UTOPY: Unrolling Algorithm Learning via Fidelity Homotopy for Inverse Problems

Roman Jacome[†], Romario Gualdrón-Hurtado[‡], Leon Suarez-Rodriguez[‡] and Henry Arguello[‡]
[†]Department of Electrical, Electronics, and Telecommunications Engineering
[‡]Department of Systems Engineering and Informatics
Universidad Industrial de Santander, Colombia, 680002

***Abstract*—Inverse problems in imaging aim to reconstruct an underlying image from undersampled, coded, and noisy observations. Within the wide range of reconstruction frameworks, the unrolling algorithm is one of the most popular due to the synergistic integration of traditional model-based reconstruction methods and modern neural networks, providing an interpretable and highly accurate reconstruction. However, when the sensing operator is highly ill-conditioned, gradient steps on the data-fidelity term can hinder convergence and degrade reconstruction quality. To address this issue, we proposed a homotopy continuation formulation for training the unrolling algorithm. Mainly, this method involves using a better-conditioned (synthetic) sensing matrix at the beginning of the unrolling network optimization. We define a continuation path strategy to transition smoothly from the synthetic fidelity to the desired ill-posed problem. This allows the network to learn from an easier problem to the desired challenging one. We theoretically show that, for projected gradient descent-like unrolling models, the proposed continuation strategy generates a smooth path of unrolling solutions. Experiments on compressive sensing and image deblurring demonstrate that our method consistently surpasses conventional unrolled training, achieving higher reconstruction performance.**

***Index Terms*—Inverse problems, Unrolling algorithm, homotopy optimization, compressed sensing, image deblurring.**

I. Introduction

Inverse problems involve reconstructing an unknown signal from noisy, corrupted, or typically undersampled observations, making the recovery process generally non-invertible and ill-posed. This work focuses on linear inverse problems, where a sensing matrix represents the forward model citebert-ero1985linear. Numerous imaging tasks rely on these principles, including image restoration—such as deblurring, denoising, inpainting, and super-resolution [1]—as well as compressed sensing (CS) [2, 3] and medical imaging applications like magnetic resonance imaging (MRI) [4] or computed tomography (CT) [5]. See [6–8] and references therein for more applications of imaging inverse problems.

Algorithm unrolling offers an interpretable and effective framework for solving inverse problems by embedding neural network architectures within classical optimization procedures [9, 10]. Unrolled models replace fixed iterative schemes with learned proximal operators and data-driven parameters, yielding improved accuracy and computational efficiency in applications such as image restoration [11], MRI recovery [12], snapshot spectral imaging [13], graph signal processing [14], and phase retrieval [15], among others. Unrolling networks fall within the framework of learning-to-optimize [16], which automates the design of optimization methods based on their performance on a set of training problems. Unrolling approaches optimize against a fixed data fidelity objective, which can lead to convergence difficulties when the measurement operator is highly ill-posed or the inverse problem is severely underdetermined.

We propose a homotopy continuation formulation for unrolling algorithm optimization to address ill-posed data fidelity within the training. Homotopy continuation methods [17] offer a general strategy to mitigate such challenges by gradually transforming an easier optimization problem into the target problem. Related ideas appear in curriculum learning [18] and progressive training for generative models [19], where tasks are ordered to improve training dynamics. Recent work on learned proximal models [20] focuses on regularization design but does not incorporate a progressive fidelity path. In inverse problems, ℓ_1 -homotopy has been widely studied for CS, where a sequence of decreasing soft/hard thresholding parameters is used to approximate the sparse solution [21, 22] Homotopy-based methods have also been explored in physics-informed neural networks [23] leverage homotopy continuation to navigate multiple solution branches in nonlinear partial differential equations inverse problems. While conceptually related, such approaches have not yet been extended to algorithm unrolling for general inverse imaging problems.

This paper introduces a homotopy-based framework for unrolling network learning with significant advantages over traditional unrolled approaches. By incorporating a fidelity homotopy continuation path, our method systematically transitions from a simpler, well-conditioned problem towards the challenging target fidelity, enhancing convergence stability and generalization capability. This progressive strategy effectively mitigates common convergence issues in ill-posed inverse problems, thus improving solution accuracy and computational robustness. We provide theoretical guarantees on the smooth transition of the different solutions in the continuation path. The experiments show the efficacy of the proposed method for single-pixel imaging and image deblurring, validating the proposed framework. Although the method was validated with the FISTA unrolling formulation, it can be extended to other approaches, such as ADMM, HQS, or Dual-Primal formulations [9] and also deep equilibrium architectures [24].

II. Background

Unfolding algorithms offer an interpretable and effective approach for solving inverse problems by embedding classical algorithms within learnable neural networks [25]. Consider the inverse problem of reconstructing an image $\mathbf{x} \in \mathbb{R}^n$ from noisy and undersampled measurements $\mathbf{y} \in \mathbb{R}^m$ (with $m \ll n$), as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}, \tag{1}$$

where $\mathbf{H} \in \mathbb{R}^{m \times n}$ is the sensing matrix characterizing the acquisition system, and $\mathbf{e} \in \mathbb{R}^m$ is additive sensor noise. To estimate the original image $\mathbf{x}$ from measurements $\mathbf{y}$, variational formulations aim to solve the following optimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} g(\mathbf{x}) + \lambda h(\mathbf{x}), \tag{2}$$

where $g(\mathbf{x})$ is the data fidelity term, typically $g(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$, and $h(\mathbf{x})$ acts as a regularizer embedding prior knowledge about the image structure. The parameter $\lambda > 0$ controls the strength of the regularization. Common regularizers include the sparsity-inducing ℓ_1 norm [26], total variation (TV) [27], and smoothness-promoting Tikhonov [28]. In contrast, unfolding methods integrate iterative optimization algorithms directly into neural network architectures, where each iteration corresponds to a neural network layer with trainable parameters [9]. The proximal step in classical methods is replaced by a neural network to learn task-specific image priors from data effectively. Without loss of generality, we consider projected gradient descent-based unrolling with the fidelity term $g(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$, the unfolding step at iteration k can be formulated as:

$$\mathbf{x}^k = \mathcal{D}_{\theta^k}\left(\mathbf{x}^{k-1} - \tau^k \overbrace{\mathbf{H}^\top(\mathbf{H}\mathbf{x}^{k-1} - \mathbf{y})}^{\nabla g(\mathbf{x}^{k-1})}\right) := T_{\Omega^k}(\mathbf{x}^k), \tag{3}$$

where $\mathcal{D}_{\theta^k}$ represents a neural network parameterized by θ^k, $\tau^k > 0$ denotes the $k-th$ step size with K is the maximum number of iterations/stages and $\Omega^k = \{\theta^k, \tau^k\}$ are the $k-th$ trainable parameters of the unfolding algorithm with $\mathbf{x}^0 = \mathbf{H}^\top\mathbf{y}$. In practice, we add an acceleration step which improves convergence of the algorithm as the one in FISTA [29].

The unfolding algorithm is trained as follows:

$$\hat{\Omega} = \arg\min_{\theta} \sum_{p=1}^{P} \mathcal{L}_{\text{rec}}\left(\mathbf{x}_p, T_{\Omega^K}(\dots T_{\Omega^1}(\mathbf{H}^\top\mathbf{y}_p))\right), \tag{4}$$

where $\mathcal{L}_{\text{rec}}(\cdot,\cdot)$ is a reconstruction loss function such as MSE or ℓ_p-norm, $\mathbf{x}_p$ and $\mathbf{y}_p$ are the training pairs of ground-truth images and measurements, respectively. Training these unfolding architectures involves learning the parameters $\mathbf{\Omega} = \{\Omega_1, \dots, \Omega_K\}$, enabling the model to implicitly capture complex, data-driven priors tailored for specific inverse problems.

III. Homotopy Fidelity for Unrolling Algorithm

A progressive fidelity approach leverages a homotopy continuation method to effectively train unrolling algorithms by smoothly transitioning from an easier fidelity constraint to the challenging fidelity constraint. First, let denote $\mathbf{y}_t = \mathbf{H}_t\mathbf{x} \in \mathbb{R}^m_t$, $m_t \geq m$, a synthetic measurement set which is related to the original in Eq. (1) but is better-conditioned.

Algorithm 1 Fidelity Homotopy for Unrolling Algorithm

Require: $\mathbf{H}_t, \mathbf{H}, \mathbf{y}_t, \mathbf{y}$, `max_epochs`, ϵ, `freq`
Initialize network weights $\mathbf{\Omega} = \{\Omega_1, \dots, \Omega_K\}$
$\alpha = 1$
for $\ell = 1 :$ `max_epochs` **do**
  if $\ell \mod$ `freq` $= 0$, then update $\alpha =$ `scheduler`$_\epsilon(\ell)$,
  $\mathbf{x}_\alpha^0 = \mathbf{z}_\alpha = \alpha\mathbf{H}_t^\top\mathbf{y}_t + (1-\alpha)\mathbf{H}^\top\mathbf{y}$
  for $k = 1 : K$ **do**
    $\mathbf{x}_\alpha^{k+1} = \mathcal{D}_\theta^k(\mathbf{z}_\alpha - \tau_k\nabla_{\mathbf{x}} g_\alpha(\mathbf{z}_\alpha))$
    $\mathbf{z}_\alpha = \mathbf{x}_\alpha^{k+1} - t^k(\mathbf{x}_\alpha^{k+1} - \mathbf{x}_\alpha^k)$ ▷ Acceleration Step
  Compute loss $\mathcal{L} = \mathcal{L}_{\text{rec}}(\mathbf{x}, \mathbf{x}_\alpha^K)$
  Update weights $\theta = \theta - \zeta\nabla_\theta\mathcal{L}$

Note: During inference (testing), set $\alpha = 0$ exclusively.

We consider two $\mathbf{H}_t$ designs for the applications:

Compressed sensing Here we consider the case where $\mathbf{H} = \mathbf{P}\mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is an orthogonal transformation e.g., Hadamard or Fourier, and $\mathbf{P} \in \mathbb{R}^{m \times n}$ is a subsampling and permutation matrix that selects few rows of the transformation matrix. Thus, we consider $\mathbf{H}_t = \mathbf{P}_t\mathbf{W}$ where $\mathbf{P}_t \in \mathbb{R}^{m_t \times n}$, selects more rows from $\mathbf{W}$ than $\mathbf{P}$. In this setting, we define $\eta = 1 - m/n$ as the augmented ratio. This is a hyperparameter of the method.

Image Deblurring The forward model $\mathbf{H}$ is built upon a Toeplitz matrix based on the convolution kernel denoted as $\mathbf{H}[i, i+j] = \mathbf{h}[j]$ with $i = 1, \dots, m$ and $j = 1, \dots, n$, where $\mathbf{h}$ is the vectorized Gaussian kernel with σ standard deviation. Thus, we considered $\mathbf{H}_t[i, i+j] = \mathbf{h}_t[j]$ with the kernel $\mathbf{h}_t$ has a standard deviation $\sigma_t < \sigma$. Note that in this case $m_t = m$, which depends on the employed padding scheme. The implementation contains zero-padding such that $n = m_t = m$.

Define the family of fidelity functions $g_\alpha(\mathbf{x}) = (1-\alpha)\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \alpha\|\mathbf{y}_t - \mathbf{H}_t\mathbf{x}\|_2^2$, where α is a homotopy parameter controlling the transition from a simpler fidelity $(\mathbf{H}_t, \mathbf{y}_t)$ towards the target fidelity $(\mathbf{H}, \mathbf{y})$. The continuation path is designed so that $\alpha : 1 \to 0$ during the training. Then, the fidelity homotopy optimization is expressed as follows:

$$\hat{\mathbf{x}}_\alpha = \arg\min_{\mathbf{x}} g_\alpha(\mathbf{x}) + \lambda h(\mathbf{x}), \tag{5}$$

The unrolling network for this optimization problem becomes

$$\mathbf{x}_\alpha^k = \mathcal{D}_{\theta^k}\left(\mathbf{x}_\alpha^{k-1} - \tau^k\nabla g_\alpha(\mathbf{x}_\alpha^{k-1})\right), \text{ where} \tag{6}$$

$$\nabla g_\alpha(\mathbf{x}) = \frac{1}{2}\alpha\mathbf{H}_t^\top(\mathbf{H}_t\mathbf{x} - \mathbf{y}_t) + \frac{1}{2}(1-\alpha)\mathbf{H}^\top(\mathbf{H}\mathbf{x} - \mathbf{y}). \tag{7}$$

Note that this gradient is well-conditioned for $\alpha = 1$.

A. Fidelity Homotopy for Unrolling Algorithm Optimization

Algorithm 1 details the fidelity homotopy, starting with standard initialization of network parameters $\mathbf{\Omega}$, typically set via standard initialization schemes. Parameter α is initialized to 1, indicating the initial focus is on the simpler fidelity term ($\mathbf{H}_t$, $\mathbf{y}_t$), with a better-conditioned measurement operator. At each

epoch ℓ, the value of α is updated following $\texttt{scheduler}_\epsilon(\ell)$. Particularly, we used two types of schedulers:

$$\texttt{scheduler}_\epsilon(\ell) = e^{\gamma\ell}, \gamma = \frac{\log(\epsilon)}{\texttt{max\_epochs}}, \quad \text{(exponential)}$$

$$\texttt{scheduler}_\epsilon(\ell) = \epsilon\alpha, \quad \text{(linear)}$$

where $\epsilon > 0$ is a parameter controlling the speed of the decreasing values of α. The initial estimate for the image at each epoch, $\mathbf{x}_\alpha^0$, is computed as a convex combination of adjoint reconstructions from both fidelity terms: $\mathbf{x}_\alpha^0 = \alpha\mathbf{H}_t^\top\mathbf{y}_t + (1-\alpha)\mathbf{H}^\top\mathbf{y}$. In the inner loop, the algorithm updates the reconstruction iteratively through learned proximal operations. Finally, during inference, the parameter α is fixed to 0, ensuring the neural network only uses the target fidelity.

B. Theoretical Analysis

We make the following assumptions: **(A1)** ∇g_α is L_α-Lipschitz and continuously differentiable in $(\mathbf{x}, \alpha)$ with $L = \sup_\alpha L_\alpha < \infty$. **(A2)** Each g_α is μ_α-strongly convex with $\mu = \inf_\alpha \mu_\alpha > 0$. **(A3)** Contractive neural proximal. The denoiser $\mathcal{D}_\theta : \mathbb{R}^n \to \mathbb{R}^n$ is built from affine layers and pointwise $\mathcal{C}^1$ activations (e.g. Swish, GELU, soft-ReLU). The chain rule implies $\mathcal{D}_\theta$ itself is differentiable everywhere. Moreover, the network $\mathcal{D}_\theta$ is β-Lipschitz with $\beta < 1$ implying

$$\|\mathcal{D}_\theta(\mathbf{u}) - \mathcal{D}_\theta(\mathbf{v})\|_2 \le \beta\|\mathbf{u} - \mathbf{v}\|_2,$$

With the step size $0 < \tau < (1-\beta)/L$, the unrolled operator is

$$T_{\alpha,\theta}(\mathbf{x}) = \mathcal{D}_\theta\big(\mathbf{x} - \tau\nabla g_\alpha(\mathbf{x})\big). \tag{8}$$

Theorem 1 (Smooth path of unrolled solutions)**.** *Under **A1–A3** the fixed-point equation $\mathbf{x} = T_{\alpha,\theta}(\mathbf{x})$ admits a unique solution $\widehat{\mathbf{x}}_\alpha = \mathrm{fix}(T_{\alpha,\theta})$ for every $\alpha \in [0,1]$. The mapping $\alpha \mapsto \widehat{\mathbf{x}}_\alpha$ lies in $\mathcal{C}^1[0,1]$ and obeys*

$$\|\widehat{\mathbf{x}}_{\alpha_1} - \widehat{\mathbf{x}}_{\alpha_2}\|_2 \le \frac{L}{1-\beta-\tau L}\,|\alpha_1 - \alpha_2|, \quad \forall\,\alpha_1, \alpha_2 \in [0,1]. \tag{9}$$

Proof. For any $\mathbf{u}, \mathbf{v}$ the β-Lipschitz property and Eq. (8) give

$$\begin{aligned}\|T_{\alpha,\theta}(\mathbf{u}) - T_{\alpha,\theta}(\mathbf{v})\|_2 &\le \beta\|\mathbf{u}-\mathbf{v}\|_2 + \beta\tau L\|\mathbf{u}-\mathbf{v}\|_2 \\ &= (\beta + \tau L)\|\mathbf{u}-\mathbf{v}\|_2.\end{aligned}$$

Because $\tau < (1-\beta)/L$, the contraction factor $\rho = \beta + \tau L$ satisfies $\rho < 1$. Now, let $G(\mathbf{x}, \alpha) = \mathbf{x} - T_{\alpha,\theta}(\mathbf{x})$. Since $\mathcal{D}_\theta$ and ∇g_α are $\mathcal{C}^1$, so is G. The corresponding Jacobian in $\mathbf{x}$ is

$$J_\mathbf{x} G = \mathbf{I} - \mathbf{J}_{\mathcal{D}_\theta}\big(\mathbf{I} - \tau\nabla^2 g_\alpha\big),$$

whose smallest singular value is at least $1 - \beta - \tau L > 0$; hence $J_\mathbf{x} G$ is invertible. The Implicit Function Theorem then yields a unique $\mathcal{C}^1$ curve $\alpha \mapsto \widehat{\mathbf{x}}_\alpha$ satisfying $G(\widehat{\mathbf{x}}_\alpha, \alpha) = 0$. Then, differentiating $G(\widehat{\mathbf{x}}_\alpha, \alpha) = 0$ with respect to α gives the linear relation $J_\mathbf{x} G\,\dot{\widehat{\mathbf{x}}}_\alpha + \partial_\alpha G = 0$. Taking norms and using $\|J_\mathbf{x} G^{-1}\|_2 \le (1-\beta-\tau L)^{-1}$ and $\|\partial_\alpha G\|_2 \le \tau L$ yields $\|\dot{\widehat{\mathbf{x}}}_\alpha\|_2 \le \tau L/(1-\beta-\tau L)$. Finally, for any $\alpha_1, \alpha_2 \in [0,1]$ apply the Fundamental Theorem of Calculus:

$$\begin{aligned}\|\widehat{\mathbf{x}}_{\alpha_1} - \widehat{\mathbf{x}}_{\alpha_2}\|_2 &= \Big\|\int_{\alpha_2}^{\alpha_1} \dot{\widehat{\mathbf{x}}}_\tau\, d\alpha\Big\|_2 \le \int_{\alpha_2}^{\alpha_1} \frac{\tau L}{1-\beta-\tau L} d\alpha \\ &= \frac{\tau L}{1-\beta-\tau L}\,|\alpha_1 - \alpha_2|.\end{aligned}$$

leads directly to the global bound Eq. (9). □

Remark 1. *Starting at $\alpha = 1$ and following the continuation path given by the function $\texttt{scheduler}_\epsilon(\ell)$. Standard results on perturbed gradient flows imply the iterates of the unrolled network track $\widehat{\mathbf{x}}_\alpha$ with error $\mathcal{O}\big((L/\mu)\gamma\big)$. This relates to curriculum-learning theory [18], where the optimizer first solves an easier problem before gradually tackling the full-fidelity objective, thereby avoiding local minima.*

IV. Results

The framework was implemented in PyTorch. For the network training, we used the CelebA dataset [30], using 24.318 images for training and 2.993 for testing. All images were resized to 64×64, thus $n = 4096$. During training, additive white Gaussian noise was added to the measurements with SNR = 35[dB]. We used the Adam optimizer [31] with a learning rate $\zeta = 1e^{-5}$, which was halved every three times during the network training. We used a composite loss function

$$\begin{aligned}\mathcal{L}_{\texttt{rec}}(\mathbf{a}, \mathbf{b}) = 0.8 \times \|\mathbf{a} - \mathbf{b}\|_1 + 0.2 \times (1 - \texttt{SSIM}(\mathbf{a}, \mathbf{b})) \\ + 0.02 \times \|\mathbf{w} \odot (|\mathbf{F}\mathbf{a} - \mathbf{F}\mathbf{b}|)\|,\end{aligned}$$

where $\mathbf{F}$ is the Fourier transform and $\mathbf{w}$ is a circular window, band passing the high frequencies. This loss function is inspired by the focal frequency loss [32] and the loss function proposed in [33]. The weights of each term were selected by grid search on the baseline training. For the implementation, we followed the library `pycolibri`[1]. The baseline consisted of the classical training of unrolling networks, i.e., $\alpha = 0 \forall \ell$. The network $\mathcal{D}_{\theta^k}$ was a UNet model with features following $[32, 64, 128, 256]$, and each level has a double block convolution, batch normalization, and ReLU. We used $K = 5$ unrolling stages, the parameters were initialized $\tau^k = 1e^{-3} \forall k$. The acceleration parameter t^k of Algorithm 1 is trainable.

A. Compressed sensing experiments:

To validate the proposed approach in single-pixel imaging [34, 35] using Hadamard coded apertures using the implementation of [36]. We set the compression ratio for all experiments, i.e., $m/n = 0.3$. We used $\texttt{max\_epochs} = 500$ and batch size of 32. Fig. 1(a) shows both schedulers, which maintain $\alpha = 1.0$ during the initial 10 epochs for the network first learns the well-conditioned fidelity $(\mathbf{H}_t, \mathbf{y}_t)$ and then decrease to $\alpha = 0$ at epoch $\ell = 350$, at which point training focuses exclusively on the objective fidelity $(\mathbf{H}, \mathbf{y})$. Fig. 1(b) reports the training PSNR averaged: early in training, all homotopy variants quickly gain PSNR by solving the simpler $(\mathbf{H}_t, \mathbf{y}_t)$ task. As $\alpha \to 0$, the PSNR temporarily declines, which reflects the shift to the ill-posed fidelity, before recovering, demonstrating effective transition. The baseline (constant $\alpha = 0$) shows faster convergence but lower performance. Fig. 1(c) presents the testing PSNR on true measurements $\mathbf{y}$: the baseline converges more rapidly due to its constant focus on the target fidelity, but the proposed homotopy-trained models achieve higher final PSNR across all η values and both scheduler types,

[1] https://pycolibri.github.io

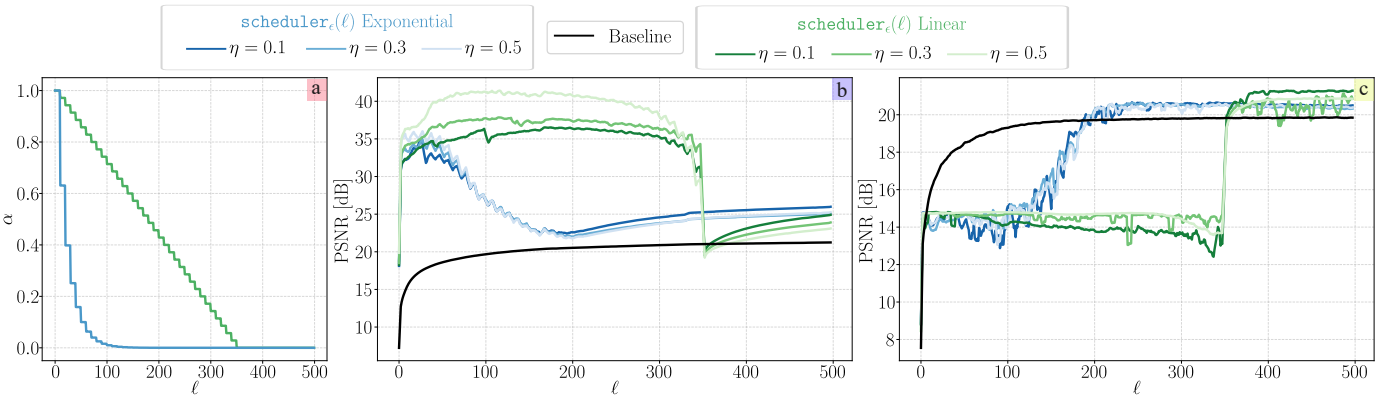

Figure 1: (a) Exponential and linear homotopy-continuation schedules $\alpha(\ell)$ over training epochs, holding $\alpha = 1.0$ for the first 10 epochs and decaying to $\alpha = 0.0$ at $\ell = 350$. (b) Training PSNR per epoch for each scheduler and η, compared against the baseline unrolled network trained with constant $\alpha = 0$. (c) Testing PSNR on the target fidelity $(\mathbf{H}, \mathbf{y})$: although the baseline converges fastest, the proposed homotopy method surpasses the baseline's final PSNR in all cases and for both schedulers.

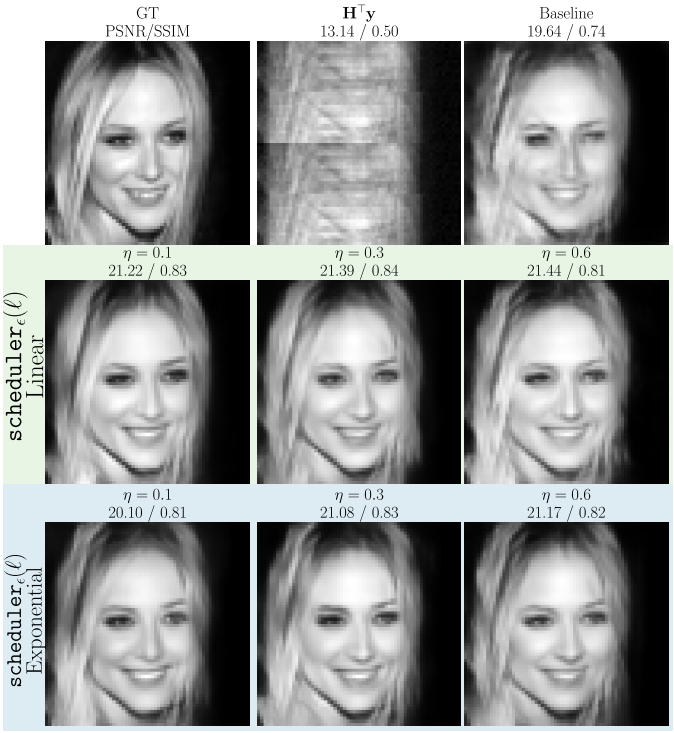

Figure 2: Visual Reconstruction for SPC experiment with different schedulers and baseline unrolling model

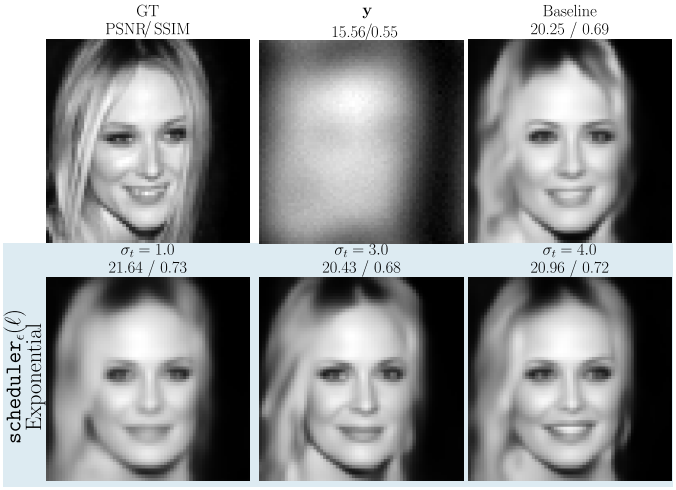

Figure 3: Visual Reconstruction for image deblurring experiment for UTOPY with exponential scheduler and baseline unrolling model

Inverse Problem	Testing Setting	Baseline		UTOPY		FISTA-PnP	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CS	$m/n = 0.25$	14.97	0.525	**16.78**	**0.620**	14.55	0.525
	$m/n = 0.30$	19.75	0.706	**21.19**	**0.771**	15.20	0.493
	$m/n = 0.35$	19.93	0.720	**21.40**	**0.784**	15.37	0.510
Deblurring	SNR = 30	20.84	0.658	**21.54**	**0.693**	19.85	0.596
	SNR = 35	21.17	0.660	**22.17**	**0.718**	19.906	0.598
	SNR = 40	21.12	0.668	**22.38**	**0.726**	19.92	0.598

Table I: Unrolling baseline, UTOPY, and FISTA-PnP performance across testing settings for each inverse problem. Setting using in training. Best performance for each testing setting in **bold teal**

confirming that progressive fidelity continuation yields superior generalization to the inverse problem of interest. In Fig. 3, where proposed method with both schedulers a different settings of η

V. Conclusion

References

[1] B. Gunturk and X. Li, Image restoration. CRC Press, 2018.

[2] Z. Zha, B. Wen, X. Yuan, S. Ravishankar, J. Zhou, and C. Zhu, "Learning nonlocal sparse and low-rank models for image compressive sensing: Nonlocal sparse and low-rank modeling," IEEE Signal Processing Magazine, vol. 40, no. 1, pp. 32–44, 2023.

[3] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling,"

IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 21–30, 2008.
[4] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing mri," IEEE signal processing magazine, vol. 25, no. 2, pp. 72–82, 2008.
[5] M. J. Willemink, P. A. de Jong, T. Leiner, L. M. de Heer, R. A. Nievelstein, R. P. Budde, and A. M. Schilham, "Iterative reconstruction techniques for computed tomography part 1: technical principles," European radiology, vol. 23, pp. 1623–1631, 2013.
[6] G. Ongie, A. Jalal, C. A. M. R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," IEEE Journal on Selected Areas in Information Theory, 2020.
[7] Y. Bai, W. Chen, J. Chen, and W. Guo, "Deep learning methods for solving linear inverse problems: Research directions and paradigms," Signal Processing, vol. 177, p. 107729, 2020.
[8] M. Bertero, P. Boccacci, and C. De Mol, Introduction to inverse problems in imaging. CRC press, 2021.
[9] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," IEEE Signal Processing Magazine, vol. 38, no. 2, pp. 18–44, 2021.
[10] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in Proceedings of the 27th international conference on international conference on machine learning, 2010, pp. 399–406.
[11] C. Mou, Q. Wang, and J. Zhang, "Deep generalized unfolding networks for image restoration," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 17 399–17 410.
[12] H. K. Aggarwal, M. P. Mani, and M. Jacob, "Modl: Model-based deep learning architecture for inverse problems," IEEE transactions on medical imaging, vol. 38, no. 2, pp. 394–405, 2018.
[13] R. Jacome, J. Bacca, and H. Arguello, "D 2 uf: Deep coded aperture design and unrolling algorithm for compressive spectral image fusion," IEEE Journal of Selected Topics in Signal Processing, 2022.
[14] B. Li, G. Verma, and S. Segarra, "Graph-based algorithm unfolding for energy-aware power allocation in wireless networks," IEEE Transactions on Wireless Communications, vol. 22, no. 2, pp. 1359–1373, 2022.
[15] S. Pinilla, K. V. Mishra, I. Shevkunov, M. Soltanalian, V. Katkovnik, and K. Egiazarian, "Unfolding-aided bootstrapped phase retrieval in optical imaging: Explainable ai reveals new imaging frontiers," IEEE Signal Processing Magazine, vol. 40, no. 2, pp. 46–60, 2023.
[16] T. Chen, X. Chen, W. Chen, H. Heaton, J. Liu, Z. Wang, and W. Yin, "Learning to optimize: A primer and a benchmark," Journal of Machine Learning Research, vol. 23, no. 189, pp. 1–59, 2022.
[17] E. L. Allgower and K. Georg, Introduction to numerical continuation methods. SIAM, 2003.
[18] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in Proceedings of the 26th Annual International Conference on Machine Learning (ICML), 2009, pp. 41–48.
[19] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in International Conference on Learning Representations (ICLR), 2018.
[20] J. Adler and O. Öktem, "Learned primal-dual reconstruction," in International Conference on Learning Representations (ICLR), 2018.
[21] L. Zhang, T. Yang, R. Jin, and Z.-H. Zhou, "A simple homotopy algorithm for compressive sensing," in Artificial Intelligence and Statistics. PMLR, 2015, pp. 1116–1124.
[22] Z. Dong and W. Zhu, "Homotopy methods based on $l\_\{0\}$-norm for compressed sensing," IEEE transactions on neural networks and learning systems, vol. 29, no. 4, pp. 1132–1146, 2017.
[23] H. Zheng, Y. Huang, Z. Huang, W. Hao, and G. Lin, "Hompinns: homotopy physics-informed neural networks for solving the inverse problems of nonlinear differential equations with multiple solutions," Journal of Computational Physics, 2024.
[24] 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.
[25] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," IEEE Signal Processing Magazine, vol. 38, no. 2, pp. 18–44, 2021.
[26] M. Schmidt, "Least squares optimization with l1-norm regularization," CS542B Project Report, vol. 504, no. 2005, pp. 195–221, 2005.
[27] X. Yuan, "Generalized alternating projection based total variation minimization for compressive sensing," in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 2539–2543.
[28] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," SIAM journal on matrix analysis and applications, vol. 21, no. 1, pp. 185–194, 1999.
[29] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," SIAM journal on imaging sciences, vol. 2, no. 1, pp. 183–202, 2009.
[30] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in Proceedings of International Conference on Computer Vision (ICCV), December 2015.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[32] L. Jiang, B. Dai, W. Wu, and C. C. Loy, "Focal frequency loss for image reconstruction and synthesis," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 13 919–13 929.
[33] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47–57, 2017.
[34] X. Hu, H. Zhang, Q. Zhao, P. Yu, Y. Li, and L. Gong, "Single-pixel phase imaging by fourier spectrum sampling," Applied Physics Letters, vol. 114, no. 5, p. 051102, 2019.
[35] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," IEEE signal processing magazine, vol. 25, no. 2, pp. 83–91, 2008.
[36] B. Monroy and J. Bacca, "Hadamard row-wise generation algorithm," CoRR, 2024.