Exploring Unrolled Optimization

Samir Agarwala, and Jared Watrous

Abstract—A natural limitation to the capturing of real-world images is the presence of blur and noise, resulting from camera motion, lens intrinsics, hardware imperfections, low light conditions, or any number of sources. Many approaches exist for deconvolution, the process of attempting to mitigate these artifacts, including direct inversion in the frequency domain, Wiener deconvolution, half-quadratic splitting (HQS), and alternating direction method of multipliers (ADMM). The latter two of these approaches involve hyperparameters, where different values give different deconvolved images, and tuning these hyperparameters well can be an nontrivial and cumbersome. In this paper, we introduce three augmentations of the HQS algorithm that attempt to abstract away the hyperparameter tuning process. Specifically, we apply backpropagation to optimize three learned models: first, a direct global optimization of the HQS hyperparameters; second, a neural network hyperparameter predictor conditioned on the input image; and finally, a neural network hyperparameter predictor conditioned on the intermediate state of each iteration. We show that all three methods produce visually reasonable results and conclude that, with our training configuration, fine granularity prediction provides diminishing returns.

Index Terms—Computational Imaging, Denoising, Inverse Problems

1 Introduction

In computational imaging, blurry and noisy images are an inherent result of imperfect hardware and measurements. Naturally, the topic of deconvolution arises with the objective of recovering a blur- and noise-free image from a flawed observation. One potentially powerful approach is the Half-Quadratic Splitting (HQS) method [1], which, despite showing promising results, requires hyperparameter tuning and heuristics to optimally deconvolve an image. In this project, we explore three methods of learning these hyperparameters through automatic differentiation, including direct global optimization, image-conditioned hyperparameter prediction, and iteration-conditioned hyperparameter prediction.

2 Related Work

2.1 Solving inverse problems using optimization

Inverse problems such as denoising are extremely underconstrained problems and trying to denoise images without a prior will often lead to degenerate results as there are multiple plausible solutions. Thus, we often see that image priors, i.e. objective functions that dictate how we expect a natural image to appear, are used to guide the optimization process. A common prior used when solving the denoising problem is the total variation (TV) regularizer which penalises gradients in images, encouraging sparse gradients [2]. Using regularizers like TV along with a traditional euclidean distance objective, it is possible to get reasonable solutions to inverse problems by using optimization methods such as gradient descent [2] or adaptive optimization methods such as ADAM [3]. Another line of work involves breaking down the optimization problem into multiple steps that can be efficiently solved with regularization constraints. This line of work includes methods such as half quadratic splitting (HQS) [1] and alternating direction method of multipliers (ADMM) [4] which have shown promising results in solving inverse problems. However, these methods require hyperparameter tuning and we explore ways to automate this hyperparameter search using data-driven methods in our project.

2.2 Modeling optimization approaches using deep learning.

Past work has explored using deep learning to model *maximum a posteriori* (MAP) optimization problems. By using deep learning, methods are able to learn patterns from data and leverage inductive biases encoded in the network architecture to potentially make more intelligent decisions during the optimization process. For instance, [5] proposed a way to leverage information about image formation models into neural networks, and unroll the networks to solve inverse problems. This is done by weight sharing in a neural network and repeatedly applying the same network in the different optimization steps. The paper presents promising results on the deblurring and denoising tasks, showing how using inductive biases or learning priors from data can help us find better solutions to inverse problems.

2.3 Adaptations to HQS

There has been interesting work in ways to adapt HQS and related methods to leverage learned priors. For instance, there has been work [6] focused on proving fixed-point convergence guarantees of the alternating direction method of multipiers (ADMM) where the z-update steps leverage denoisers that are learned neural networks, i.e. there is guaranteed to be convergence to a solution if the initialization lies in a certain region under certain denoisers. Other work [7] has explored modelling the different iteration steps of ADMM using separately learned modules for each of the update steps. However, there is still scope to explore ways in which we can learn priors to predict hyperparameters for existing iterative solutions to inverse problems. For instance, hyperparameters such as ρ and λ, present in both HQS and

• *S. Agarwala and J. Watrous are MS CS students at Stanford University.*

ADMM, are often set by hand. By learning priors from data, we can potentially learn task-specific values, or those of even finer granularity, that lead to quicker convergence and better performance in solving inverse problems. Through our work, we hope to contribute towards the effective selection of hyperparameters for HQS.

3 Proposed Method

We propose three different methods of augmenting the HQS algorithm to learn hyperparameters using automatic differentiation, each method increasing in relative granularity. In all cases of backpropagation, we optimize using the AdamW algorithm, and apply a mean squared error (MSE) loss function on the pixels of the training image set.

3.1 Direct Global Backpropagation ("Vanilla")

Our first approach, which we refer to as "vanilla," is effectively a baseline method with the standard HQS pipeline. In this method, we define the hyperparameters ρ and λ to be learnable model parameters. We then backpropagate through the unrolled computation graph of the standard HQS algorithm, optimizing the parameters ρ and λ to minimize the error of the output image. We postulate that this method is effectively equivalent to an automated hyperparameter sweep, where the same learned hyperparameter values could be transferred to any standard HQS pipeline and see equivalent results.

3.2 Image-Conditioned Hyperparameter Prediction

Our second approach introduces a predictor model which takes an input image and outputs global predictions for the hyperparameters ρ and λ. Our predictor model is a fully connected feed-forward neural network which predicts the optimal values of ρ and λ directly from the pixels of the blurry input image. We hence refer to this approach as "image-conditioned" or "per-image," as the hyperparameters ρ and λ are dependent on the image being deconvolved, but the same values are used across all iterations of the HQS algorithm. To train the predictor neural network, we unroll the compute graph of the HQS algorithm across all iterations and backpropagate with the same pixel-space MSE loss as our baseline approach. The gradients are then used to update the predictor's weights such that the final output image of the algorithm is optimized to match the ground truth image.

3.3 Image-Conditioned Per-Iteration Hyperparameter Prediction

Our final approach extends our image-conditioned method by increasing the granularity of the hyperparameter predictions. In this method, we use the same hyperparameter predictor neural network as described in the previous section, which takes a blurry image as input and provides estimates of the optimal hyperparameters ρ and λ. In this method, however, rather than predicting the hyperparameters once at the start of the algorithm, we run the predictor model at the beginning of each iteration, before the HQS x-update, using the intermediate image x as input. As before, we unroll the entire compute graph of HQS, which this time includes one pass through the predictor model for each iteration, and backpropagate through the entire algorithm to optimize the weights of the predictor model using a pixel-space MSE loss on the output image.

For the first iteration of this approach, we provide as input to the predictor model the initial prediction x_0 of the output image. In our implementation, this initial prediction is an array of all zeros, indicating that the ρ and λ values used in the first iteration of this method is independent of the input image. That is, although the second and later iterations use hyperparameters predicted from the intermediate image x, the first iteration always uses the same globally predicted hyperparameters. We justify this design choice with the observation that if the first iteration conditioned on the input image, the per-iteration approach would be a complete superset of the previously described image-conditioned method, limiting the diversity of our explored HQS augmentations.

4 Experimental Results

We perform an ablation study examining how the choice of regularizer, noise level, blur level and number of steps during network unrolling affect denoising performance.

4.1 Ablation Study

4.1.1 Different Regularizers

We present quantitative results comparing different regularizers including a pretrained denoising CNN, anistopric TV (ani-tv) and isotropic TV (iso-tv) in Table 4.1.1 and qualitative results in Figure 4. From the results, it is clear that the denoising CNN significantly outperforms TV as a regularizer, showing how important learned priors are in inverse problems such as deconvolution.

	Vanilla PSNR	Per-Image PSNR	Per-Iter PSNR
DnCNN	19.0386	19.1763	**19.2469**
Ani-TV	14.6589	14.6894	**14.6993**
Iso-TV	14.6589	14.6894	**14.6993**

TABLE 1
PSNR values for images generated by each of the three methods using different regularizers: a denoising Convolutional Neural Network (DnCNN), anisotropic Total Variation (Ani-TV), and isotropic Total Variation (TV), with $\sigma_{noise} = 0.1$, $\sigma_{blur} = 2.5$, and 5 HQS iterations.

4.1.2 Different Noise and Blur Levels

We present quantitative results comparing our proposed methods across different noise and blur levels in Tables 4.1.2 and 4.1.2, and present qualitative results in Figures 5 and 6. We see that performance significantly drops as the standard deviation of the noise or blur increases which correlates with the task getting more challenging as seen qualitatively in the input images.

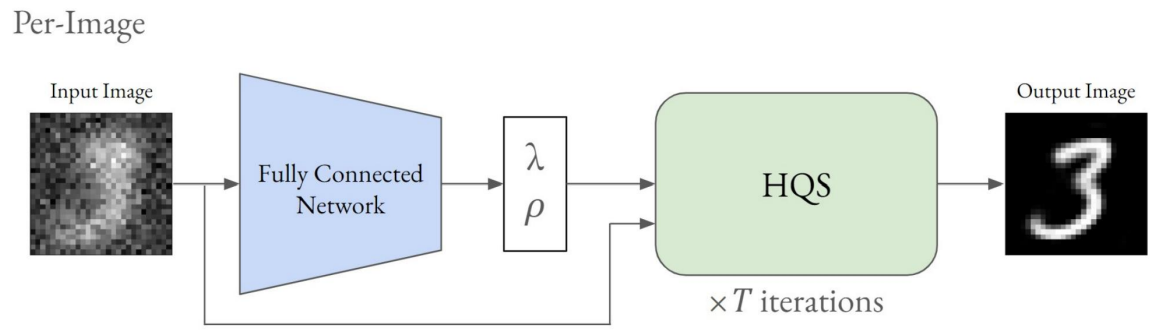

Fig. 1. Our image-conditioned HQS augmentation method, using a fully connected neural network to predict global hyperparameters ρ and λ from the initial input image.

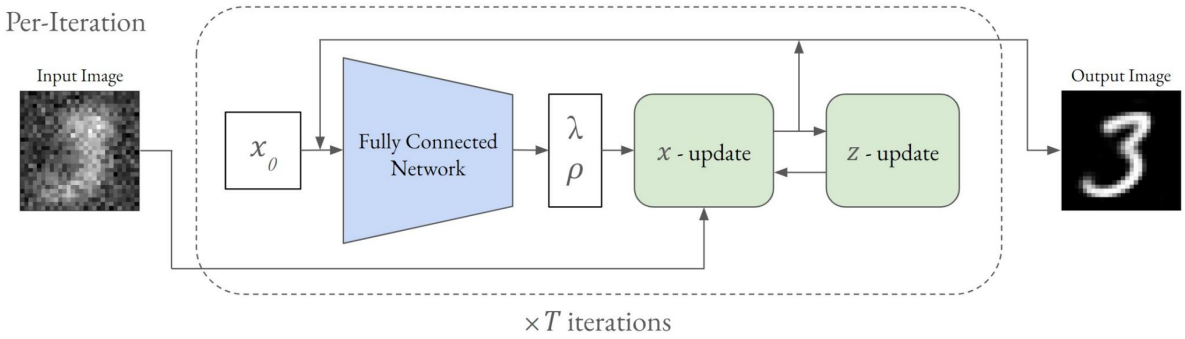

Fig. 2. Our image-conditioned per-iteration HQS augmentation method, using a fully connected neural network to predict hyperparameters ρ and λ at each iteration. In the first iteration, we use the initial image prediction x_0, which is an array of all zeros.

	Vanilla PSNR	Per-Image PSNR	Per-Iter PSNR
$\sigma_n = 0.01$	24.8461	25.7526	**25.8529**
$\sigma_n = 0.1$	19.0386	19.1763	**19.2469**
$\sigma_n = 1$	**12.6915**	12.6700	12.6843

TABLE 2
PSNR values for images generated by each of the three methods using different noise levels with standard deviations σ_{noise} of 0.01, 0.1, and 1, with a DnCNN regularizer, $\sigma_{blur} = 2.5$, and 5 HQS iterations.

	Vanilla PSNR	Per-Image PSNR	Per-Iter PSNR
$\sigma_b = 1$	25.4636	**25.5458**	25.5074
$\sigma_b = 2.5$	19.0386	19.1763	**19.2469**
$\sigma_b = 5$	14.4015	14.3945	**14.4790**

TABLE 3
PSNR values for images generated by each of the three methods using different Gaussian blur kernels with standard deviations σ_{blur} of 1, 2.5, and 5, with a DnCNN regularizer, $\sigma_{noise} = 0.1$, and 5 HQS iterations.

4.1.3 Different Number of Unrolling Steps

In our last ablation study, we examine the effect of unrolling our network for a different number of steps and present quantitative results in Table 4.1.3 and qualitative results in Figure 7. From the results, we see a general trend that unrolling the network for more steps performs better across the methods showing the importance of unrollling for as many steps as feasible.

	Vanilla PSNR	Per-Image PSNR	Per-Iter PSNR
$n = 1$	14.4812	**14.6237**	14.6018
$n = 5$	19.0386	19.1763	**19.2469**
$n = 10$	19.1541	19.3271	**19.3406**

TABLE 4
PSNR values for images generated by each of the three methods using different numbers of HQS steps with $n = 1$, $n = 5$, and $n = 10$, with a DnCNN regularizer, $\sigma_{noise} = 0.1$, and $\sigma_{blur} = 2.5$.

4.2 Comparison with Baselines

After performing the ablation study, we now have the best parameters for each method i.e. we get best results across the board when unrolling the network for 10 steps and use a deep neural network as the regularization prior. Thus, in this section we use this configuration of models to compare our proposed methods to traditional deconvolution methods such as inverse and Wiener deconvolution. We present a quantitative comparison in Table 4.2 and qualitative

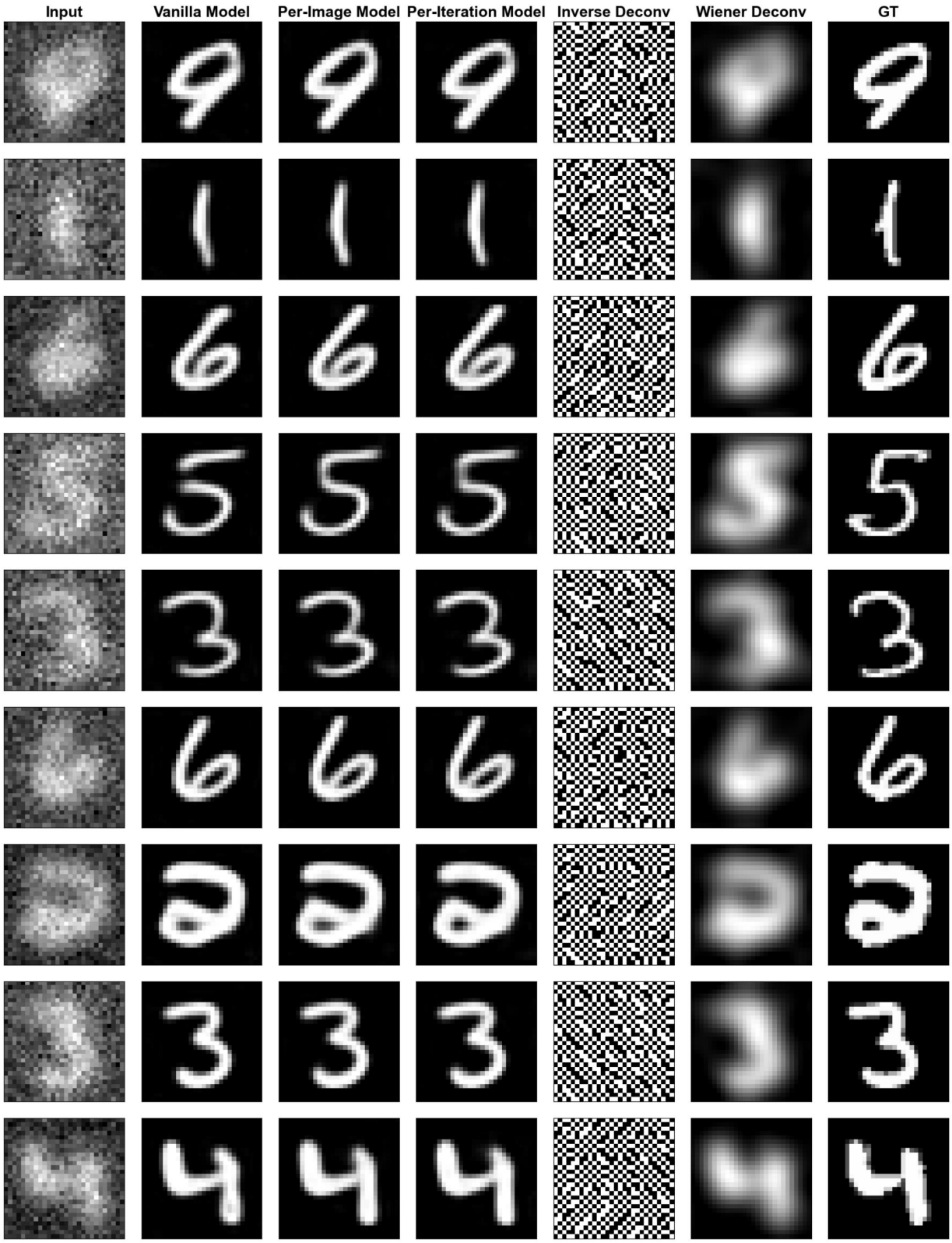

Fig. 3. Qualitative comparison of our examined methods with baselines.

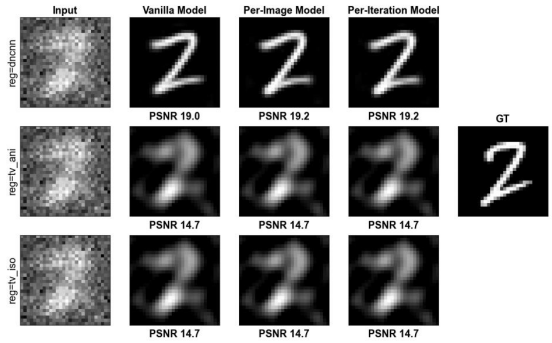

Fig. 4. Different regularizers

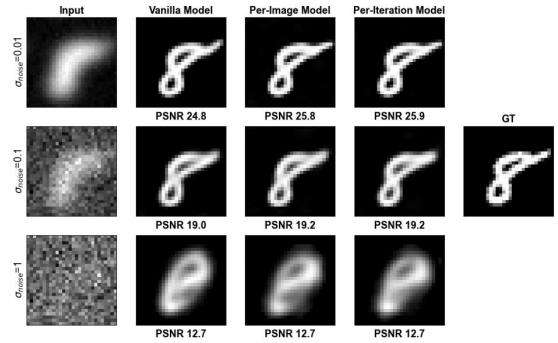

Fig. 5. Different noise levels

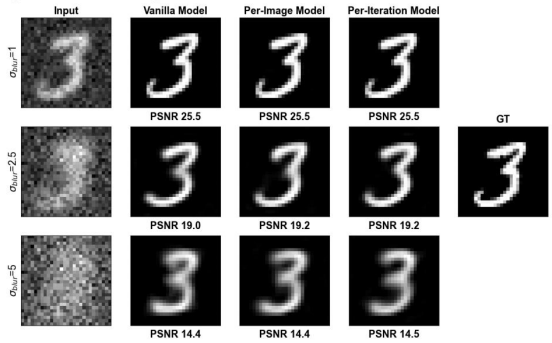

Fig. 6. Different blur levels

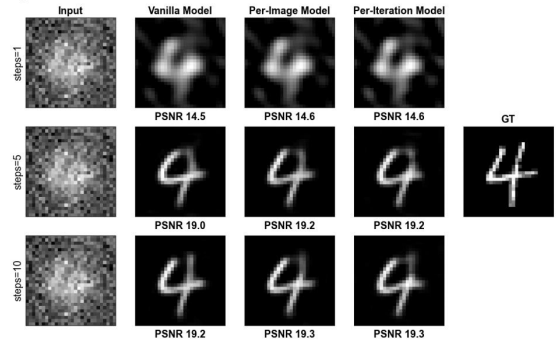

Fig. 7. Different number of steps

Fig. 8. Qualitative ablation study involving different regularisers, noise levels, blur levels and number of steps we unroll the network during training.

	Vanilla	Per-Image	Per-Iter	Inverse Deconv.	Wiener Deconv.
Test PSNR	19.1541	19.3271	19.3406	3.1735	11.7828

TABLE 5
Summary of PSNR values on the MNIST test set for images deconvolved by the best model for each examined method along with a Weiner and Inverse deconvolution baseline. The noisy images from the MNIST test set have $\sigma_{noise} = 0.1$ and $\sigma_{blur} = 2.5$ in this experiment.

results in Figure 3. From the results, it is clear that our proposed approaches perform significantly better than both the Wiener and the inverse deconvolution. Additionally, we see the inverse deconvolution gives degenerate results while Wiener results does a reasonable job but is unable to reach similar levels of performance likely because it does incorporate priors about natural images like we do in our methods. Our work thus shows the potential of using data to optimise hyperparameters for algorithms such as HQS for solving inverse problems.

We also observe from the results presented in Table 4.2 and the ablation studies performed, that in general the per-iteration methods performs better than the per-image method and these are followed by the vanilla method. Although the differences are small we do observe this trend, and this is likely because the per-iteration method adapts parameters at each step and thus should be able to lead to better results compared to having a more rigid per-image or global hyperparameters. The small difference in the parameters do indicate that maybe there is no need to find per-image or per-iteration parameters for these models and simply using task-specific parameters may work well in practice. Lastly, the difference in performance could also be an artifact of the MNIST dataset which is relatively simple to model and thus there is a need to evaluate this model on more complex datasets and see if there are more significant differences in the proposed methods.

5 Conclusion

Our results show a clear difference in quality between the inverse deconvolution and Wiener deconvolution baselines and our three HQS augmentations. The quality difference across our three augmentations is much more subtle. Quantitatively, we see a consistent trend in which finer granularity hyperparameter prediction delivers slightly higher PSNRs. Perceptually, however, we find that most examples require close inspection to observe the differences between the methods.

We draw from the observation that the MNIST dataset, which our experiments were trained and tested on, is perceptually fairly simple. We speculate that, while our results hold for this data, training on larger and more complex images could provide more variance in the perceptual quality of the output images, potentially offering more insight

into the benefits and drawbacks of each of our three HQS augmentations.

We also acknowledge that the HQS algorithm is often run for much longer, typically ranging from 20 to 50 iterations for image deconvolution, as compared to our 1 to 10 iterations. Although we already observe our methods quantitatively plateauing with the number of iterations, we surmise that, combined with a more complex dataset, increasing the number of iterations could bring the quality comparisons closer to those of deconvolution algorithms used in practice, offering a more practical analysis of our approaches.

With the above in mind, we consider that simple global hyperparameter optimization may be sufficient in practice, with the footnote that this conclusion primarily applies to our specific training configuration with MNIST and few iterations. As we consider it beyond our compute limitations at this time, we leave the exploration of scaling these approaches, with larger datasets and more iterations, to future work.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Gordon Wetzstein for helpful discussions and feedback. We also acknowledge that we used parts of the starter code for the homework assignments in the class and our solutions to the assignments in the implementation of this project.

REFERENCES

[1] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE transactions on Image Processing*, vol. 4, no. 7, pp. 932–946, 1995.

[2] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

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

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

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

[6] 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.

[7] C. Zhou and M. R. Rodrigues, "Admm-based hyperspectral unmixing networks for abundance and endmember estimation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2021.