Prior for Multi-Task Inverse Problems in Image Reconstruction Using Deep Equilibrium Models

Samuel Willingham, Mårten Sjöström, Christine Guillemot

Prior for Multi-Task Inverse Problems in Image Reconstruction Using Deep Equilibrium Models

Samuel Willingham
Inria Rennes,
Rennes, France
samuel.willingham@inria.fr

Mårten Sjöström
Mid Sweden University
Sundsvall, Sweden

Christine Guillemot
Inria Rennes,
Rennes, France
Mid Sweden University
Sundsvall, Sweden

***Abstract*—Inverse problems in imaging consider the reconstruction of clean images from degraded observations, like deblurring or inpainting. These inverse problems are generally ill-posed. Solving them therefore requires regularization, which exists in multiple approaches: plug-and-play (pnp) methods are designed to generically solve any inverse problem by replacing a regularizing proximal operator with a denoiser. Unrolled methods perform a fixed number of iterations and train a network end-to-end for a specific degradation, but necessitate re-training for each specific degradation. Deep equilibrium models (DEQs), on the other hand, iterate an unrolled method until convergence and thereby enable end-to-end training on the reconstruction error with simplified back-propagation. We have investigated to what extent a solution for several inverse problems can be found by employing a multi-task DEQ. This MT-DEQ is used to train a prior on the actual estimation error, in contrast to a theoretical noise model used for pnp methods. This has the advantage that the resulting prior is trained for a range of degradations beyond pure Gaussian denoising. The investigation also demonstrates that different search methods can be used in training (forward-backward) and in testing (alternating direction method of multipliers).**

***Index Terms*—Inverse Problems, Computer Vision, Image Reconstruction, Deep Equilibrium Models**

I. Introduction

Inverse problems refer to a broad class of problems which appear in a plethora of image processing and computer vision applications. In imaging, inverse problems refer to the problem of recovering images from a set of degraded or incomplete observations, for instance inpainting, super-resolution, and deblurring. We consider an observed image $y = A\hat{x} + \varepsilon$ to be the ground truth image $\hat{x} \in \mathbb{R}^d$ degraded by a degradation-operator $A : \mathbb{R}^d \to \mathbb{R}^{d'}$ with Additive White Gaussian Noise (AWGN) $\varepsilon \in \mathbb{R}^{d'}$, and $d, d' \in \mathbb{N}$.

This problem is generally ill-posed and can be solved by minimizing

$$\min_x \frac{1}{2}\|Ax - y\|_2^2 + R(x), \tag{1}$$

where the first term is called the data-term, $\|\cdot\|_2$ is the 2-norm and $R : \mathbb{R}^d \to \mathbb{R}$ is a regularization term that is intended to impose prior knowledge on the estimated image. To solve this minimization, the combination of optimization methods and trainable, complex image priors have recently emerged as a powerful tool for image reconstruction.

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.

Plug-and-play (pnp) methods are one approach to combine optimization and trainable priors: a pre-trained, network-based prior is plugged into an iterative optimization algorithm, such as the alternating direction method of multipliers (ADMM) [1], or the forward backward algorithm (FB) [2]. These algorithms use a proximal operator to implicitly represent the prior. A proximal operator is defined as

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

In the Gaussian case, this proximal operator for regularziation solves a Gaussian denoising problem, so [3] explains that the proximal operator can be replaced with a denoiser, because the proximal operator does not have to be exact and it is sufficient for it to decrease the underlying objective function. This approach has led to successful, general image-reconstruction methods [1]–[5], but it is worth considering that the regularizing proximal operator can be quite different from a Gaussian denoiser when its input is not Gaussian - as is the case for pixel-wise completion with no AWGN.

Priors can take several other forms, such as approximations of a projection operator on a learned image subspace as in [6]. Another well known approach is called regularization by denoising (RED) [7]. This approach uses a Laplacian regularizer that represents the cross-correlation between an estimated image and its denoising residual produced by some pre-trained denoiser. Finally, there are other ways to leverage pre-trained priors for regularization, e.g. Regularizing Gradient (pnp-ReG) [8], which trains the gradient of a Maximum A Posteriori regularizer jointly with its corresponding denoiser to be used in a gradient-based iterative algorithm. Pre-trained priors have the advantage of being generic in the sense that they can be used for any inverse problem without re-training. However, being trained independently of the task at hand, pnp methods may not result in optimal performance.

Unrolled optimization methods are less general in application, but by coupling optimization algorithms with end-to-end trained regularization, they recently emerged as powerful solutions to inverse problems [9]–[11]. Similar to pnp methods, the neural network (NN) is used to regularize the inverse problem. However, training such unrolled NNs end-to-end can

come with a large memory footprint and unrolled methods do not generally train the underlying prior, because they have a fixed number of iterations and do not generally converge [12].

In the context of image reconstruction, deep equilibrium (DEQ) models [12]–[16] can be seen as an extension of unrolled methods to with a theoretically infinite amount of iterations. DEQ models leverage fixed-point properties (convergence for over-parametrized structures is shown in [16]), allowing for simpler back-propagation, which can even be done in a Jacobian-free manner [14], visualized in Fig. 1.

DEQ models allow the training of an iterative method on the resulting reconstruction-error [12], while still training the full iterative method instead of limiting the amount of iterations. While the Gaussian assumption made for pnp does provide good results [1]–[5], it is worth exploring how one could improve the training of a plug-and-play prior, as the noise generated by a data-step in ADMM or FB is generally not Gaussian in nature. DEQ models allow for this by allowing end-to-end training the full iterative scheme on the reconstruction error.

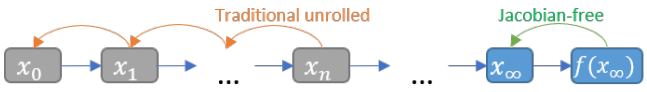


Fig. 1. Traditional back-propagation and Jacobian-free back-propagation [14] (only back-propagates through one step using the fixed-point x_∞ as input).

The contributions of this paper are:

- An MTDEQ model is used to train a network for multi-task proximal regularization on the resulting reconstruction-error, resulting in competitive performance.
- We demonstrate that the resulting MTDEQ-prior outperforms a Gaussian denoiser when used in pnp-ADMM, especially when the input to the regularization is very different from an image that is perturbed by AWGN.
- Using hyper-parameters that were set up for the pnp-ADMM with a Gaussian denoiser to test the MTDEQ-prior shows that it is flexible with respect to hyper-parameters.
- Training a prior with the FB algorithm and testing it in a pnp-ADMM algorithm shows the trained regularization is not constrained to the scheme it was trained in.

II. METHODS

We propose the multi-task DEQ (MTDEQ) model as a method of training a multi-task prior that can be used for different problems and different noise-levels. Since DEQ models can train a regularization for an inverse problem on the resulting reconstruction-error, we train a multi-task regularizer by exposing the NN to different degradations and noise-levels in training.

To solve the inverse problem for multiple tasks, we use the FB algorithm in conjuction with the DEQ model on a range of noise-levels and degradations, resulting in our MTDEQ model. We chose the FB algorithm for training, because it does not rely on a proximal operator for the minimization of the data-term.

Algorithm 1 MTDEQ

Require: Training Data $D \subset \mathbb{R}^d$, $d \in \mathbb{N}_{>0}$
  Set of problems $\mathcal{A} \subset \{A : \mathbb{R}^d \to \mathbb{R}^{\tilde{d}} \mid \tilde{d} \in \mathbb{N}_{>0}\}$
  Set of noise-levels $\Sigma \subset \mathbb{R}$
  Jacobian-free back-propagation procedure Optim
  Resizing procedure $P : \mathbb{R}^{\tilde{d}} \to \mathbb{R}^d$
  for a number of epochs **do**
    for $\hat{x} \in D$ **do**
      for $i = 1, ..., N$ **do**
        (A_i, σ_i) picked from $\mathcal{A} \times \Sigma$
        AWGN $\varepsilon_i \in \mathbb{R}^{\tilde{d}}$ with standard deviation σ_i
        $y_i \leftarrow A_i(\hat{x}) + \varepsilon_i$ ▷ Generate degraded images
        $z_i \leftarrow P(y_i)$
        while $\frac{1}{d}\|z_i - f_{i,\theta}(z_i)\|_2^2 \geq 10^{-7}$ **do**
          $z_i \leftarrow f_{i,\theta}(z_i)$ ▷ Find equilibrium point
        end while
        $L_i \leftarrow \frac{1}{d}\|f_{i,\theta}(z_i) - \hat{x}\|_2^2$
      end for
      $L \leftarrow \frac{1}{N}\sum_{i=1}^N L_i$ ▷ Sum losses for all problems
      $\tilde{\theta} \leftarrow \text{Optim}(\theta, L)$ ▷ Update NN-parameters
    end for
  end for

The MTDEQ that trains the prior is described in algorithm 1. For each ground truth image $\hat{x}$, we pick N degradations and noise-levels at random, leading to N degraded and noisy observations. This in turn leads to N estimates produced by the iterative procedure used. We use the FB algorithm in this training of the prior:

$$f_{i,\theta}(z) := Q_{\theta,\sigma_i}\left(z - \eta \frac{\delta \|A_i z - y_i\|_2^2}{\delta z}\right), \tag{3}$$

where $Q_{\theta,\sigma}$ represents regularization in form of a DRUNet (as in [4] and [5]) with parameters θ used on noise-level σ_i; $\eta \in \mathbb{R}$ is a step size. The iteration in the while-loop results in an equilibrium point $z_i^\infty \in \mathbb{R}^d$, which is the estimated reconstructed image. The loss L_i for each of the N estimates is averaged and then used for Jacobian-free back-propagation, which finds a direction of descent p that minimizes the loss for the N degradation problems considered:

$$p = \frac{1}{N}\sum_{i=1}^N \frac{\partial l(z, \hat{x})}{\partial z}\bigg|_{z=z_i^\infty} \frac{\partial f_{i,\theta}(z_i^\infty)}{\partial \theta}, \tag{4}$$

where $l(z, \hat{x}) = \frac{1}{d}\|z - \hat{x}\|_2^2$.

The NN is exposed to different degradations in training, and is so guided to represent a regularizer, as the data-fidelity step (i.e. the gradient descent on the data-term) takes care of degradation-dependent operations. Because a DEQ model iterates until convergence, the resulting NN represents a prior for the problem at hand that is not limited to the specific

architecture used. This means that if the regularizer is end-to-end trained while being exposed to different degradations and noise-levels, we can strive to train a more general prior that contains information on what proper clean images look like - beyond the assumption that a proximal regularization behaves like a Gaussian denoiser.

III. EXPERIMENTS

A. Step-sizes and degradations

We chose a step-size of $\eta = 0.49$ in the FB algorithm for the MTDEQ model. The noise-standard-deviations considered in training are taken from $\Sigma_{\text{noise}} = \{0.005 \cdot s \mid s = 0, 1..., 49\}$ (pixel-values are normalized to be in $[0, 1]$). The degradations used for training are:

- super-resolution by using bicubic down-sampling with factor $f_{\text{downscale}} \in \{1, 2, 3, 4\}$, where the iterations are initialized using bilinear resizing to the original size
- blurring with a Gaussian kernel with standard-deviation σ_{blur} picked from $\Sigma_{\text{blur}} = \{0.4, 0.8, 1.2, 1.6, 2.0, 2.4\}$
- pixel-wise completion with each pixel having a chance of $p_{\text{drop}} \in \{0.2, 0.4, 0.5, 0.6, 0.8, 0.9\}$ of being dropped - each selection of p_{drop} corresponds to a different degradation

Each image is degraded by 5 degradations and the resulting losses are averaged for back-propagation, meaning one ground-truth image is degraded five ways.

B. Dataset, architecture and optimizer

The dataset used in training is a collection of BSD500 (300 images [17]), flick2k (2650 images [18]), the Waterloo exploration database (4744 images [19]) and DIV2k (900 images [20]), similar to the one used in [4], [5], [8]. When an image is drawn from the dataset, it is cropped to be 128 times 128 pixels in size (the location of this cropping is random at each iteration).

The NN used is a DRUNet that uses the noise-level-map as additional input as in [5]; it is pre-trained as a Gaussian denoiser, using the NN obtained in [5]. The noise-standard-deviation given as input in each regularization step of the FB algorithm is $\sigma_{\text{denoise}} = \eta\sigma_{\text{noise}}$, where $\sigma_{\text{noise}} \in \Sigma_{\text{noise}}$ is the noise-level considered. Back-propagation is done in a Jacobian-free manner [14].

The algorithm is considered to have converged after the mean square error (MSE) of the difference between iterates is less than 10^{-7} and we capped the amount of iterations at 350. Further, we terminate iteration if any coordinate of the estimate has an absolute value larger than 10^{10}. In training, the procedure properly converges before 350 iterations and without any coordinate exceeding 10^{10} in more than 99% of cases.

The weights of the NNs are updated using the adam optimizer [21] with a batch-size of 16, using 250 epochs, starting with a learning-rate of 10^{-5} and scaling it by a factor of 0.25 every 25 epochs.

TABLE I

COMPARING OUR MTDEQ-BASED PRIOR USED IN PNP-ADMM ON THE SET5 DATASET [22] FOR COMPLETION WITH $\sigma_{\text{NOISE}} = 0$ AND 80 AND 90 PERCENT OF PIXELS DROPPED, RESPECTIVELY AS WELL AS AND GAUSSIAN DEBLURRING WITH DIFFERENT DEGREES OF BLURRING AND $\sigma_{\text{NOISE}} = 0.01$. ST COMP AND ST SR ARE SINGLE-TASK DEQ PRIORS TESTED IN PNP-ADMM (SEE III-C). THE RESULTS FOR EGULARIZATION BY DENOISING (RED) [7], PNP-ADMM WITH THE GAUSSIAN DENOISER (GAUSS) FROM [5] AND PNP-REG [8] ARE TAKEN FROM [8]. ALL RESULTS ARE REPORTED IN PEAK-SIGNAL-TO-NOISE-RATIO (PSNR) AND DB. THE HIGHEST PSNR PER TASK IS WRITTEN **BOLD** AND THE SECOND-BEST IS UNDERLINED.

Methods	Completion		Deblurring	
	80%	90%	$\sigma_{\text{blur}} = 1.6$	$\sigma_{\text{blur}} = 2.0$
RED	27.17	22.75	31.76	30.62
ST Comp	<u>30.42</u>	<u>27.11</u>	31.86	30.56
ST SR	29.71	25.67	32.22	30.77
Gauss	30.20	26.20	32.06	30.88
pnp-ReG	30.36	26.94	<u>32.51</u>	<u>31.19</u>
Ours	**30.72**	**27.15**	**32.74**	**31.72**
Ours vs. Gauss	+0.52	+0.95	+0.68	+0.84

TABLE II

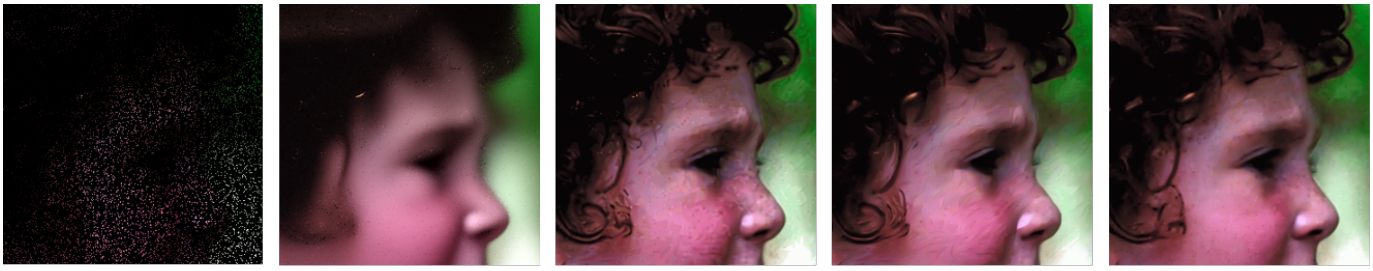
THIS TABLE SHOWS THE RESULTS FOR SUPER-RESOLUTION ON AN IMAGE THAT WAS DOWN-SAMPLED USING A GAUSSIAN KERNEL WITH $\sigma_{\text{BLUR}} = 0.5$ AND THE RESULTS FOR SUPER-RESOLUTION ON A BICUBICALLY DOWN-SAMPLED IMAGE. THESE DEGRADATIONS WERE TESTED ON TWO NOISE-LEVELS AND WITH SUPER-RESOLUTION FACTORS TWO AND THREE, RESPECTIVELY.

	Methods	Bicubic w/ σ_{noise}		Gaussian w/ σ_{noise}	
		0.00	0.01	0.00	0.01
2x SR	RED	35.05	33.78	34.99	32.84
	ST Comp	33.87	33.48	33.21	32.42
	ST SR	**35.78**	<u>34.02</u>	**35.6**1	32.85
	Gauss	35.20	33.80	35.14	32.74
	pnp-ReG	35.34	**34.29**	35.30	**33.41**
	Ours	<u>35.55</u>	34.01	<u>35.31</u>	<u>32.94</u>
	Ours vs. Gauss	+0.32	+0.21	+0.17	+0.20
3x SR	RED	31.47	30.78	31.44	30.05
	ST Comp	30.33	30.01	29.86	28.83
	ST SR	<u>31.84</u>	30.17	<u>31.66</u>	29.04
	Gauss	31.49	30.39	31.45	29.17
	pnp-ReG	31.75	**31.13**	31.60	**30.39**
	Ours	**32.13**	<u>30.62</u>	**31.93**	<u>29.51</u>
	Ours vs. Gauss	+0.64	+0.23	+0.48	+0.34

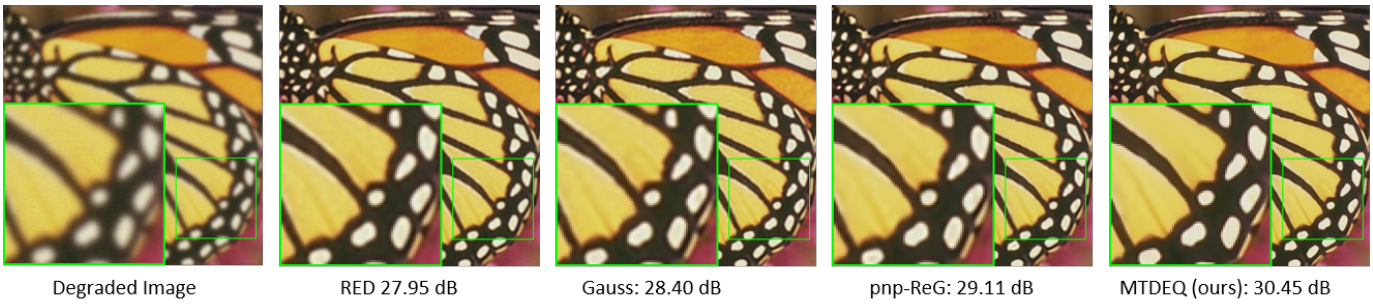
C. Comparing to plug-and-play

We compare our prior to RED, pnp-ADMM with a Gaussian denoiser and pnp-ReG by plugging the prior resulting from the MTDEQ into the ADMM algorithm as described and used for testing in [8]. The number of iterations, minimal denoising standard-deviation σ_0 and maximal denoising standard-deviation σ_m used in the ADMM are the ones that produced the best results for the pnp-ADMM reference method based on a Gaussian denoiser and are listed in [8]. This means that the hyperparameters were not fine-tuned for the MTDEQ-prior. The degradations that the MTDEQ model was trained on are Gaussian deblurring, dropped pixels and super-resolution on an image that was down-sampled using a bi-cubic kernel; this is significant because one of the degradations tested is super-

(a) Example for completion with 90 percent of pixels dropped and $\sigma_{\text{noise}} = 0$.



(b) Example of Gaussian deblurring with $\sigma_{\text{blur}} = 1.6$ and $\sigma_{\text{noise}} = 0.01$.



(c) Example for bicubic super-resolution with factor two and $\sigma_{\text{noise}} = 0.01$.

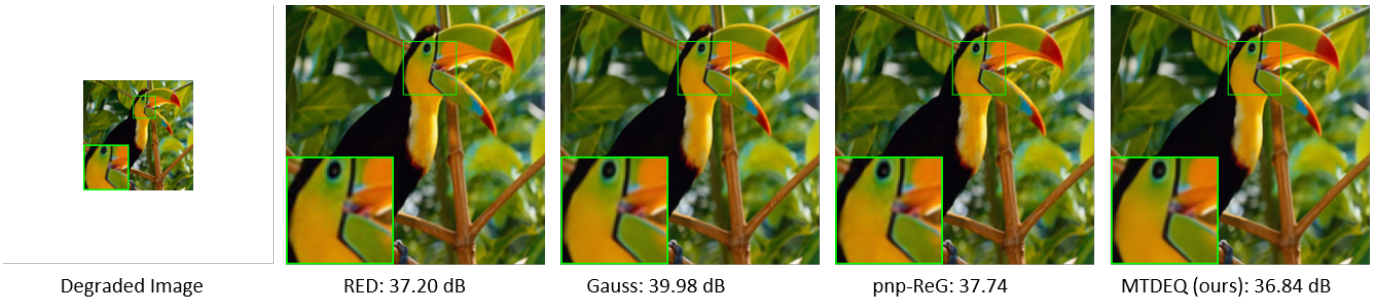


Fig. 2. Image-comparisons with RED [7], pnp-ADMM with the Gaussian denoiser from [5] (Gauss), pnp-ReG [8], and pnp-ADMM with the prior obtained via MTDEQ; results are reported in PSNR.

resolution on a image that is degraded using a Gaussian kernel (with $\sigma_{\text{blur}} = 0.5$). The results can be seen in Tables I and II and the visual results can be observed in Fig. 2. The pnp-ADMM is the closest comparison to our MTDEQ-prior, since it uses the same algorithm in testing, only with a Gaussian denoiser as prior.

ST Comp and ST SR are single-task DEQ priors trained using the MTDEQ with the following parameters: Both have N=1 and ST Comp is trained for $\mathcal{A}$ only containing pixel-wise completion with probability $p_{\text{drop}} = 0.9$ and no AWGN (i.e. $\Sigma_{\text{noise}} = \{0.00\}$). ST SR is trained for $\mathcal{A}$ containing only bicubic down-sampling with factor $f_{\text{downscale}} = 2$ and AWGN of $\Sigma_{\text{noise}} = \{0.01\}$. In testing, both single-task priors were plugged into the pnp-ADMM scheme from [8] with the same parameters as the Gaussian denoiser and our MTDEQ-prior.

The following section will discuss the results found from the above experiments and talk about the advantages and disadvantages of our MTDEQ-prior when compared to other methods.

IV. Results and Discussion

Using the MTDEQ model to train on different degradations and noise-levels leads to a multi-task prior usable for iterative methods, like the pnp-ADMM algorithm we use for testing. Different from pnp training, the prior trained with MTDEQ is exposed to a larger range of qualitatively different inputs (i.e. images that are not purely perturbed by AWGN).

The results were obtained using a different iterative algorithm from the one used to train the MTDEQ-prior. This means that the NN represents a regularization that is not closely linked to the exact structure of the iterative algorithm used and can be said to represent an image prior for the problems at hand.

As discussed in III-A, the MTDEQ-prior is trained for a single step size and still performs well when used in the fine-tuned setting of the pnp-ADMM algorithm [8], where different step-sizes and denoising-levels are used for different degradations. This suggests that the method could be further improved, if one could find a correspondence between different degradations and step-sizes to be used for the MTDEQ.

The proposed MTDEQ-prior performs well on multi-task image reconstruction and this is especially obvious on problems with no AWGN added to the input, where the degradation itself is responsible for the entire difference between observation and ground truth. This is quite reasonable, because the best denoiser for a zero-noise-problem is the identity, which provides no regularization at all. Pnp methods usually remedy this by using a low denoising-level instead of zero, so the denoiser still provides some regularization. A trick of this sort is not necessary for a regularizer that was trained on the reconstruction error, as it can learn to add meaningful regularization, even when the "noise-level-map" is set to zero. Also, the closer the inverse problem is to a Gaussian denoising problem, the more appropriate a Gaussian denoiser is for the purpose of regularization. Thus, it can be seen that the use of a Gaussian denoiser for proximal pnp methods can be further improved using DEQ, especially when the inverse problems considered differ significantly from pure Gaussian denoising problems.

Single-task DEQ models do not perform especially well when compared in a pnp-ADMM algorithm, but we have seen that a single-task DEQ model clearly outperforms the MTDEQ on the problem and noise-level it was trained on, when used in FB. While the comparisons in Tables I and II are unfair to single-task DEQ models because they are only trained for a single noise-level and pnp-ADMM uses scaling denoising-levels, the comparisons showcase that the MTDEQ model adds significant performance when used on a range of noise-levels and degradations. Thus, the multi-task training-structure does add performance besides the performance single-task DEQ models bring to the table.

We consider the pnp-ADMM using a Gaussian denoiser for proximal regularization to be the closest comparison to our prior, as the only difference is the used regularization. When comparing the reconstruction results for the pnp-ADMM which uses a Gaussian denoiser and our MTDEQ-prior, it is clear that the MTDEQ-prior produces better performance across all the degradations and noise-levels tested, meaning that using an MTDEQ model to fine-tune a prior beyond a Gaussian denoiser does indeed increase performance.

As the DEQ requires iterating until convergence before allowing updates to the weights, it is significantly slower in training when compared to a denoiser, which only has one forward-operation. Combining the training objectives of a denoiser with the loss function used by a MTDEQ model approach may allow for a speed up in training while further improving performance.

V. Conclusion

Among the different ways of solving inverse problems, DEQ models permit end-to-end training of a regularization and are well-suited to solve multi-task inverse problems, bridging the gap between end-to-end trained methods like unrolled models or single-task DEQ models and pnp methods: An MTDEQ model can be used to train a multi-task prior that outperforms a Gaussian denoiser in a pnp-ADMM scheme.

References

[1] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play admm for image restoration: Fixed-point convergence and applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, 2016.

[2] 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.

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

[4] M. Le Pendu and C. Guillemot, "Preconditioned plug-and-play admm with locally adjustable denoiser for image restoration," *arXiv preprint arXiv:2110.00493*, 2021.

[5] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[6] J. Rick Chang, C.-L. Li, B. Poczos, B. Vijaya Kumar, and A. C. Sankaranarayanan, "One network to solve them all–solving linear inverse problems using deep projection models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5888–5897.

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

[8] R. Fermanian, M. L. Pendu, and C. Guillemot, "Learned gradient of a regularizer for plug-and-play gradient descent," *arXiv preprint arXiv:2204.13940*, 2022.

[9] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein, "Unrolled optimization with deep priors," *arXiv preprint arXiv:1705.08041*, 2017.

[10] M. Mardani, Q. Sun, S. Vasawanal, V. Papyan, H. Monajemi, J. Pauly, and D. Donoho, "Neural proximal gradient descent for compressive imaging," *arXiv preprintarXiv:1806.03963*, 2018.

[11] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep admm-net for compressive sensing mri," in *Proceedings of the 30th international conference on neural information processing systems*, 2016, pp. 10–18.

[12] D. Gilton, G. Ongie, and R. Willett, "Deep equilibrium architectures for inverse problems in imaging," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 1123–1133, 2021.

[13] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," *arXiv preprint arXiv:1909.01377*, 2019.

[14] S. W. Fung, H. Heaton, Q. Li, D. McKenzie, S. Osher, and W. Yin, "Fixed point networks: Implicit depth models with jacobian-free backprop," *arXiv preprint arXiv:2103.12803*, 2021.

[15] E. Winston and J. Z. Kolter, "Monotone operator equilibrium networks," *arXiv preprint arXiv:2006.08591*, 2020.

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

[17] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2010.161

[18] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[19] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2016.

[20] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proceedings of the British Machine Vision Conference*. BMVA press, 2012, pp. 135.1–135.10.