An $\mathcal{A}$-adaptive Loop Unrolled Architecture for Solving Inverse Problems with Forward Model Mismatch

Peimeng Guan [1] **Naveed Iqbal** [2] **Mark A. Davenport** [1] **Mudassir Masood** [2]

Abstract

In inverse problems (IP) we aim to recover the underlying signal from noisy measurements that are generated according to a known forward model. Classical methods for solving IPs usually involve minimizing a least-squares data fidelity term together with a predetermined regularization function, which often leads to unsatisfactory reconstructions. *loop unrolling* (LU) architecture addresses this issue by unrolling the optimization iterations into a sequence of neural networks that in effect learn a regularization function from data. While LU is currently a state-of-the-art method in many applications, the accuracy of the forward model is crucial for its success. This assumption can be limiting in many physical applications due to model simplifications or uncertainties in the apparatus. To address forward model mismatch, this work introduces a forward model residual network, and with an extra variable splitting step, the proposed method can adapt to uncertain forward models accordingly. The method achieves ~ 2 dB PSNR increment in image blind deblurring and seismic blind deconvolution tasks by effectively learning the updates in reconstruction and forward model jointly.

1. Introduction

Consider an inverse problem of the following form:

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{\epsilon}. \tag{1}$$

The goal is to reconstruct the latent signal $\boldsymbol{x}$ from the measurements $\boldsymbol{y}$ in the presence of noise $\boldsymbol{\epsilon}$, where typically $\mathcal{A}$ is assumed to be known. Inverse problems are generally challenging because they are often ill-posed, *i.e.,* the solution is not unique, or the reconstruction is highly sensitive to noise and/or model mismatch.

The traditional approach to recovering $\boldsymbol{x}$ from the measurements $\boldsymbol{y}$ is by solving a regularized optimization problem of the form:

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x})\|_2^2 + \gamma r(\boldsymbol{x}), \tag{2}$$

where $\gamma \geq 0$ is an appropriately-chosen parameter. The regularizer r is usually predetermined based on some known or desired structure, e.g., ℓ_1-, ℓ_2-, or total variation (TV) norm to promote sparsity, smoothness, or edges in image reconstructions, respectively. Solving (2) requires careful consideration of the underlying physics or the forward model $\mathcal{A}$ to obtain a stable and accurate reconstruction. However, knowing $\mathcal{A}$ can be challenging in practice. The reasons for this include inaccurate measurements and challenging calibration, highly nonlinear and/or computationally expensive models replaced by simplified versions, as well as access to only approximations of certain features. Usually, some knowledge of the true model, designated $\mathcal{A}_0$ in this work, is available. This occurs in many applications, including blind deconvolution/deblurring problems, recovering seismic layer models using the simplified acoustic wave equation as the forward model, determining fault locations in media with unknown structures, etc.

When the forward model is precisely known, the inverse problem in (2) can be solved via classical optimizations where r is predefined. Machine-learning approaches, as summarized in (Arridge et al., 2019) and (Ongie et al., 2020), have demonstrated superior reconstruction performance by effectively learning a regularizer from data. For example, the Plug-and-Play method (Venkatakrishnan et al., 2013) trains a general denoiser independently of the forward model and iteratively minimizes (2) with the learned denoiser as the regularization updates. Another class of approaches, the *loop unrolling* (LU) method (Gregor & LeCun, 2010; Hershey et al., 2014; Adler & Öktem, 2018; Gu et al., 2017) and its extension using deep equilibrium model (Gilton et al., 2021a), builds on the observation that many iterative algorithms for solving (2) can be re-expressed as a sequence

[1] Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA, United States [2] Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. Correspondence to: Peimeng Guan <guanpeimeng@gatech.edu>.

Accepted after peer-review at the 1st workshop on Synergy of Scientific and Machine Learning Modeling, SynS & ML ICML, Honolulu, Hawaii, USA. July, 2023. Copyright 2023 by the author(s).

of neural networks (Gregor & LeCun, 2010), which can then be trained to remove artifacts and noise patterns associated with a known $\mathcal{A}$, resulting in higher quality reconstructions.

While these optimization-guided machine learning solvers have demonstrated impressive performance, they encounter challenges while dealing with inaccurate forward models. Plug-and-Play, LU, and its deep equilibrium extensions all entail a gradient update of the data-fidelity term, which can be in error when the forward model is inaccurate. This causes the learned denoiser or regularizer to be ineffective. In particular, while LU often exhibits state-of-the-art performance, has faster runtime, and has improved stability in training (Guan et al., 2022) compared to other approaches, it requires precise knowledge of the forward model. Appendix A demonstrates the sensitivity of LU to the forward model.

To handle errors in $\mathcal{A}$, one natural approach is to train the network with various incorrect $\mathcal{A}$'s. A deep equilibrium inverse problem solver is trained using this approach in (Hu et al., 2023), and is shown to be more robust to changes in $\mathcal{A}$ than the Plug-and-Play method. However, this approach does not address the incorrectness in the model and it requires a large amount of data. Our experiments in Section 4 show that an LU method trained with mismatched $\mathcal{A}$'s ineffectively updates the reconstruction. Model mismatches in linear inverse problems can be more directly resolved by alternatively updating the model parameters and the underlying signal. For instance, (Fergus et al., 2006; Cai et al., 2009; Levin et al., 2009; Cho & Lee, 2009) reconstructed the latent images with updates in the forward model. However, these methods are typically limited to linear cases with predefined regularizers. To improve performance in this setting, (Nan & Ji, 2020) and (Gilton et al., 2021b) learn a regularizer from data, resulting in improved performance on linear inverse problems.

Our work further introduces a residual network that can learn the forward model more flexibly to handle nonlinear problems and hence has broader scientific applications. Existing LU approaches learn a black-box solver that treats the provided $\mathcal{A}_0$ as if it is the true forward model, while $\mathcal{A}$-adaptive LU updates the forward model and the reconstruction jointly, providing both a solution to the IP and an estimate of the true forward model.

This paper presents a machine learning-based algorithm for solving inverse problems when an approximation of the forward model is available. Unlike LU which requires precise knowledge of $\mathcal{A}$, the proposed method $\mathcal{A}$-adaptive LU can iteratively update the forward model along with the reconstruction. The proposed approach introduces a neural network that can learn the forward model mismatch and can accommodate both linear and nonlinear inverse problems. Compared to LU trained with various mismatched $\mathcal{A}$'s, we demonstrate an improvement of $\sim$2 dB in reconstruction in image blind deblurring and seismic blind deconvolution tasks.

2. Related Methods

2.1. Loop Unrolling Methods

The traditional approach to solving inverse problems is by formulating them as an optimization problem of the form (2). A natural approach to solving (2) is via a *proximal gradient descent* algorithm, which can be applied even when the regularizer r is not differentiable, as is often the case. The resulting update involves first taking a gradient step $\boldsymbol{x}_k - \eta\mathcal{A}^\top(\boldsymbol{y} - \mathcal{A}(\boldsymbol{x}_k))$ (with a fixed step size η) that aims to minimize the data-fidelity term in (2). This is then followed by a proximal step, resulting in the iteration:

$$\boldsymbol{x}_{k+1} = \text{prox}_{\gamma,r}(\boldsymbol{x}_k - \eta\mathcal{A}^\top(\boldsymbol{y} - \mathcal{A}(\boldsymbol{x}_k))), \qquad (3)$$

where $\mathcal{A}^\top$ denotes the adjoint operator of $\mathcal{A}$, $\gamma > 0$ is again the parameter that controls the weight of the regularizer, and the proximal operator is defined as:

$$\text{prox}_{\gamma,r}(\boldsymbol{x}) = \arg\min_{\boldsymbol{z}} \frac{1}{2\gamma}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + r(\boldsymbol{z}). \qquad (4)$$

As we can see, the choice of regularizer manifests itself entirely through the proximal operator. The LU algorithm essentially keeps the update in (3) but replaces the proximal operator with a neural network, and limits the algorithm to a finite number of iterations K. The final output $\boldsymbol{x}_K$ is compared with the ground truth $\boldsymbol{x}$ and the network parameters are updated accordingly through *end-to-end* training.

By structuring the network in a way that mirrors proximal gradient descent – and taking advantage of an accurate descent direction derived from knowledge of $\mathcal{A}$ – the learned portion of the network can be interpreted as the proximal operator of a learned regularizer that enforces desired signal structures. However, when the forward model is inexact or only approximately known, the gradient update in (3) can introduce errors. Since LU is trained end-to-end, the error will manifest itself in the learned proximal operators. As a result, the neural network used in LU will no longer act as a pure proximal operator since it must both enforce signal structure as well as compensate for the errors in $\mathcal{A}$, potentially becoming less effective and interpretable.

2.2. Half-Quadratic Splitting (HQS)

Another key tool that we will rely on in our approach is *variable splitting*. Variable splitting is an iterative optimization method that solves problems where the objective function is a sum of multiple components (Geman & Yang, 1995; Nikolova & Ng, 2005; Bergmann et al., 2015; Hurault et al., 2022; Yang & Wang, 2017). It works by introducing an auxiliary variable and iteratively optimizing the objective function with respect to each variable while fixing the others. For example, by introducing an auxiliary variable $\boldsymbol{z}$ we can re-express the optimization problem in (2) as

$$\underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x})\|_2^2 + \gamma r(\boldsymbol{z}), \ \text{s.t. } \boldsymbol{x} = \boldsymbol{z}. \quad (5)$$

The constraints can be removed by solving

$$\underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x})\|_2^2 + \gamma r(\boldsymbol{z}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2, \quad (6)$$

where $\mu \geq 0$ is a tuning parameter. This optimization can be solved by iteratively updating $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ until convergence:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{x})\|_2^2 + \mu\|\boldsymbol{x} - \boldsymbol{z}_k\|_2^2, \quad (7)$$

$$\boldsymbol{z}_{k+1} = \arg\min_{\boldsymbol{z}} \gamma r(\boldsymbol{z}) + \frac{\mu}{2}\|\boldsymbol{z} - \boldsymbol{x}_{k+1}\|_2^2. \quad (8)$$

3. Proposed Method: An $\mathcal{A}$-adaptive Loop Unrolled Architecture

The discussion above has assumed exact knowledge of $\mathcal{A}$. Here we now suppose that we have an initial guess of the forward model $\mathcal{A}_0$ and a consider neural network f_θ that learns the measurement residual due to model misfit given the signal $\boldsymbol{x}$ and $\mathcal{A}_0$, i.e.,

$$\boldsymbol{y} = \mathcal{A}_0(\boldsymbol{x}) + \boldsymbol{\epsilon} = \mathcal{A}_0(\boldsymbol{x}) + f_\theta(\boldsymbol{x}, \mathcal{A}_0) + \boldsymbol{\epsilon}. \quad (9)$$

We assume that $\mathcal{A}_0$ is a useful estimate of the true forward model, but not exact. We can then express the optimization problem in (2) as

$$\begin{aligned} \underset{\boldsymbol{x},\theta}{\text{minimize}} \ & \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}_0(\boldsymbol{x}) - f_\theta(\boldsymbol{x}, \mathcal{A}_0)\|_2^2 + \gamma r(\boldsymbol{x}) \\ & + \tau\|f_\theta(\boldsymbol{x}, \mathcal{A}_0)\|_2^2. \end{aligned} \quad (10)$$

Introducing an auxiliary variable $\boldsymbol{z}$, the solution of (10) is equivalent to solving

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{z},\theta}{\text{minimize}} \ & \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}_0(\boldsymbol{z}) - f_\theta(\boldsymbol{z}, \mathcal{A}_0)\|_2^2 + \gamma r(\boldsymbol{x}) \\ & + \tau\|f_\theta(\boldsymbol{z}, \mathcal{A}_0)\|_2^2 + \lambda\|\boldsymbol{x} - \boldsymbol{z}\|_2^2. \end{aligned} \quad (11)$$

Similar to the HQS updates, we first initialize $\boldsymbol{x}_0$, $\boldsymbol{z}_0$, and θ_0. We then update each variable in the objective function by keeping other variables fixed. For $k = 1, 2, \ldots, K$, we have

$$\begin{aligned} \boldsymbol{z}_{k+1} &= \arg\min_{z} \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}_0(\boldsymbol{z}) - f_{\theta_k}(\boldsymbol{z}, \mathcal{A}_0)\|_2^2 \\ &\quad + \tau\|f_{\theta_k}(\boldsymbol{z}, \mathcal{A}_0)\|_2^2 + \lambda\|\boldsymbol{x}_k - \boldsymbol{z}\|_2^2, \\ \theta_{k+1} &= \arg\min_{\theta} \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}_0(\boldsymbol{z}_k) - f_\theta(\boldsymbol{z}_k, \mathcal{A}_0)\|_2^2 \\ &\quad + \tau\|f_\theta(\boldsymbol{z}_k, \mathcal{A}_0)\|_2^2, \\ \boldsymbol{x}_{k+1} &= \text{prox}_{\lambda,r}(\boldsymbol{x}_k - \eta(\boldsymbol{x}_k - \boldsymbol{z}_{k+1})). \end{aligned} \quad (12)$$

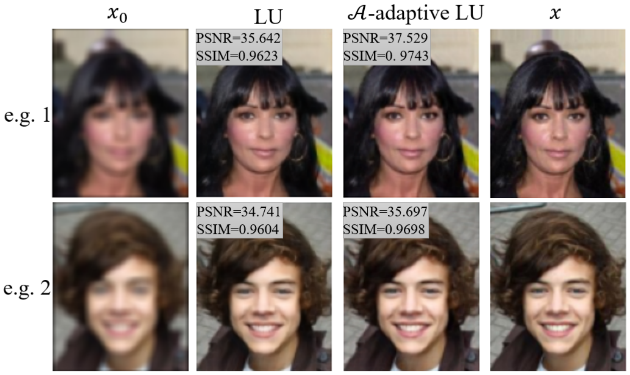

Figure 1. Comparing the deblurring results using LU and $\mathcal{A}$-adaptive LU to the ground truth, where $\boldsymbol{x}_0$ is the initial blurry images and $\boldsymbol{x}$ is the ground truth.

It should be noted that the update on $\boldsymbol{z}$ no longer has a closed-form solution due to the nonlinearity of f_θ. However, this can be efficiently computed using Autograd in Pytorch (Paszke et al., 2017) or other differentiation computing algorithms (Revels et al., 2016). Meanwhile, the update on θ follows the regular backpropagation for network parameters. We can then replace the proximal operator with a neural network, and the update on $\boldsymbol{x}$ connects all the components to form an LU network. The parameters in the proximal network are learned through end-to-end training, where η is a trainable step size.

4. Experiments

In this study, we demonstrate the efficacy of the proposed $\mathcal{A}$-adaptive LU algorithm and show an improvement in reconstruction by learning a more accurate forward model compared to the following baseline methods: 1) a neural network, $d_\phi : \boldsymbol{x}_0 \mapsto \boldsymbol{x}$, which maps from the initial reconstruction $\boldsymbol{x}_0$ to the ground-truth $\boldsymbol{x}$, and 2) an LU network that is trained with inexact $\mathcal{A}_0$'s from the initial $\boldsymbol{x}_0$. Notice that $\boldsymbol{x}_0$ is initialized with $\mathcal{A}^\top \boldsymbol{y}$ when $\boldsymbol{x}$ and $\boldsymbol{y}$ are from different spaces or $\boldsymbol{x}_0 = \boldsymbol{y}$ if they are from the same space. We evaluate the algorithms on image blind deblurring and seismic blind deconvolution tasks. For each task, the network d_ϕ and the proximal network in LU and in $\mathcal{A}$-adaptive LU share the same architecture.

Image Blind Deblurring. In this problem we aim to remove the blur from images $\boldsymbol{y}$ when a small amount of noise is present. The forward model is defined by a Gaussian blur kernel where we have inaccurate knowledge of the variance and size of the kernel. To validate this approach, experiments are conducted on the CelebA dataset (Liu et al., 2015), which was resized to $3 \times 120 \times 100$. The blurry images are generated with different Gaussian kernels for each pair of data $(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{A}_{0,i})$.

Seismic Blind Deconvolution. Next, a seismic deconvolution problem commonly encountered in geophysical surveys is considered. In this scenario, an acoustic wave generated by a vibroseis truck on the surface of the earth propagates through the earth's layers. The reflected signals are collected by the geophones and stored as raw measurements. These observations undergo some data pre-processing such as filtering, normal moveout correction, and common midpoint staking (Mousa & Al-Shuhail, 2011) to remove undesirable signals and render the seismic signals interpretable. After the pre-processing, the measurements $\boldsymbol{y}$ are obtained in the form of (1). Examples of $\boldsymbol{y}$ and $\boldsymbol{x}$ can be found as $\boldsymbol{x}_0$ and $\boldsymbol{x}$ in Figure 2. However, the frequencies of the acoustic wave are often inaccurately recorded due to the limited resolution and noise in the measurement process (Zabihi Naeini & Sams, 2017), which leads to an inaccurate estimate of the wavelet in the forward model. The forward process can be viewed as a convolution between the acoustic wave and the layer reflectivity $\boldsymbol{x}$. Therefore, the goal of seismic deconvolution is to reconstruct the layer reflectivities $\boldsymbol{x}$, with the inaccurate estimate of the wavelet in the forward model taken into account. The true model for each data pair $(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{A}_{0,i})$ can be expressed as follow,

$$\boldsymbol{y}_i = \boldsymbol{A}_{0,i}\boldsymbol{x}_i + f_\theta(\boldsymbol{x}_i, \boldsymbol{w}_i) + \boldsymbol{\epsilon}_i, \tag{13}$$

where $\boldsymbol{A}_{0,i}$ is a Toeplitz matrix derived from the inexact source wavelet $\boldsymbol{w}_i$. The measurement is simulated by applying an inaccurate wavelet with small additive Gaussian noise to the forward model. Notice that noise added to the true model may result in artifacts due to an extra magnification factor applied by $\boldsymbol{x}_i$, which is distinct from the measurement noise $\boldsymbol{\epsilon}_i$. The data is generated following the procedure in (Iqbal et al., 2019).

Reconstruction Results. Table 1 presents the comparison of the average testing mean-squared loss (MSE), peak Signal-to-Noise Ratio (PSNR) in dB, and Structural Similarity Index Measure (SSIM) of the reconstructions for all tasks. The neural network d_ϕ learns a direct inverse independently of $\mathcal{A}_{0,i}$. It's worth noting that LU outperforms d_ϕ even when the gradient update points to incorrect directions. This is due to its ability to break down a difficult ill-posed problem into multiple smaller proximal steps that are easier to learn. Moreover, end-to-end training enables LU to fix errors resulting from the model mismatch, thus maintaining higher-quality results. However, LU trains a general inverse problem solver with inaccurate forward models. The proposed $\mathcal{A}$-adaptive LU improved LU by adapting for the model mismatch for each data pair, significantly outperforming LU and d_ϕ in all tasks. In addition to the quantitative evaluation, we also present a visual comparison of the reconstruction results using LU and $\mathcal{A}$-adaptive LU for the three tasks in Figure 1 and 2. These figures show that the proposed algorithm removes more artifacts while preserving more detailed information.

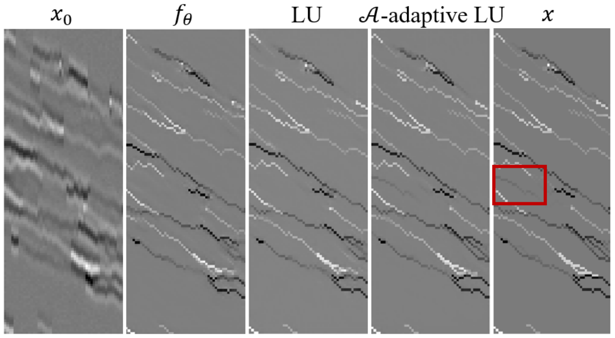

Figure 2. Comparing the deconvolution results using f_θ, LU, $\mathcal{A}$-adaptive LU to the ground truth. $\boldsymbol{x}_0$ shows the noisy trace after pre-processing and x is the ground-truth earth layer reflectivity. Regions corresponding to qualitative improvements is highlighted in red boxes in the ground truth.

Table 1. Average testing MSE, PSNR and SSIM for d_ϕ, LU and $\mathcal{A}$-adaptive LU. The best performances for each task are in bold.

		Deblurring	Deconvolution
d_ϕ	MSE	0.004238	0.002822
	PSNR	24.002	22.816
	SSIM	0.7689	0.7421
LU	MSE	0.000507	0.001914
	PSNR	34.380	24.785
	SSIM	0.94295	0.8407
$\mathcal{A}$-adaptive LU	MSE	**0.000355**	**0.001309**
	PSNR	**35.956**	**27.271**
	SSIM	**0.9585**	**0.8960**

5. Conclusion

Although the loop unrolling method is a powerful inverse problem solver, it requires accurate knowledge of the forward models, which can be impractical for many applications. To address this problem, this paper introduced a novel architecture of LU to learn the forward mismatch. The experimental results in two different tasks (image blind deblurring, seismic blind deconvolution) have shown the proposed $\mathcal{A}$-adaptive LU not only outperformed LU trained as a unified solver with inaccurate forward models but also more effectively learned the reconstruction and the forward model updates.

Acknowledgements

References

Adler, J. and Öktem, O. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

Arridge, S., Maass, P., Öktem, O., and Schönlieb, C.-B. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

Bergmann, R., Chan, R. H., Hielscher, R., Persch, J., and Steidl, G. Restoration of manifold-valued images by half-quadratic minimization. *arXiv preprint arXiv:1505.07029*, 2015.

Cai, J.-F., Ji, H., Liu, C., and Shen, Z. Blind motion deblurring from a single image using sparse approximation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 104–111. IEEE, 2009.

Cho, S. and Lee, S. Fast motion deblurring. In *ACM SIGGRAPH Asia 2009 papers*, pp. 1–8. 2009.

Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T. Removing camera shake from a single photograph. In *Acm Siggraph 2006 Papers*, pp. 787–794. 2006.

Geman, D. and Yang, C. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7):932–946, 1995. doi: 10.1109/83.392335.

Gilton, D., Ongie, G., and Willett, R. Deep equilibrium architectures for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:1123–1133, 2021a.

Gilton, D., Ongie, G., and Willett, R. Model adaptation for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:661–674, 2021b.

Gregor, K. and LeCun, Y. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pp. 399–406, 2010.

Gu, S., Xie, Q., Meng, D., Zuo, W., Feng, X., and Zhang, L. Weighted nuclear norm minimization and its applications to low level vision. *International journal of computer vision*, 121:183–208, 2017.

Guan, P., Jin, J., Romberg, J., and Davenport, M. A. Loop unrolled shallow equilibrium regularizer (luser)–a memory-efficient inverse problem solver. *arXiv preprint arXiv:2210.04987*, 2022.

Hershey, J. R., Roux, J. L., and Weninger, F. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv preprint arXiv:1409.2574*, 2014.

Hu, J., Shoushtari, S., Zou, Z., Liu, J., Sun, Z., and Kamilov, U. S. Robustness of deep equilibrium architectures to changes in the measurement model. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Hurault, S., Leclaire, A., and Papadakis, N. Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. In *International Conference on Machine Learning*, pp. 9483–9505. PMLR, 2022.

Iqbal, N., Liu, E., Mcclellan, J. H., and Al-Shuhail, A. A. Sparse multichannel blind deconvolution of seismic data via spectral projected-gradient. *IEEE Access*, 7:23740–23751, 2019. doi: 10.1109/ACCESS.2019.2899131.

Levin, A., Weiss, Y., Durand, F., and Freeman, W. T. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 1964–1971. IEEE, 2009.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Mousa, W. A. and Al-Shuhail, A. A. Processing of seismic reflection data using matlab™. *Synthesis lectures on signal processing*, 5(1):1–97, 2011.

Nan, Y. and Ji, H. Deep learning for handling kernel/model uncertainty in image deconvolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2388–2397, 2020.

Nikolova, M. and Ng, M. K. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal on Scientific computing*, 27(3):937–966, 2005.

Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Revels, J., Lubin, M., and Papamarkou, T. Forward-mode automatic differentiation in julia. *arXiv preprint arXiv:1607.07892*, 2016.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. Plug-and-play priors for model based reconstruction. In

2013 IEEE Global Conference on Signal and Information Processing, pp. 945–948. IEEE, 2013.

Yang, X. and Wang, L. Fast half-quadratic algorithm for image restoration and reconstruction. *Applied Mathematical Modelling*, 50:92–104, 2017.

Zabihi Naeini, E. and Sams, M. Accuracy of wavelets, seismic inversion, and thin-bed resolution. *Interpretation*, 5(4):T523–T530, 2017.

A. Sensitivity of Loop Unrolling Algorithm to Uncertain $\mathcal{A}$

Figure 3 illustrates the sensitivity of LU to the forward model $\mathcal{A}$. The top row displays a sampled reconstruction process using an 8-iteration LU for deblurring with the precise forward model. As expected, the reconstruction quality is visually improving with more LU iterations $k = 1, \ldots, 8$. However, the bottom row shows a significant degradation in performance and poor final reconstruction $\boldsymbol{x}_8$ when a small perturbation is added to $\mathcal{A}$ during inference using the same network.

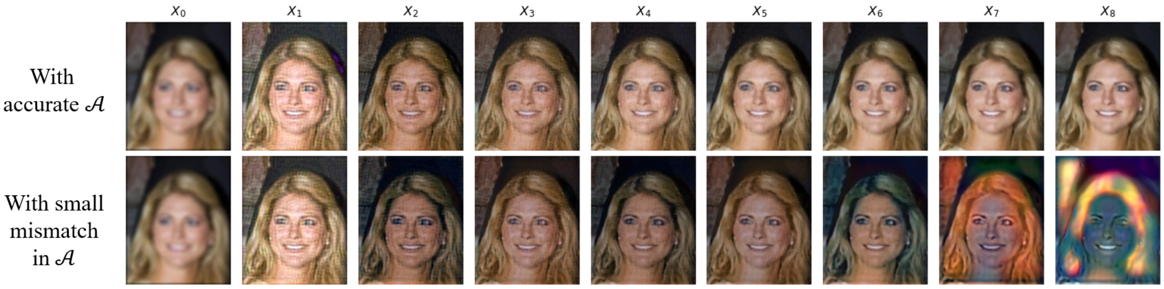
x_0
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
With accurate $\mathcal{A}$
With small mismatch in $\mathcal{A}$

Figure 3. A proximal LU network is trained for a deblurring task using a *single* forward model. The top row shows the intermediate reconstructions over 8 iterations using the true model, while the bottom row shows the evaluation results when a small perturbation is added to the forward model (the Peak Signal-to-Noise Ratio of the true kernel to the noisy kernel is 40.9 dB).

B. Training Details

The proximal network for LU and $\mathcal{A}$-adaptive LU uses DnCNN architecture with different numbers of layers for each task. Image deblurring uses 5-layer DnCNN and seismic deconvolution uses 9-layer DnCNN. The hidden feature is 64 for all applications.

The forward mismatch neural network f_θ and g_ψ incorporate the features in $\mathcal{A}_0$ differently for each task. The features of $\mathcal{A}_0$ are defined by the incorrect Gaussian kernel for the image deblurring task, defined by the incorrect 1D wavelet for the seismic deconvolution task, and defined by the depth profile of the defogging task. As illustrated in 4, The features of $\mathcal{A}_0$ are passed to a multi-layer perceptron (MLP) block and concatenated with $\mathcal{A}_0(\boldsymbol{x})$, and is then sent into the convolutional network (CNN), where it is added to $\mathcal{A}_0(\boldsymbol{x})$ to estimate $\boldsymbol{y}$. The CNNs are 3 layers with the hidden feature of 64 for all three tasks.

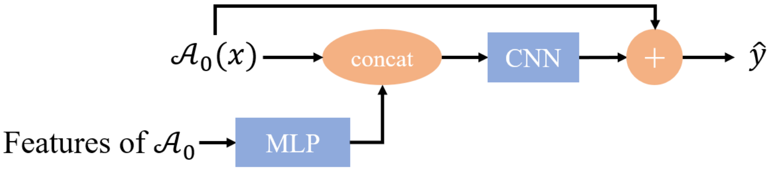

Figure 4. Architecture of the forward model mismatch network f_θ and g_ψ.

LU and $\mathcal{A}$-adaptive LU are both trained with a maximum of $K = 5$ iterations. For $\mathcal{A}$-adaptive LU, the hyper-parameters $\lambda = 0.01$ and $\tau = 0.1$ in (11) for both tasks. The learning rates used to update $\boldsymbol{z}$, $\boldsymbol{\theta}$ and the parameters in the proximal network were 0.0001. All networks were trained using the AdamW optimizer on an NVIDIA RTX 3080 with 10 GB of RAM.

C. Software

The GitHub link will be posted here if the paper is accepted.

D. Intermediate Reconstructions of $\mathcal{A}$-adaptive LU

We also explore the effectiveness of reconstruction by comparing the intermediate reconstruction $\boldsymbol{x}_k$'s in the proximal LU architecture and our proposed method in Figure 5. The average testing Mean-Squared Error (MSE) between $\boldsymbol{x}_k$ and the ground truth x is recorded in each $k = 1, \ldots, 5$. The intermediate MSE of the LU network remains high until the final iteration, indicating that the gradient update of the data fidelity term in (2) was not pointing in the correct directions due to the errors in $\mathcal{A}$. While LU provided some good results, the proximal operator was less interpretable and seemed to learn an unknown mapping from the erroneous gradient step to the true signal. In contrast, we found that an $\mathcal{A}$-adaptive LU resulted in a constant MSE decrement, gradually approaching the best estimate.

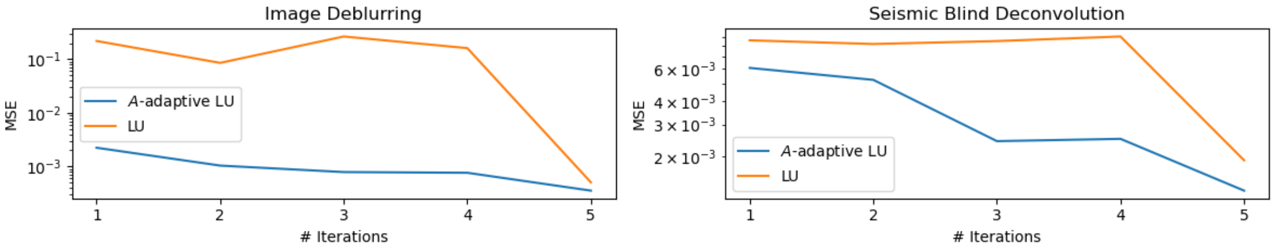

Figure 5. This figure depicts the average mean-squared error (MSE) of the intermediate reconstructions for the 5-iteration LU network and the $\mathcal{A}$-adaptive LU network. Comparing the mean-squared error (MSE) of the two approaches, it is apparent that the MSE of LU remains high until the final iteration, where it drops sharply. In contrast, $\mathcal{A}$-adaptive LU exhibits a more effective MSE decrement as it adapts the forward model along the way.

E. Broader Impact

In this work, a general inverse problem solver is proposed to deal with model mismatch. Each application has additional requirements for signal reconstruction. For example, in medical image reconstruction, particular attention needs to be paid to the accuracy of recovering specific tissue details. Task-specific designs need to be carefully considered on top of the proposed algorithm.