Deep Memory Unrolled Networks for Solving Imaging Linear Inverse Problems

Eric Chen*, Xi Chen†, Shirin Jalali† and Arian Maleki*
*Department of Statistics, Columbia University, New York, USA
†Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, USA
Email: yc4041@columbia.edu, xi.chen15@rutgers.edu, shirin.jalali@rutgers.edu, arian@stat.columbia.edu

***Abstract*—Unrolled networks have emerged as one of the most successful methods in imaging applications. Although they have demonstrated remarkable efficacy in solving specific computer vision and computational imaging tasks, their adaptation to other applications presents considerable challenges. This is primarily due to the multitude of design decisions that practitioners working on new applications must navigate, each potentially affecting the network's overall performance. These decisions include selecting the optimization algorithm to unroll, defining the loss function, deciding on the structure of residual connections, and determining the number of convolutional layers, among others. Compounding the issue, evaluating each design choice requires time-consuming simulations to train the neural network. As a result, the process of exploring multiple options and identifying the optimal configuration becomes time-consuming and computationally demanding. The main objectives of this paper are (1) to unify some ideas and methodologies used in unrolled networks to reduce the number of design choices a user has to make, and (2) to report a comprehensive numerical study to discover the optimal design choices. We anticipate that this study will help scientists and engineers design unrolled networks for their applications and diagnose problems within their networks efficiently.**

***Index Terms*—Compressed Sensing, Unrolled Networks, Computational Imaging.**

I. Introduction

A. Unrolled Networks for Imaging Linear Inverse Problems

In many imaging applications, ranging from magnetic resonance imaging (MRI) to seismic imaging and nuclear magnetic resonance (NMR), the measurement process can be modeled in the following way:

$$y = Ax^* + w,$$

where $y \in \mathbb{R}^m$ represents the collected measurements, $x^* \in \mathbb{R}^n$ denotes the vectorized image that we aim to capture, $A \in \mathbb{R}^{m\times n}$ represents the forward operator or measurement matrix of the imaging system (either known exactly or with some small error), and w represents the measurement noise, which is not known, but some information about its statistical properties (such as the approximate shape of the distribution) may be available.

Recovering x^* from y has been extensively studied in the last 15 years since the emergence of compressed sensing, with many successful algorithms proposed [1–10]. More recently, *unrolled networks* have emerged as a successful direction for solving inverse problems [11–14]. To motivate unrolled networks, consider the hypothetical situation in which images of interest belong to a set $\mathcal{C} \subset \mathbb{R}^n$. Under this assumption, one way to recover the image x^* from y is to find

$$\arg\min_{x\in\mathcal{C}} \|y - Ax\|_2^2.$$

One classical approach to solving this optimization problem is projected gradient descent (PGD), which iteratively performs

$$\begin{aligned} \tilde{x}^i &= x^i + \mu A^T(y - Ax^i), \\ x^{i+1} &= P_{\mathcal{C}}(\tilde{x}^i). \end{aligned} \tag{1}$$

Here x^i is the estimate of x^* in iteration i, μ is the step size and $P_{\mathcal{C}}$ denotes the projection onto the set $\mathcal{C}$. Figure 1 shows the graph of the projected gradient descent algorithm. Since the set $\mathcal{C}$ is not accessible in real-world applications, unrolled networks have been developed to replace the projection operator $P_{\mathcal{C}}$ with neural networks, allowing these projectors to be learned directly from training data. Unrolled networks can also be constructed using optimization algorithms other than PGD, including the Alternating Direction Method of Multipliers (ADMM) [15], Nesterov's Accelerated First-Order Method [16, 17], and Approximate Message Passing (AMP) [18]. Many of these alternatives offer faster convergence for *convex* optimization problems, raising the prospect of reducing the number of neural networks projectors required. It should be noted that, given the complex landscape of training loss in neural networks, it remains unclear whether these alternatives actually improve the performance of unrolled networks.

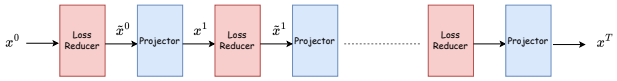

Fig. 1: Diagram of projected gradient descent. Starting with $x^0 = 0$, the i^{th} loss reducer performs the operation $\tilde{x}^i = x^i + \mu A^T(y - Ax^i)$, and the i^{th} projection performs $x^{i+1} = P_{\mathcal{C}}(\tilde{x}^i)$.

B. Challenges in Using Unrolled Networks

The adaptation of unrolled networks to new applications presents considerable challenges. These difficulties stem primarily from two factors: (1) the multitude of design choices practitioners must navigate, and (2) the computational cost of evaluating different configurations. Our main objective is to significantly simplify the design process of unrolled networks.

Before introducing our approach, we highlight several key design decisions that critically impact recovery performance, which we study in this paper.

1) Choice of Optimization Algorithm: As mentioned above, the first fundamental decision is selecting the optimization algorithm to unroll. Although practitioners often default to simpler algorithms like PGD, there exist many alternatives including ADMM, Nesterov's method, and AMP. Each algorithm presents different trade-offs in terms of convergence speed and computational complexity for convex optimization problems. The relative benefits of these choices remain unclear in the context of unrolled networks, leaving scientists and engineers uncertain about which option is best.

2) Choice of Loss Function: Conventionally, unrolled networks are trained using only the final output x^T through the mean squared error loss (called the last layer loss throughout this paper):

$$\ell_{ll}(x^T, x^*) = \|x^T - x^*\|_2^2$$

where x^T is the network's final output and x^* is the ground truth. While this approach seems intuitive since our primary focus is on final reconstruction quality, the nonconvexity of the training landscape raises uncertainty about whether optimizing only the final output truly leads to the best possible reconstruction. Inspired by the successes in image classification [19], one can also consider alternative loss functions that incorporate intermediate supervision such as:

- *Weighted Intermediate Loss with weight* $\omega \in (0, 1]$:

$$\ell_{i,\omega}(x^1, x^2, \dots, x^T, x^*) = \sum_{i=1}^{T} \omega^{T-i} \|x^i - x^*\|_2^2,$$

- *Skip-L-Layer Loss (L is a factor of T)*:

$$\ell_{s,L}(x^1, x^2, \dots, x^T, x^*) = \sum_{i=0}^{T/L-1} \|x^{T-iL} - x^*\|_2^2.$$

While similar intermediate loss strategies have shown promise in image restoration and super-resolution tasks [20, 21], their benefits for solving inverse problems have not been explored. Hence, the best choice of the loss function has remained unclear for practitioners.

3) Number of Unrolled Steps: Practitioners also have to pick the number of steps T to unroll an optimization algorithm. Increasing T often comes with additional computational burdens and may also lead to overfitting. A well-chosen value of T can allow efficient training while maintaining strong performance.

4) Complexity of the Neural Network: Similar to the above, the choice of neural network for replacing $P_{\mathcal{C}}$ also has a significant impact on the performance of the network. The options entail the number of layers or depth of the network, whether or not to include residual connections, etc. If the projector is designed to have only little complexity, the unrolled network may have poor recovery. However, if the projector has excessive complexity, the network may become computationally expensive to train and prone to overfitting. Again, practitioners have to make multiple decisions regarding the choice of the projector.

II. Our Approach and Deep Memory Unrolled Networks

To address the multitude of design choices practitioners face, we propose a two-stage approach. First, we develop a novel neural network architecture termed Deep Memory Unrolled Network (DeMUN). DeMUN includes various optimization algorithms such as PGD and AMP as special cases and uses data to automatically learn the "optimal" algorithm to unroll. This approach eliminates the need to manually select an optimization algorithm to unroll. Second, through extensive empirical studies, we investigate the impact of different loss functions, residual connections, network depth, and network complexity to provide practical guidelines for implementing these networks.

A. Deep Memory Unrolled Network (DeMUN)

While traditional unrolled algorithms rely on a fixed optimization algorithm such as PGD or AMP, we propose the Deep Memory Unrolled Network (DeMUN), a general framework that learns to adaptively combine the gradient information of all previous iterations. At the i-th iteration in DeMUN, the update of $\tilde{x}^i$ is given by:

$$\tilde{x}^i = \alpha^i x^i + \sum_{j=0}^{i} \beta_j^i A^T(y - Ax^j),$$

for $i \in \{0, \dots, T-1\}$ where $x^0 = 0$. This update is exhibited in Figure 2. Note that since the gradient information was computed in the previous iteration, DeMUN does not require any additional matrix vector multiplication compared to, for instance, PGD. Unlike PGD in (1), DeMUN retains the full optimization gradient history and discovers the optimal combination by learning the weights $\{\beta_j^i\}_j$.

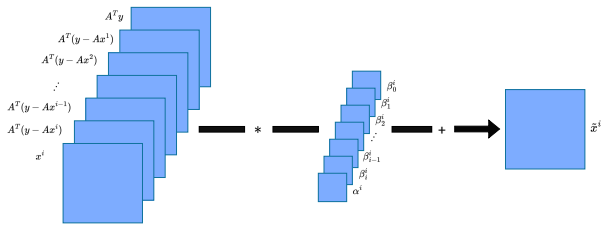

Fig. 2: The reconstructed image $\tilde{x}^i$ is updated by the weighted sum of memory terms $\{A^T(y - Ax^j)\}_{j=0}^i$ and image at i-th iteration x^i with learnable coefficients $\{\beta_j^i\}_{j=0}^i$ and α^i.

B. Simulation Setup

As previously mentioned, the rest of the paper presents extensive simulation results evaluating key design choices, such as the loss function, number of unrolled steps, and network complexity. Due to space constraints, we present only a subset of our simulations here. For more extensive results, please refer to the extended version of our paper [22]. For

all simulations reported here, we consider only two sampling rates m/n for the measurement matrix A: 20% and 40%. Each entry in the measurement matrix is i.i.d. Gaussian, where $A_{ij} \sim \mathcal{N}(0, 1/m)$. All training images have resolution 50×50. In addition, we first consider two settings for the number of unrolled steps T: 15 and 30. For all results below, we report the Peak Signal-to-Noise Ratio (PSNR) on a test set of 2500 images. Please also see our extended paper [22] for further details on data collection and processing, training of unrolled networks, and their evaluation.

In our simulations, we adopt the DnCNN architecture outlined by Zhang et al. [23] as our neural network projectors $P_{\mathcal{C}}$. DnCNN with L intermediate layers consists of an input layer with 64 filters of size $3 \times 3 \times 1$ followed by a ReLU activation function to map the input image to 64 channels, L layers consisting of 64 filters of size $3 \times 3 \times 64$ followed by Batch Normalization and ReLU, and a final reconstruction layer with a single filter of size $3 \times 3 \times 64$ to map to the output dimension of $50 \times 50 \times 1$. We use $L = 5$ in our initial simulations, and discuss the impact of L in Section III-D.

III. Impact of Our Design Choices

In this section, we first demonstrate that DeMUNs offer superior recovery performance compared to other unrolled algorithms. We then show that unrolled networks trained with the intermediate loss function $\ell_{i,1}$ consistently outperform their counterparts trained with the other loss functions. Finally, we examine the other design choices.

A. DeMUNs and Intermediate Loss

We first compare the performance of four unrolled algorithms trained with the last-layer loss ℓ_{ll} against their counterparts trained with the intermediate loss $\ell_{i,1}$.

1) Deep Memory Unrolled Network (DeMUN).
2) Projected Gradient Descent (PGD) as outlined in (1).
3) Nesterov's First-Order Method (Nesterov): an optimization method that uses memory to accelerate convergence [16].
4) Approximate Message Passing (AMP): an iterative algorithm tailored for Gaussian sensing matrices [6, 24].

m	DeMUN	PGD	Nesterov	AMP
$0.2n$	27.23	27.71	26.36	23.43
$0.4n$	30.20	30.53	26.47	24.71

Last Layer Loss ℓ_{ll}

m	DeMUN	PGD	Nesterov	AMP
$0.2n$	29.97	28.97	28.39	29.11
$0.4n$	33.68	32.52	31.47	32.72

Intermediate Loss $\ell_{i,1}$

TABLE I: Average Test PSNR (dB) Under **15** Projections

Tables I & II and Figures 3 & 4 reveal two key findings:

- *Improved Performance with Intermediate Loss.* Training with the intermediate loss function consistently improves reconstruction quality across all algorithms and sampling rates. For instance, at $0.4n$ sampling rate with 15 projections, DeMUN improves from 30.20dB to 33.68dB when switching from last layer to intermediate loss. Similar improvements are observed with 30 projections from 31.31dB to 34.05dB. As shown in Figures 3 and 4, intermediate loss enables each projection step to contribute meaningfully to reconstruction quality, whereas the last-layer loss leads to stagnation in early iterations.
- *Effectiveness of DeMUN.* Among all algorithms tested, DeMUN achieves the highest PSNR values when trained with intermediate loss. At $0.4n$ sampling rate with 30 projections, DeMUN achieves 34.05dB, outperforming all other unrolled algorithms. However, we note that this advantage diminishes with the last layer loss. This might be due to the fact that memory networks with their larger parameter space may get stuck in local minima. Intermediate loss mitigates this by optimizing reconstruction performance across all steps, especially benefiting the earlier layers.
- *Other Loss Functions*: Due to space constraints, we omit results for other loss functions mentioned in Section I-B2, but our extended paper [22] confirms that none of the loss

m	DeMUN	PGD	Nesterov	AMP
$0.2n$	26.37	27.30	27.00	19.99
$0.4n$	31.31	30.33	29.95	22.66

Last Layer Loss ℓ_{ll}

m	DeMUN	PGD	Nesterov	AMP
$0.2n$	29.86	29.07	28.35	29.12
$0.4n$	34.05	32.33	31.15	32.87

Intermediate Loss $\ell_{i,1}$

TABLE II: Average Test PSNR (dB) Under **30** Projections

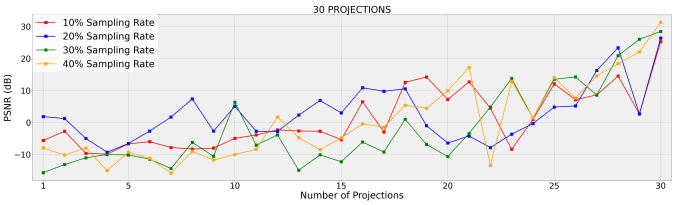

Fig. 3: DeMUN trained with loss ℓ_{ll} and $T = 30$ projections. The plot displays the PSNR after each intermediate projection.

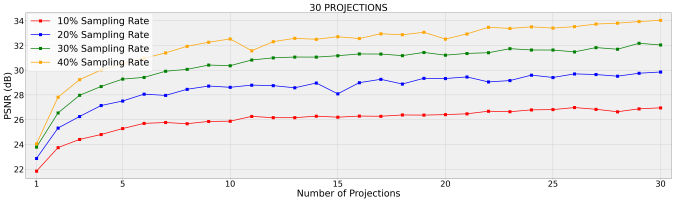

Fig. 4: DeMUN trained with loss $\ell_{i,1}$ and $T = 30$ projections. The plot displays the PSNR after each intermediate projection.

functions mentioned in Section I-B2 outperforms $\ell_{i,1}$.

B. Residual Connections

Next, we examine the impact of including residual connections in the form $x^{i+1} = P_{\mathcal{C}}(\tilde{x}_i) + \tilde{x}_i$. Residual connections are known to alleviate issues such as vanishing gradients and facilitate the training of deeper networks by allowing gradients to propagate more effectively through the intermediate layers [25, 26]. Here, we focus our comparison between DeMUN and PGD to evaluate this architectural choice.

m	Algorithm	**15** Steps	**30** Steps
$0.2n$	DeMUN	29.46	29.39
	PGD	29.33	29.27
$0.4n$	DeMUN	33.09	32.89
	PGD	32.70	32.79

TABLE III: Average PSNR (dB) with Residuals and ℓ_{ll}

m	Algorithm	**15** Steps	**30** Steps
$0.2n$	DeMUN	30.33	30.74
	PGD	29.74	30.07
$0.4n$	DeMUN	34.43	34.86
	PGD	33.41	33.74

TABLE IV: Average PSNR (dB) with Residuals and $\ell_{i,1}$

Tables III & IV show the performance of DeMUN and PGD after incorporating residual connections. By comparing the results of Tables I & II with Tables III & IV, we see that including residual connection consistently improves reconstruction quality. For instance, at 0.4 sampling rate with intermediate loss, DeMUN's performance improves from 33.68dB to 34.43dB with 15 projections, and from 34.05dB to 34.86dB with 30 projections. Similar improvements are observed for PGD across all sampling rates. Furthermore, the combination of residual connections with intermediate loss and DeMUN yields the best performance, demonstrating these design choices are complementary rather than redundant. Based on these results, we recommend including residual connections in unrolled network architectures.

C. Number of Unrolled Steps

Another critical observation from our empirical results is that increasing the number of unrolled steps consistently improves reconstruction quality when using intermediate loss, without signs of overfitting. As shown in Tables I & II, at $0.4n$ sampling rate with intermediate loss, DeMUN's performance improves from 33.68dB with 15 projections to 34.05dB with 30 projections. This improvement pattern persists with residual connections, where performance increases from 34.43dB to 34.86dB. Figures 3 & 4 provide further insight: under the intermediate loss, each additional projection contributes meaningfully to reconstruction quality, while performance plateaus after a certain number of steps. This property simplifies the design process of unrolled networks and suggests that practitioners can safely increase the number of projections within their computational constraints, as overfitting does not appear to be a concern.

D. Projector Capacity

Finally, we examine the effect of varying the number of intermediate layers L in the DnCNN projector architecture while fixing our selected choices from above. Our experiments with $L \in \{3, 5, 10, 15\}$ layers reveal two key findings:

m	$L = 3$	$L = 5$	$L = 10$	$L = 15$
$0.2n$	30.06	30.33	30.34	30.19
$0.4n$	34.49	34.43	34.44	34.30

TABLE V: Average PSNR (dB) Under **15** Projection Steps

m	$L = 3$	$L = 5$	$L = 10$	$L = 15$
$0.2n$	30.32	30.74	30.71	30.60
$0.4n$	34.44	34.86	34.69	34.95

TABLE VI: Average PSNR (dB) Under **30** Projection Steps

- Increasing the number of layers from 5 to 15 results in negligible changes in the performance of DeMUNs, regardless of the number of projection steps.
- By comparing $L = 3$ and $L = 5$, we conclude that reducing $L \leq 3$ may impair performance. This suggests that while the performance is insensitive to the choice of L, some minimum network capacity, e.g. $L = 5$ is needed to learn effective image representations [27].

While these findings are specific to DnCNN architectures, we believe the general principle—that performance plateaus beyond a certain architectural capacity but degrades below a minimum threshold—likely extends to other projector designs. The careful study of this issue for other projector architectures is left for future research.

IV. Conclusion

In this paper, we conducted a comprehensive empirical study on the design choices of unrolled networks to solve linear inverse problems. We introduced the Deep Memory Unrolled Network (DeMUN) that eliminates the need to manually select an optimization algorithm by letting the data decide on the optimal gradient combination. Through extensive simulations, we demonstrated that training DeMUN with an unweighted intermediate loss function and incorporating residual connections represents the best existing practice, delivering superior performance compared to existing unrolled algorithms. Our simulations provide clear guidelines for selecting the number of convolutional layers in the projection step and the required projections in the unrolled algorithm.

References

[1] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006. ISSN 0018-9448. doi: 10.1109/TIT.2006.871582. URL http://ieeexplore.ieee.org/document/1614066/.

[2] Michael Lustig, David Donoho, and John M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, December 2007. ISSN 0740-3194, 1522-2594. doi: 10.1002/mrm.21391. URL https://onlinelibrary.wiley.com/doi/10.1002/mrm.21391.

[3] E.J. Candes and M.B. Wakin. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, March 2008. ISSN 1053-5888. doi: 10.1109/MSP.2007.914731. URL http://ieeexplore.ieee.org/document/4472240/.

[4] Richard G Baraniuk. Compressive sensing [lecture notes]. *IEEE signal processing magazine*, 24(4):118–121, 2007.

[5] David L. Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, November 2009. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0909892106. URL https://pnas.org/doi/full/10.1073/pnas.0909892106.

[6] Christopher A. Metzler, Arian Maleki, and Richard G. Baraniuk. From denoising to compressed sensing. *IEEE Transactions on Information Theory*, 62:5117–5144, 2016.

[7] Singanallur V. Venkatakrishnan, Charles A. Bouman, and Brendt Wohlberg. Plug-and-Play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948, Austin, TX, USA, December 2013. IEEE. ISBN 978-1-4799-0248-4. doi: 10.1109/GlobalSIP.2013.6737048. URL http://ieeexplore.ieee.org/document/6737048/.

[8] Shirin Jalali and Arian Maleki. From compression to compressed sensing. *Applied and Computational Harmonic Analysis*, 40(2):352–385, 2016.

[9] Sajjad Beygi, Shirin Jalali, Arian Maleki, and Urbashi Mitra. An efficient algorithm for compression-based compressed sensing. *Information and Inference: A Journal of the IMA*, 8(2):343–375, 2019.

[10] Yaniv Romano, Michael Elad, and Peyman Milanfar. The Little Engine that Could: Regularization by Denoising (RED), September 2017. URL http://arxiv.org/abs/1611.02862. arXiv:1611.02862 [cs].

[11] Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing, August 2020. URL http://arxiv.org/abs/1912.10557. arXiv:1912.10557 [eess].

[12] Yuelong Li, Or Bar-Shira, Vishal Monga, and Yonina C. Eldar. Deep Algorithm Unrolling for Biomedical Imaging, August 2021. URL http://arxiv.org/abs/2108.06637. arXiv:2108.06637 [cs].

[13] Jian Zhang, Bin Chen, Ruiqin Xiong, and Yongbing Zhang. Physics-Inspired Compressive Sensing: Beyond deep unrolling. *IEEE Signal Processing Magazine*, 40(1):58–72, January 2023. ISSN 1053-5888, 1558-0792. doi: 10.1109/MSP.2022.3208394. URL https://ieeexplore.ieee.org/document/10004834/.

[14] Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing, June 2018. URL http://arxiv.org/abs/1706.07929. arXiv:1706.07929 [cs].

[15] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep ADMM-Net for Compressive Sensing MRI. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/1679091c5a880faf6fb5e6087eb1b2dc-Paper.pdf.

[16] Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. doi: 10.1137/080716542. URL https://doi.org/10.1137/080716542. _eprint: https://doi.org/10.1137/080716542.

[17] Chunyan Zeng, Yan Yu, Zhifeng Wang, Shiyan Xia, Hao Cui, and Xiangkui Wan. GSISTA-Net: generalized structure ISTA networks for image compressed sensing based on optimized unrolling algorithm. *Multimedia Tools and Applications*, March 2024. ISSN 1573-7721. doi: 10.1007/s11042-024-18724-9. URL https://link.springer.com/10.1007/s11042-024-18724-9.

[18] Zhonghao Zhang, Yipeng Liu, Jiani Liu, Fei Wen, and Ce Zhu. AMP-Net: Denoising-Based Deep Unfolding for Compressive Image Sensing. *IEEE Transactions on Image Processing*, 30:1487–1500, 2021. ISSN 1057-7149, 1941-0042. doi: 10.1109/TIP.2020.3044472. URL https://ieeexplore.ieee.org/document/9298950/.

[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[20] Mariana-Iuliana Georgescu, Radu Tudor Ionescu, and Nicolae Verga. Convolutional Neural Networks With Intermediate Loss for 3D Super-Resolution of CT and MRI Scans. *IEEE Access*, 8:49112–49124, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2980266. URL https://ieeexplore.ieee.org/document/9034049/.

[21] Chong Mou, Qian Wang, and Jian Zhang. Deep Generalized Unfolding Networks for Image Restoration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17378–17389, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-66546-946-3. doi: 10.1109/CVPR52688.2022.01688. URL https://ieeexplore.ieee.org/document/9878586/.

[22] Eric Chen, Xi Chen, Arian Maleki, and Shirin Jalali. Comprehensive Examination of Unrolled Networks for Solving Linear Inverse Problems, January 2025. URL http://arxiv.org/abs/2501.04608. arXiv:2501.04608 [eess].

[23] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian Denoiser: Residual

Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017. ISSN 1057-7149, 1941-0042. doi: 10.1109/TIP.2017.2662206. URL https://ieeexplore.ieee.org/document/7839189/.

[24] Christopher A. Metzler, Ali Mousavi, and Richard G. Baraniuk. Learned D-AMP: Principled Neural Network based Compressive Image Recovery, November 2017. URL http://arxiv.org/abs/1704.06625. arXiv:1704.06625 [cs, stat].

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. URL http://arxiv.org/abs/1512.03385. arXiv:1512.03385 [cs].

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks, July 2016. URL http://arxiv.org/abs/1603.05027. arXiv:1603.05027 [cs].

[27] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4):611–629, August 2018. ISSN 1869-4101. doi: 10.1007/s13244-018-0639-9. URL https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9.