step-size $\eta_{\text{FB}} = 0.49$, which is taken from (Willingham et al., 2023).

We consider the following degradations in training:

- Gaussian deblurring with the level of blur σ_{b} in $[0,4]$
- Super-resolution with factors 1,2 and 4
- Pixel-wise completion where each pixel has a chance of $p_{\text{drop}} \in [0,0.99]$. Each selection of p_{drop} refers to a degradation.

Additionally, we used noise-levels sampled from a uniform distribution on $\Sigma := [0,50/255]$. For each iteration, there is a $1/3$ chance of choosing deblurring, super-resolution or pixel-wise completion. After this choice, the degree of the degradation is sampled via a uniform distribution on the corresponding set.

5.2 Dataset and Optimizer

The networks are optimized using the adam optimizer (Kingma and Ba, 2014) on a training-dataset consisting of the data-set from DIV2k (Agustsson and Timofte, 2017), the training-set from BSD500 (Arbelaez et al., 2011), flick2k (Lim et al., 2017) and the Waterloo Exploration Database (Ma et al., 2016). Overall, this data-set contains 8394 images. Each iteration takes a batch of 16 images and crops each image at a random location to the size of 128 by 128 pixels.

For the finding of the FP, the forward iterations of GD or FB are terminated if one or more of the following three conditions hold:

- The forward iteration has gone on for 500 iterations.
- The absolute value of any entry of an estimate is larger than 100.
- The mean square distance between two consecutive estimates is less than 10^{-7}.

The first condition is necessary because of time constraints; the second condition is in place to avoid the network forgetting what has been trained previously if the iterative method starts to diverge. This avoids unreasonably large gradients in failure-cases. The final condition is the proper convergence condition. Choosing a smaller margin for convergence or a higher number of maximum forward iterations tends to lead to a more accurate FP estimation. As a result, this can be expected to lead to more accurate, but slower, training.

In our training, the networks for the RG are initialized with the pnpReG (Fermanian et al., 2023), while the networks representing the PO are initialized with the MTDEQ regularizer (Willingham et al., 2023). The networks are trained for 150 epochs, starting with a step-size of 10^{-5}. The step-size is reduced by 75% every 15 epochs.

5.3 Comparisons to Other Methods

We compare our different algorithms to MTDEQ (Willingham et al., 2023), pnp ADMM with a Gaussian denoiser (called Gauss in the tables) and pnpReG. The PSNR values for the two latter methods are drawn from (Fermanian et al., 2023). The comparisons are done on set5 (Bevilacqua et al., 2012). Hyper-parameters used for the testing of the RGs in a GD algorithm are taken from (Fermanian et al., 2023) and were tuned to produce the best results for pnpReG, meaning they were not further optimized for RG1, DEQGD or RERG. Note that the weight of the regularization is given by the AWGN in the inverse problem, but for problems with no AWGN, a weight larger than zero was chosen to allow for regularization.

Based on these experiments, the next section will compare the performance of the different approaches and discuss the results.

6 RESULTS AND DISCUSSION

To showcase the differences that appear when using the introduced bound for RG1 and RERG, this section will compare the performance of four different RGs and examine differences in reconstruction performance.

Training a DEQGD is difficult, as convergence of the forward iterations is elusive and in our training only about half the iterations converged before reaching 500 iterations. This is why we used adam in forward iteration as well as the stopping condition of any entry having absolute value over 100. Training RG1 was quite stable and did not lead to any larger issues in training, as the FB algorithm used tends to be much more stable and converge faster.

The RG1 prior performs best for completion (see table 1), while the DEQGD performs better for many of the other applications (especially the noisy ones). This is likely the case because the GD procedure directly uses the (known) noise-level in each iteration. The FB algorithm, on the other hand, uses a proximal operator, in which the link between the level of AWGN given to the proximal operator is processed by a neural net, necessitating a training of this relationship. For zero-noise problems with little to no AWGN, this is a boon, because the regularization weight in a GD procedure may become too small to provide meaningful regularization. This is why the weight for the regularization is chosen to be larger than zero at test-time, when a noise-less problem is considered.

On any given task, RERG performs close to whichever of RG1 and DEQGD performs best, outperforming DEQGD on most problems. This means that using the upper bound in addition to L^{GD} in training can lead to a procedure that leverages the reconstruction errors of both, a GD and a FB algorithm to perform well across all the degradations considered. If one looks at the visual examples in figure 1, it appears as if the RERG avoids some of the artifacts that appear in RG1 and DEQGD, like the blurring of the butterfly's pattern on the bottom right in figure 1 (b), or the artifacts produced by RG1 in (a). While RG1 or the DEQGD both perform well for some problems and worse for others, RERG appears to perform more consistently across the different degradations considered.

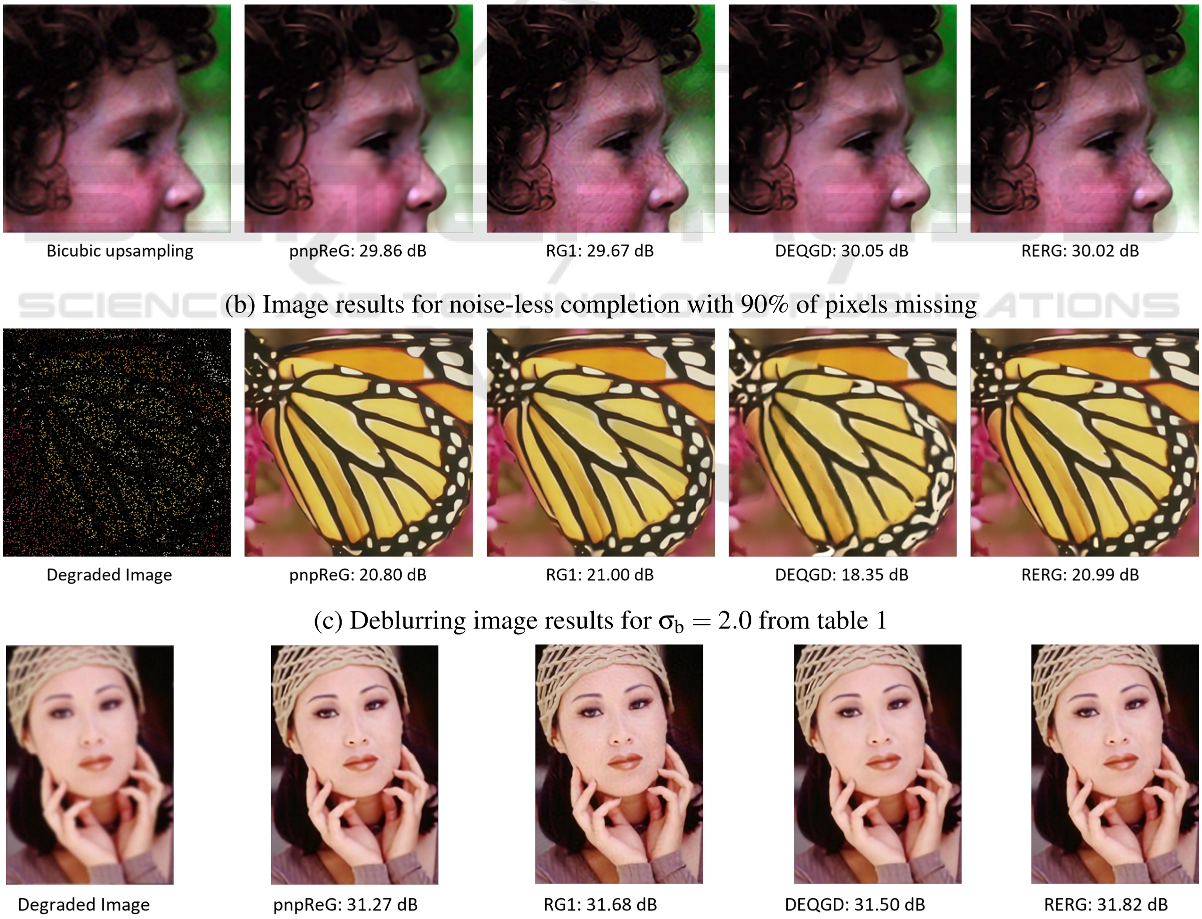
Figure 2 shows that for inverse problems where convergence is slow (necessitating a higher step-size) DEQGD and RERG may converge faster, because they are trained with a larger step-size and a limited amount of iterations. The problem highlighted in fig-

Table 1: This table compares methods on noise-less pixel-wise completion with 80 and 90 % of the pixels missing, respectively. It also displays deblurring results for $\sigma_{noise} = 0.01$ and Gaussian kernels with two different levels of blur σ_{b}. Results are reported as PSNR (dB) | SSIM (Structural Similarity Index Measure (Wang et al., 2004)). Both metrics are computed on the red green blue color channels.

Methods	Completion		Deblurring	
	80%	90%	$\sigma_{\mathrm{b}} = 1.6$	$\sigma_{\mathrm{b}} = 2.0$
Gauss	30.20 | 0.893	26.20 | 0.821	32.06 | 0.884	30.88 | 0.866
MTDEQ	**30.72** | 0.897	27.09 | **0.837**	32.82 | 0.898	31.83 | 0.881
pnpReG	30.36 | 0.894	26.94 | 0.830	32.51 | 0.898	31.19 | 0.884
RG1	30.58 | **0.899**	**27.18** | **0.837**	32.31 | 0.884	31.67 | 0.877
DEQGD	29.59 | 0.892	24.34 | 0.811	32.80 | 0.901	32.03 | 0.888
RERG	30.29 | 0.894	26.86 | 0.830	**32.90** | **0.902**	**32.17** | **0.890**

Table 2: Results for super-resolution on images that were down-sampled using a Gaussian kernel with $\sigma_b = 0.5$ and a bicubic kernel, respectively. This is done on a factor of 2 and 3, as well as two different noise-levels. Results are reported in the form PSNR | SSIM.

	Methods	Bicubic w/ σ_{noise}				Gaussian w/ σ_{noise}			
		0.00		0.01		0.00		0.01	
2x SR	Gauss	35.20	0.940	33.80	0.917	35.14	0.938	32.74	0.900
	MTDEQ	35.61	0.942	**34.43**	0.917	35.42	0.939	33.45	0.901
	pnpReG	35.34	0.943	34.29	0.922	35.30	0.942	33.41	0.906
	RG1	**35.66**	**0.944**	33.86	0.908	**35.57**	**0.943**	32.51	0.876
	DEQGD	35.50	0.941	34.40	**0.924**	35.40	0.940	33.66	**0.911**
	RERG	35.61	0.943	34.42	0.922	35.52	0.941	**33.69**	**0.911**
3x SR	Gauss	31.49	0.892	30.39	0.861	31.45	0.890	29.17	0.819
	MTDEQ	**32.10**	0.900	31.15	0.865	**31.94**	0.896	30.17	0.842
	pnpReG	31.75	0.896	31.13	0.877	31.60	0.896	30.39	0.858
	RG1	31.73	0.899	31.21	0.871	31.06	0.892	30.33	0.847
	DEQGD	32.07	0.899	31.31	**0.881**	31.74	0.898	30.61	**0.867**
	RERG	32.09	**0.901**	**31.38**	**0.881**	31.67	**0.899**	**30.74**	**0.867**


(a) Image results for noisy 3x Gaussian super-resolution.

(b) Image results for noise-less completion with 90% of pixels missing

(c) Deblurring image results for $\sigma_b = 2.0$ from table 1

Figure 1: Image results for the GD based methods with corresponding PSNR in dB.

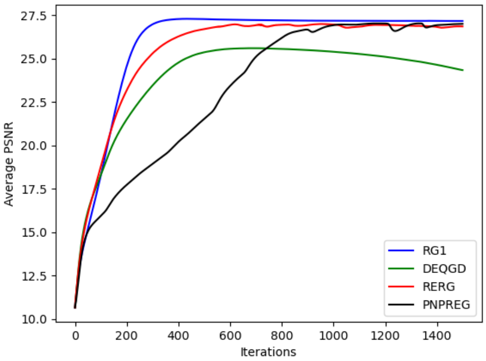


Figure 2: Average PSNR on set5 across the GD iterations for the solution of a noise-less completion problem with 90% of the pixels missing (see table 1).

ure 2 is reconstructed with a step-size of 0.025, which is the largest step-size used in testing, while most of the other problems use much smaller step-sizes and do not exhibit the same phenomenon.

One big issue with any GD-based scheme so far is that they are quite slow to test (and to train, if one uses a DEQGD approach). The test we performed had 1500 forward iterations, as do the tests in (Fermanian et al., 2023), meaning this is much slower than the pnp ADMM used for the MTDEQ (hyper-parameters for the pnp ADMM algorithm used can also be found in (Fermanian et al., 2023)).

Further investigation of different hyper-parameters for the training of a RERG could provide even better performance on the tasks considered and improve convergence speed at testing. There are procedures that can be used to speed up FP calculations, like the method from (Bai et al., 2021) or the correction terms from (Bai et al., 2022), that could be incorporated to speed up inference and FP estimation in training.

7 CONCLUSION

In this paper, we introduced an upper bound that can be used for the training of a GD procedure as both a training objective and a regularization. We compared four different types of RGs on a range of different inverse problems and discussed some of the differences, showing that the use of an upper bound for regularization to create a RERG can mitigate some of the disadvantages of the DEQGD and the RG1.

So far, few investigations have been done on RGs and we extended the theoretical framework introduced in (Fermanian et al., 2023) while proposing two novel ways of training RGs. We compared the resulting RGs and demonstrated that the RERG that combines both the upper bound and a DEQGD produces strong reconstruction results across all the inverse problems considered.

REFERENCES

Agustsson, E. and Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPR workshops*, pages 126–135. IEEE.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916.

Bai, S., Geng, Z., Savani, Y., and Kolter, J. Z. (2022). Deep equilibrium optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 620–630.

Bai, S., Kolter, J. Z., and Koltun, V. (2019). Deep equilibrium models. *arXiv preprint arXiv:1909.01377*.

Bai, S., Koltun, V., and Kolter, J. Z. (2021). Neural deep equilibrium solvers. In *International Conference on Learning Representations*.

Bauschke, H. H., Combettes, P. L., Bauschke, H. H., and Combettes, P. L. (2017). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer.

Bevilacqua, M., Roumy, A., Guillemot, C., and Alberi-Morel, M. L. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, pages 135.1–135.10. BMVA press.

Chan, S. H., Wang, X., and Elgendy, O. A. (2016). Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE TCI*, 3(1):84–98.

Fermanian, R., Le Pendu, M., and Guillemot, C. (2023). Pnp-reg: Learned regularizing gradient for plug-and-play gradient descent. *SIIMS*, 16(2):585–613.

Fung, S. W., Heaton, H., Li, Q., McKenzie, D., Osher, S., and Yin, W. (2022). Jfb: Jacobian-free back-propagation for implicit networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6648–6656.

Gilton, D., Ongie, G., and Willett, R. (2021). Deep equilibrium architectures for inverse problems in imaging. *IEEE TCI*, 7:1123–1133.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Le Pendu, M. and Guillemot, C. (2023). Preconditioned plug-and-play admm with locally adjustable denoiser for image restoration. *SIIMS*, 16(1):393–422.

Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M. (2017). Enhanced deep residual networks for single image super-resolution. In *CVPR*. IEEE.

Ling, Z., Xie, X., Wang, Q., Zhang, Z., and Lin, Z. (2022). Global convergence of over-parameterized deep equilibrium models. *arXiv preprint arXiv:2205.13814*.

Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., and Zhang, L. (2016). Waterloo exploration database: New challenges for image quality assessment models. *IEEE TIP*, 26(2):1004–1016.

Pesquet, J.-C., Repetti, A., Terris, M., and Wiaux, Y. (2021). Learning maximally monotone operators for image recovery. *SIIMS*, 14(3):1206–1237.

Romano, Y., Elad, M., and Milanfar, P. (2017). The little engine that could: Regularization by denoising (red). *SIIMS*, 10(4):1804–1844.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. (2013). Plug-and-play priors for model based reconstruction. In *GlobalSIP*, pages 945–948. IEEE.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

Willingham, S., Sjöström, M., and Guillemot, C. (2023). Prior for multi-task inverse problems in image reconstruction using deep equilibrium models. In *European Signal Processing Conference (EUSIPCO)*.

Winston, E. and Kolter, J. Z. (2020). Monotone operator equilibrium networks. *arXiv preprint arXiv:2006.08591*.

Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., and Timofte, R. (2021). Plug-and-play image restoration with deep denoiser prior. *IEEE TPAMI*.