A SCALABLE TRAINING STRATEGY FOR BLIND MULTI-DISTRIBUTION NOISE REMOVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite recent advances, developing general-purpose universal denoising and artifact-removal networks remains largely an open problem: Given fixed network weights, one inherently trades-off specialization at one task (e.g., removing Poisson noise) for performance at another (e.g., removing speckle noise). In addition, training such a network is challenging due to the curse of dimensionality: As one increases the dimensions of the specification-space (i.e., the number of parameters needed to describe the noise distribution) the number of unique specifications one needs to train for grows exponentially. Uniformly sampling this space will result in a network that does well at very challenging problem specifications but poorly at easy problem specifications, where even large errors will have a small effect on the overall mean squared error.

In this work we propose training denoising networks using an adaptive-sampling/active-learning strategy. Our work improves upon a recently proposed universal denoiser training strategy by extending these results to higher dimensions and by incorporating a polynomial approximation of the true specification-loss landscape. This approximation allows us to reduce training times by an order of magnitude. We test our method on joint Poisson-Gaussian-speckle noise and demonstrate that, with our proposed training strategy, a single blind, generalist denoiser network can achieve mean squared errors within a uniform bound of specialized denoiser networks across a large range of operating conditions.

1 INTRODUCTION

Neural networks have become the gold standard for solving a host of imaging inverse problems Ongie et al. (2020). From denoising and deblurring to compressive sensing and phase retrieval, modern deep neural networks significantly outperform classical techniques like BM3D Dabov et al. (2007) and KSVD Aharon et al. (2006).

The most straightforward and common approach to apply deep learning to inverse problems is to train a neural network to learn a mapping from the space of corrupted images/measurements to the space of clean images. In this framework, one first captures or creates a training set consisting of clean images $x_1, x_2, \dots$ and corrupted images $y_1, y_2, \dots$ according to some known forward model $p(y_i|x_i, \theta)$, where $\theta \in \Theta$ denotes the latent variable(s) specifying the forward model. For example, when training a network to remove additive white Gaussian noise

$$p(y_i|x_i, \theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{\|y_i - x_i\|^2}{2\sigma^2}, \tag{1}$$

and the latent variable θ is the standard deviation σ. With a training set of L pairs $\{x_i, y_i\}_{i=1}^{L}$ in hand, one can then train a network to learn a mapping from y to x.

Typically, we are not interested in recovering signals from a single corruption distribution (e.g., a single fixed noise standard deviation σ) but rather a range of distributions. For example, we might want to remove additive white Gaussian noise with standard deviations anywhere in the range $[0, 50]$ ($\Theta = \{\sigma|\sigma \in [0, 50]\}$). The size of this range determines how much the network needs to generalize and there is inherently a trade-off between specialization and generalization. By and large, a network

trained to reconstruct images over a large range of corruptions (a larger set Θ) will under-perform a network trained and specialized over a narrow range Zhang et al. (2017).

This problem becomes significantly more challenging when dealing with mixed, multi-distribution noise. As one increases the number of parameters (e.g., Gaussian standard deviation, Poisson rate, number of speckle realizations, ...) the space of corrupted signals one needs to reconstruct grows exponentially: The specification space becomes the Cartesian product (e.g., $\Theta = \Theta_{Gaussian} \times \Theta_{Poisson} \times \Theta_{speckle}$) of the spaces of each of the individual noise distributions.

This expansion does not directly prevent someone (with enough compute resources) from training a "universal" denoising algorithm. One can sample from Θ, generate a training batch, optimize the network to minimize some reconstruction loss, and repeat. However, this process depends heavily on the policy/probability-density-function π used to sample from Θ. As noted in Gnanasambandam & Chan (2020) and corroborated in Section 5, uniformly sampling from Θ will produce networks that do well on hard examples but poorly (relative to how well a specialized network performs) on easy examples.

1.1 Our Contribution

In this work we develop an adaptive-sampling/active-learning strategy that allows us to train a single "universal" network to remove mixed Poisson-Gaussian-speckle noise such that the network consistently performs within a uniform bound of specialized bias-free DnCNN baselines Zhang et al. (2017); Mohan et al. (2019). Our key contribution is a novel, polynomial approximation of the specification-loss landscape. This approximation allows us to tractably apply (using two orders of magnitude fewer training examples than it would otherwise require) the adaptive-sampling strategy developed in Gnanasambandam & Chan (2020), wherein training a denoiser is framed as a constrained optimization problem.

2 Related Work

Overcoming the specialization-generalization trade-off has been the focus of intense research efforts over the last 5 years.

2.1 Adaptive Denoising

One approach to improve generalization is to provide the network information about the current problem specifications θ at test time. For example, Gharbi et al. (2016) demonstrated one could provide a constant standard-deviation map as an extra channel to a denoising network so that it could adapt to i.i.d. Gaussian noise. Zhang et al. (2018) extends this idea by adding a general standard-deviation map as an extra channel, to deal with spatially-varying Gaussian noise. This idea was recently extended to deal with correlated Gaussian noise Metzler & Wetzstein (2021). The same framework can be extended to more complex tasks like compressive sensing, deblurring, and descattering as well Wang et al. (2022); Tahir et al. (2022). These techniques are all non-blind and require an accurate estimate of the specification parameters θ to be effective.

2.2 Universal Denoising

Somewhat surprisingly, the aforementioned machinery may be unnecessary if the goal is to simply remove additive white Gaussian noise over a range of different standard deviations. Mohan et al. (2019) recently demonstrated one can achieve significant invariance to the noise level by simply removing biases from the network architecture. Wang & Morel (2014) also achieves similar invariance to noise level by scaling the input images to the denoiser to match the distribution it was trained on. Alternatively, at a potentially large computational cost, one can apply iterative "plug and play" or diffusion models that allow one to denoise a signal contaminated with noise with parameters θ' using a denoiser/diffusion model trained for minimum mean squared error additive white Gaussian noise removal Venkatakrishnan et al. (2013); Romano et al. (2017); Kawar et al. (2021). These plug and play methods are non-blind and require knowledge of the likelihood $p(y|x, \theta')$ at test time.

2.3 TRAINING STRATEGIES

Generalization can also be improved by modifying the training set Elman (1993). In the context of image restoration problems like denoising, Gao & Grauman (2017) propose updating the training data sampling distribution each epoch so as to sample the data that the neural network performed worse on during the prior epoch preferentially, in an ad-hoc way. In Gnanasambandam & Chan (2020) the authors developed a principled adaptive training strategy by framing training a denoiser across many problem specifications as a minimax optimization problem. This strategy will be described in detail in Section 4.

2.4 RELATIONSHIP TO EXISTING WORKS

We go beyond Gnanasambandam & Chan (2020) by incorporating a polynomial approximation of the specification-loss landscape. This approximation is the key to scaling the adaptive training methodology to high-dimensional latent parameter spaces. It allows us to efficiently train a blind image denoiser that can operate effectively across a large range of noise conditions.

3 PROBLEM FORMULATION

3.1 NOISE MODEL

This paper focuses on removing joint Poisson-Gaussian-speckle noise using a single blind image denoising network. Such noise occurs whenever imaging scenes illuminated by a coherent (e.g., laser) source. In this context, photon/shot noise introduces Poisson noise, read noise introduces Gaussian noise, and the constructive and destructive interference caused by the coherent fields scattered off optically rough surfaces causes speckle noise Goodman (2007).

The overall forward model can be described by

$$y_i = \frac{1}{\alpha}\text{Poisson}(\alpha(r \circ w_i)) + n_i, \tag{2}$$

where the additive noise n_i follows a Gaussian distribution $\mathcal{N}(0, \sigma^2\mathbf{I})$; the multiplicative noise w_i follows a Gamma distribution with concentration parameter B/β and rate parameter B/β, where B is the upper bound on β; and α is a scaling parameter than controls the amount of Poisson noise. The forward model is thus specified by the set of latent variables $\theta = \{\sigma, \alpha, \beta\}$.

A few example images generated according to this forward model are illustrated in Figure 1. Variations in the problem specifications results in drastically different forms of noise.

3.2 SPECIFICATION-LOSS LANDSCAPE

A *specification* is a set of n parameters that define a task. In our setting, the specifications are the distribution parameters describing the noise in an image. Each of these parameters is bounded in an interval $[l_i, r_i]$, for $1 \leq i \leq n$. The *specification space* Θ is the Cartesian product of these intervals: $\Theta = [l_1, r_1] \times \cdots \times [l_n, r_n]$. Suppose we have a function f that can solve a task (e.g., denoising) at any specification in Θ, albeit with some error. Then the *specification-loss landscape*, for a given f over Θ, is the function $\mathcal{L}_f$ which maps points θ from Θ to the corresponding error that f achieves at that specification.

Now suppose that all functions f under consideration come from some family of functions $\mathcal{F}$. Let the ideal function from $\mathcal{F}$ that solves a task at a particular specification θ be $f^\theta_{\text{ideal}} = \arg\min_{f\in\mathcal{F}} \mathcal{L}_f(\theta)$. With this in mind, we define the *ideal specification-loss landscape* as the function that maps points θ in Θ to the loss that f^θ_{ideal} achieves on the task with specification θ, and denote it $\mathcal{L}_{\text{ideal}}$.

3.3 THE UNIFORM GAP PROBLEM

Our goal is to find a single function $f^* \in \mathcal{F}$ that achieves consistent performance across the specification space Θ, compared to the ideal function at each point $\theta \in \Theta$, f^θ_{ideal}. More precisely, we

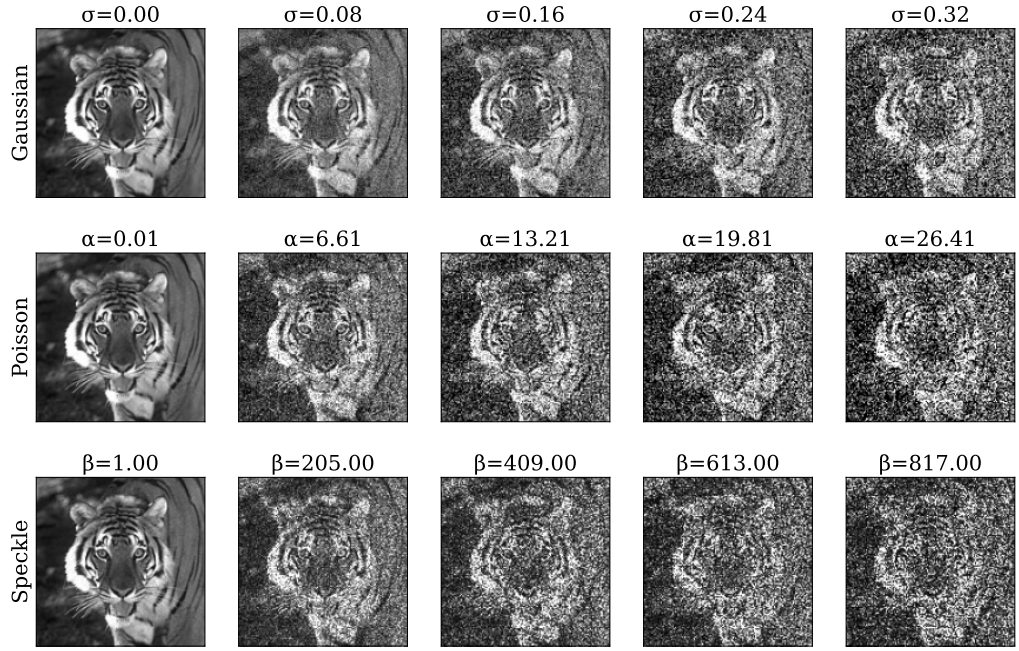

Figure 1: **Varying the noise specifications.** The first row shows images corrupted by Gaussian Noise, the second row shows images corrupted by Poisson noise, and the last row shows images corrupted by speckle noise. In each of the rows, the other noise parameters are held fixed at 0, 0.01, and 1.00, respectively.

want to minimize the maximum gap in performance between f^* and $f^{\theta}_{\text{ideal}}$ across all of Θ. Following Gnanasambandam & Chan (2020), we can frame this objective as the following optimization problem

$$f^* = \arg\min_{f \in \mathcal{F}} \sup_{\theta \in \Theta} \left\{ \mathcal{L}_f(\theta) - \mathcal{L}_{\text{ideal}}(\theta) \right\}, \tag{3}$$

which we call the *uniform gap problem*.

4 Proposed Method

4.1 Adaptive Training

To solve the optimization problem given in equation 3, Gnanasambandam & Chan (2020) propose rewriting it in its Lagrangian dual formulation and then using dual ascent, which yields the following iterations:

$$f^{t+1} = \arg\min_{f \in \mathcal{F}} \left\{ \int_{\theta \in \Theta} \mathcal{L}_f(\theta) \lambda^t(\theta) d\theta \right\} \tag{4}$$

$$\lambda^{t+\frac{1}{2}} = \lambda^t + \gamma^t \left(\frac{\mathcal{L}_{f^{t+1}}}{\mathcal{L}_{\text{ideal}}} - 1 \right) \tag{5}$$

$$\lambda^{t+1} = \lambda^{t+\frac{1}{2}} / \int_{\theta \in \Theta} \lambda^{t+\frac{1}{2}}(\theta) d\theta, \tag{6}$$

where $\lambda(\theta)$ represents a dual variable at specification $\theta \in \Theta$, and γ is the dual ascent step size.

We can interpret equation 4 as fitting a model f to the training data, where $\lambda(\theta)$ is the probability of sampling a task at specification θ to draw training data from. Next, equation 5 updates the sampling distribution $\lambda(\theta)$ based on the difference between the current model f^{t+1}'s performance across $\theta \in \Theta$ and the ideal models' performances. Lastly equation 6 ensures that $\lambda(\theta)$ is a properly normalized probability distribution. We provide the derivation of the dual ascent iterations from Gnanasambandam & Chan (2020) in Appendix C.

While Θ has been discussed thus far as a continuum, in practice we sample Θ at discrete locations and compare the model being fit to the ideal model performance at these discrete locations only, so that $|\Theta|$ is finite. Computing $\mathcal{L}_{\text{ideal}}(\theta)$ for each $\theta \in \Theta$ is extremely computationally time-intensive if $|\Theta|$ is large; if f is a neural network, it becomes necessary to train $|\Theta|$ neural networks. Furthermore, while $\mathcal{L}_{\text{ideal}}(\theta)$ can be computed offline independent of the dual ascent iterations, during the dual ascent iterations, each update of λ requires the evaluation of $\mathcal{L}_{f^{t+1}}$ for each $\theta \in \Theta$, which is also time intensive if $|\Theta|$ is large.

The key insight underlying our work is that one can approximate $\mathcal{L}_{\text{ideal}}$ and $\mathcal{L}_f$ in order to drastically accelerate the training process.

4.2 Specification-loss landscape approximations

Let $\mathcal{P}$ be a class of functions which we will use to approximate the specification-loss landscape. Instead of computing $\mathcal{L}_{\text{ideal}}(\theta)$ for each $\theta \in \Theta$, we propose instead computing $\mathcal{L}_{\text{ideal}}(\theta)$ at a set of locations $\theta \in \Theta_{\text{sparse}}$, where $|\Theta_{\text{sparse}}| \ll |\Theta|$, and then using these values to form an approximation P_{ideal} of $\mathcal{L}_{\text{ideal}}(\theta)$, as

$$P_{\text{ideal}} = \underset{P \in \mathcal{P}}{\arg\min} \sum_{\theta \in \Theta_{\text{sparse}}} ||P(\theta) - \mathcal{L}_{f_{\text{ideal}}}(\theta)||_2^2. \tag{7}$$

We can similarly approximate $\mathcal{L}_{f^{t+1}}$ with a polynomial $P_{f^{t+1}}$. Then we can solve equation 3 using dual ascent as before, replacing $\mathcal{L}_{\text{ideal}}$ and $\mathcal{L}_{f^{t+1}}$ with P_{ideal} and $P_{f^{t+1}}$ where appropriate, resulting in a modification to equation 5:

$$\lambda^{t+\frac{1}{2}} = \lambda^t + \gamma^t \left(\frac{P_{f^{t+1}}}{P_{\text{ideal}}} - 1 \right). \tag{8}$$

To justify our use of this approximation, we first consider a linear subspace projection "denoiser" and show that its specification-loss landscape is linear with respect to its specifications, and is thus easy to approximate.

Example 1. *Let $y = \alpha \text{Poisson}(\frac{1}{\alpha} x_o) + n$ with $n \sim N(0, \sigma^2 \mathbf{I})$, let C denote a k-dimensional subspace of $\mathbb{R}^n$ ($k < n$), and let the denoiser be the projection of y onto subspace C denoted by $P_C(y) = \mathbf{P}y$. Then, assuming $\frac{1}{\alpha} x_o$ is large, for every $x_o \in C$*

$$\mathbb{E}\|P_C(y) - x_o\|_2^2 \approx k\sigma^2 + \alpha tr(\mathbf{P}\text{diag}(x)\mathbf{P}^t),$$

where $tr(\cdot)$ denotes the trace. The loss landscape is linear with respect to σ^2 and α.

Proof. First note that if $\frac{1}{\alpha} x_o$ is large the distribution of $\alpha\text{Poisson}(x/\alpha)$ can be approximated with $N(x, \alpha\text{diag}(x))$. Accordingly, $y \approx x + \nu$ where $\nu \sim N(0, \sigma^2 \mathbf{I} + \alpha\text{diag}(x))$. Since the projection onto a subspace is a linear operator and since $P_C(x_o) = x_o$ we have

$$\mathbb{E}\|P_C(y) - x_o\|_2^2 \approx \mathbb{E}\|x_o + P_C(\nu) - x_o\|_2^2 = \mathbb{E}\|P_C(\nu)\|_2^2.$$

Let $r = \mathbf{P}\nu$. Note that $r \sim N(0, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = \sigma^2 \mathbf{P}\mathbf{P}^t + \alpha \mathbf{P}\text{diag}(x)\mathbf{P}^t$. Accordingly,

$$\begin{aligned} \mathbb{E}\|P_C(\nu)\|_2^2 = \mathbb{E}\|r\|^2 = tr(\mathbf{\Sigma}) &= \sigma^2 tr(\mathbf{P}\mathbf{P}^t) + \alpha tr(\mathbf{P}\text{diag}(x)\mathbf{P}^t), \\ &= k\sigma^2 + \alpha tr(\mathbf{P}\text{diag}(x)\mathbf{P}^t), \end{aligned}$$

where the last equality follows from the fact that $\mathbf{P}\mathbf{P}^t = \mathbf{P}$ and the trace of a k-dimensional projection matrix is k.

□

Additionally, beyond this theoretical example, we plot the specification-loss landscapes for two of the denoising problems we consider. We show the achievable PSNR (peak signal-to-noise ratio) versus noise parameters plots for denoising Poisson-Gaussian and Speckle-Gaussian noise in Figure 2. The same figure for Speckle-Poisson noise can be found in Appendix B. They are clearly smooth and well-behaved, and we justify their approximation by polynomials using cross-validation, details of

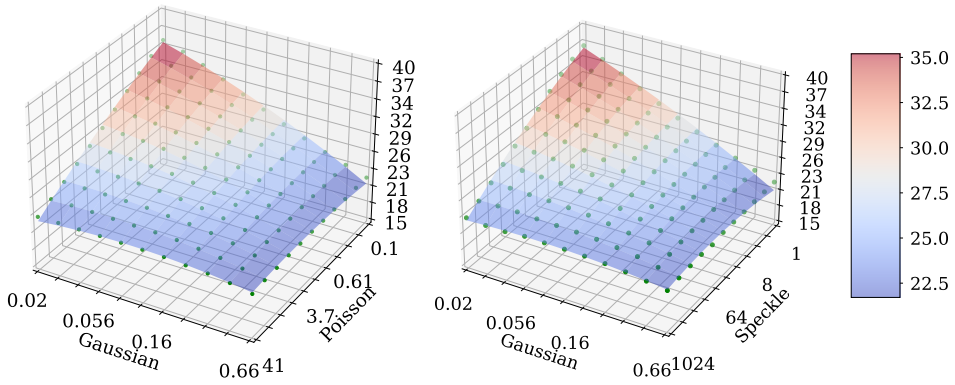

Figure 2: **Loss Landscape Visualizations.** PSNR, which we use as our proxy for error, versus denoising task specifications. The specification-loss landscapes are smooth and amenable to approximation.

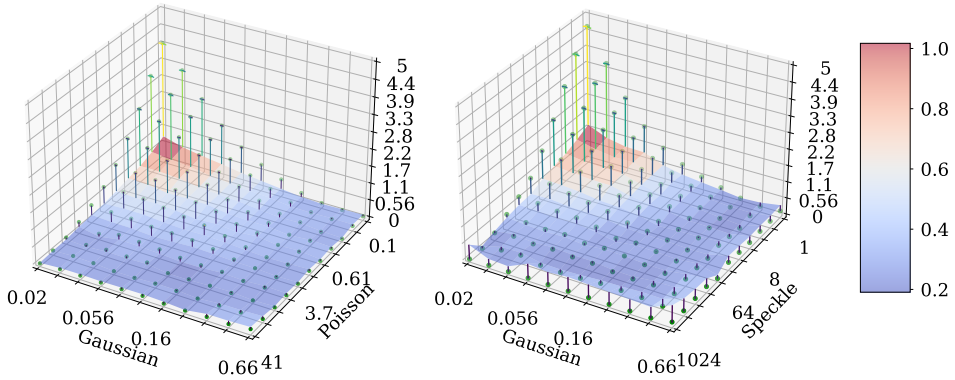

Figure 3: **Performance comparison between a network trained with adaptive training, using dense sampling of the specification-loss landscape, and a network trained with uniform sampling of the noise levels of the training data.** The surface in the above plots represents the difference in performance between a network trained with the adaptive training strategy with dense sampling of the specification-loss landscape and the ideal networks, and the points represent the differences in performance between a network trained with uniform sampling of the noise levels of the training data and the ideal. Adaptively sampling the noise levels of the training data using a dense sampling of the specification-loss landscape results in a network which uniformly under-performs specialized networks (surface is flat) whereas uniform sampling results in networks that do terribly under certain conditions (points are very high in some regions).

which can be found in Appendix A. In practice, we approximate the ideal PSNRs as $Q(s)$ rather than the ideal mean squared errors because we desire a more uniform PSNR gap rather than a more uniform MSE gap, in the context of denoising. Then, following Gnanasambandam & Chan (2020), we convert the ideal PSNRs to mean squared errors with the mapping $P(s) = 10^{-Q(s)/10}$ for use in the dual ascent iterations.

5 Experimental Results

5.1 Implementation Details

We use the 20-layer DnCNN architecture Zhang et al. (2017) for our denoiser. We remove all biases from the network layers, following Mohan et al. (2019). We train all of our networks for 50 epochs, with 3000 mini-batches per epoch and 128 image patches per batch, for a total of 384,000 image

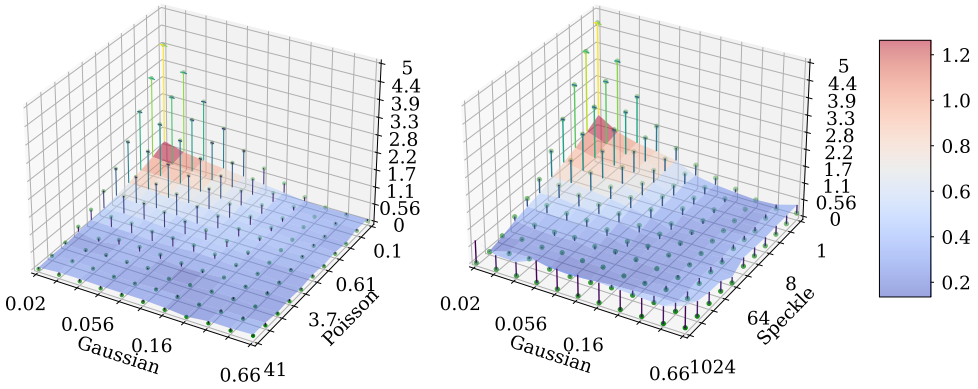

Figure 4: **Performance comparison between a network trained with adaptive training, using sparse sampling of the specification-loss landscape, and a network trained with uniform sampling of the noise levels of the training data.** The surface in the above plots represents the difference in performance between the a network trained with the adaptive strategy with sparse sampling and polynomial interpolation of the specification-loss landscape and the ideal networks, and the points represent the differences in performance between a network trained with uniform sampling of the noise levels of the training data and the ideal networks. Like the networks trained with adaptive noise level sampling and dense sampling, we still achieve performance that is uniformly worse than the ideal without severe failure modes.

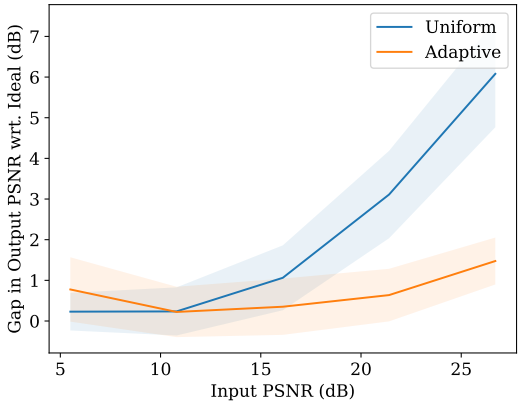

Figure 5: **Adaptive vs Uniform Training, 3D specification space.** Adaptive sampling with the polynomial approximation works effectively in the 3D problem space and produces a network whose performance is consistently close to the ideal. By contrast, a network trained by uniformly sampling from the space performs far worse than the specialized networks in certain contexts. The error bars represent one standard-deviation.

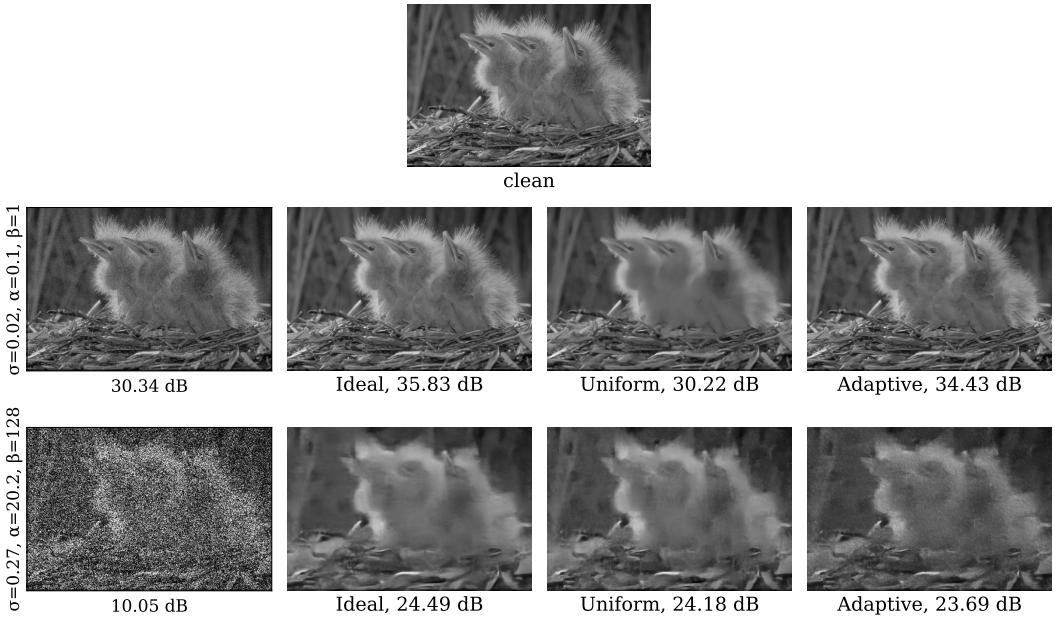

Figure 6: **Qualitative comparisons.** Comparison between the performance of the ideal, uniform-trained, and adaptive-trained denoisers on a sample image corrupted with a low amount of noise and corrupted with a high amount of noise. Our adaptive blind training strategy performs only marginally worse than an ideal, non-blind baseline when applied to "easy" problem specifications, and significantly better than the uniform baseline, while also being only marginally worse than an ideal baseline and uniform baseline under "hard" problem specifications.

patches total. We use the Adam optimizer Kingma & Ba (2015) to optimize the weights with a learning rate of 1×10^{-4}, with an L2 loss.

In practice, rather than training a model to convergence, to save training time, we approximately equation 4 of the dual ascent iterations by training the model for 10 epochs. While the resulting adaptively trained denoisers do not achieve a totally uniform gap with respect to the ideal baselines, their gap is much closer to uniform than that of the uniform specification sampling trained denoiser baseline, as can be seen in Figures 3, 4, and 5.

To construct the loss-landscapes for each mixed noise type, we sample the loss at 10 random specifications as well as the loss at the specification support's endpoints, for a total of 14 samples for Poisson-Gaussian, Speckle-Poisson, and Speckle-Gaussian noise and 18 samples for Speckle-Poisson-Gaussian noise. In Appendix E, we compute exactly how many points from the specification-loss landscape are required to be known in order to fit our approximation, and show that the computation cost scales quadratically in the number of dimensions. Note that the specification space for the first three noise types contains 100 specifications, and the last noise type 1000, which implies saving of 1 and 2 orders of magnitude of training time, respectively.

5.2 Data

To train our denoising models, we curate a high-quality image dataset that combines multiple high resolution image datasets: the Berkeley Segmentation Dataset Martin et al. (2001), the Waterloo Exploration Database Ma et al. (2017), the DIV2K dataset Agustsson & Timofte (2017), and the Flick2K dataset Lim et al. (2017). To test our denoising models, we use the validation dataset from the DIV2K dataset. We use a patch size of 40 pixels by 40 pixels, and patches are randomly cropped from the training images with flipping and rotation augmentations, to generate a total of 384000 patches. All images are grayscale and scaled to the range $[0, 1]$. We use the BSD68 dataset Roth & Black (2005) as our testing dataset.

5.3 SETUP

Noise Parameters. We consider four types of mixed noise distributions: Poisson-Gaussian, Speckle-Poisson, Speckle-Gaussian, and Speckle-Poisson-Gaussian noise. In each mixed noise type, $\sigma \in [0.02, 0.66]$, $\alpha \in [0.1, 41]$, and $\beta \in [1, 1024]$, and we discretize each range into 10 bins. Note that $B = 1024$ for speckle noise, following the parameterization in Section 3.1. These ranges were chosen as they correspond to input PSNRs of roughly 5 to 30 dB.

Training Setup. For each of Poisson-Gaussian, Speckle-Poisson, and Speckle-Gaussian noises separately, we use the approximations P_{ideal}, P_f to adaptively train a denoiser f^*_{sparse}, use $\mathcal{L}_{\text{ideal}}$, $\mathcal{L}_f$ to adaptively train a denoiser f^*_{dense}, and use uniform specification sampling to train a denoiser f^*_{uniform}. We compare f^*_{sparse} and f^*_{dense} to f^*_{uniform} by plotting $\mathcal{L}_{f^*_{\text{sparse}}} - \mathcal{L}_{\text{ideal}}$ and $\mathcal{L}_{f^*_{\text{dense}}} - \mathcal{L}_{\text{ideal}}$ versus $\mathcal{L}_{f^*_{\text{uniform}}} - \mathcal{L}_{\text{ideal}}$. Because we report PSNR metrics, in practice we compute the previous differences for a function f by subtracting the PSNR corresponding to $\mathcal{L}_f$ from the PSNR corresponding to $\mathcal{L}_{\text{ideal}}$, which is shown in Figures 3 and 4. Speckle-Poisson results can be found in Appendix B.

For Speckle-Poisson-Gaussian noise, the set of possible specifications is too large to train an ideal denoiser for each specification, so we only report summarized results comparing adaptive training with the approximations P_{ideal}, P_f to training with uniform specification sampling in Figure 5.

5.4 DISCUSSION

Quantitative Results. Figure 3 shows that Chan et. al's sampling strategy for training denoisers can be applied directly to mixed noise distributions to achieve performance much more uniformly close to the ideal when compared a uniform specification sampling strategy.

However, Figure 4 shows that instead of constructing the entire loss landscape, we can sparsely sample the specification-loss landscape and interpolate an approximation to the true specification-loss landscape to adaptively training a denoiser. The resulting performance is more closely uniformly bounded from the ideal performance compared to the denoiser trained with uniform specification sampling.

Finally, we see in Figure 5 that we can achieve performance uniformly bound from the ideal with adaptive training even in settings where computing all the ideal losses is computational infeasible by applying our approximation method. Additional quantitative results can be found in Appendix D.

Qualitative Results. Figure 6 illustrates the denoisers trained with the different strategies (ideal, uniform, adaptive) on an example corrupted with a low amount of noise and an example corrupted with a high amount of noise. Notice that in the low-noise regime the uniform trained denoiser oversmooths the image so achieves worse performance than the adaptive trained denoiser, whereas in the high noise regime the uniform trained denoiser outperforms the adaptive trained denoiser. Addditional qualitative results can be found in Appendix F.

Time Savings. We train the DnCNNs on Nvidia GTX 1080Ti GPUs, which takes about 6 hours to train per network. We parallelized the training across 32 GPUs at a time, which means that training 10 networks takes only about 6 hours, 100 takes about 1 day, and 1000 networks would take 10 days. Thus, for Poisson-Gaussian, Speckle-Poisson, and Speckle-Gaussian noise, our sparse approximation method saves 18 hours of training time, and for Speckle-Poisson-Gaussian noise we saved almost 10 days of training time, or nearly a year in GPU hours. More detailed quantitative results can be found in Appendix D.

6 CONCLUSIONS

In this work, we demonstrate that we can leverage a polynomial approximation of the specification-loss landscape to train a denoiser to achieve performance which is uniformly bounded away from the ideal performance across a variety of problem specifications. Furthermore, with this approximation, our method demonstrates significant savings in storage space, train time, and performance when compared to baseline methods. Beyond denoising, this suggests that a polynomial approximation of the specification-loss landscape is potentially useful across a range of imaging and computer visions tasks where one can smoothly vary the difficulty of the problem.

ETHICS STATEMENT

Our work presents a method for decreasing the amount of training required to achieve consistent denoising performance across a range of specifications, which reduces the energy impact of training denoisers and possibly reduces potential harmful impacts on the climate. However, we acknowledge that modifying the training data sampling may introduce or exacerbate biases in the network's performance.

REPRODUCIBILITY STATEMENT

We completely describe the composition, preprocessing, and sampling of the corruptions of our training and testing datasets in Sections 3.1 and 5.2. We also discuss our model and training parameters in Section 5.1. Our code will be released publicly upon acceptance.

REFERENCES

Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11): 4311–4322, 2006.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16 (8):2080–2095, 2007.

Jeffrey L. Elman. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99, July 1993. doi: 10.1016/0010-0277(93)90058-4. URL https://doi.org/10.1016/0010-0277(93)90058-4.

Ruohan Gao and Kristen Grauman. On-demand learning for deep image restoration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Trans. Graph.*, 35(6), nov 2016. ISSN 0730-0301. doi: 10.1145/2980179.2982399. URL https://doi.org/10.1145/2980179.2982399.

Abhiram Gnanasambandam and Stanley Chan. One size fits all: Can we train one denoiser for all noise levels? In *International Conference on Machine Learning*, pp. 3576–3586. PMLR, 2020.

Joseph W Goodman. *Speckle phenomena in optics: theory and applications*. Roberts and Company Publishers, 2007.

Bahjat Kawar, Gregory Vaksman, and Michael Elad. Stochastic image denoising by sampling from the posterior distribution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1866–1875, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo Exploration Database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, Feb. 2017.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pp. 416–423, July 2001.

Christopher A Metzler and Gordon Wetzstein. D-vdamp: Denoising-based approximate message passing for compressive mri. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1410–1414. IEEE, 2021.

Sreyas Mohan, Zahra Kadkhodaie, Eero P Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. *arXiv preprint arXiv:1906.05478*, 2019.

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

Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (RED). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, January 2017. doi: 10.1137/16m1102884. URL https://doi.org/10.1137/16m1102884.

S. Roth and M.J. Black. Fields of experts: a framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pp. 860–867 vol. 2, 2005. doi: 10.1109/CVPR.2005.160.

Waleed Tahir, Hao Wang, and Lei Tian. Adaptive 3d descattering with a dynamic synthesis network. *Light: Science & Applications*, 11(1), February 2022. doi: 10.1038/s41377-022-00730-x. URL https://doi.org/10.1038/s41377-022-00730-x.

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*, pp. 945–948. IEEE, 2013.

Alan Q. Wang, Adrian V. Dalca, and Mert R. Sabuncu. Computing multiple image reconstructions with a single hypernetwork. *Machine Learning for Biomedical Imaging*, 1, 2022. ISSN 2766-905X. URL https://melba-journal.org/papers/2022:017.html.

Yi-Qing Wang and Jean-Michel Morel. Can a single image denoising neural network handle all levels of gaussian noise? *IEEE Signal Processing Letters*, 21(9):1150–1153, 2014. doi: 10.1109/LSP.2014.2314613.

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, 2017.

Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

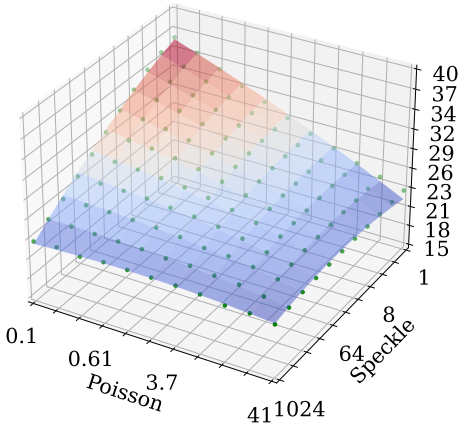

Figure 7: **Specification-loss Landscape.** PSNR, which we use as our proxy for error, versus denoising task specifications. The surfaces are highly smooth with respect to task specification.

A Polynomial Approximation

We used cross-validation to empirically determine what degree polynomial we should use to fit the specification-loss landscape. Though densely sampling the specification-loss landscape becomes intractable if its dimension is 3 or greater, we can still densely sample 2 dimensional specification-loss landscapes then subsample to simulate sparse sampling. Through this method, we compared a linear, quadratic, and cubic approximation to the specification-loss landscape and determined that a quadratic polynomial is the most suitable for approximating the specification-loss landscape in this setting. More specifically, this means for a given point $s \in S$ we approximate the corresponding loss with the function

$$P(s) = s^T As + b^T s + c,$$

where $A \in R^{n \times n}$ is symmetric, and $b, c \in \mathbb{R}^n$. We fit this quadratic using linear least squares with a ridge penalty, where the ridge penalty parameter is also determined using cross-validation. We swept over the values $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$ and settled on 0.00001.

B Speckle-Poisson Noise Results

We plot the specification-loss landscape and performance comparisons for adaptive sampling versus uniform sampling for Speckle-Poisson noise in Figures 7, 8, and 9.

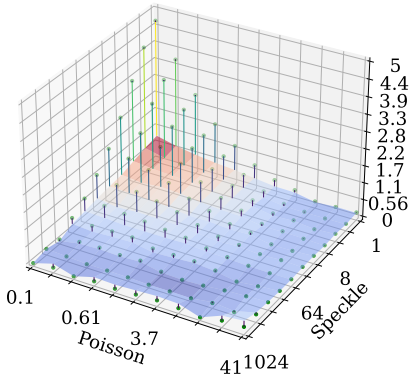

Figure 8: **Performance comparison between a network trained with adaptive training, using sparse sampling of the specification-loss landscape, and a network trained with uniform sampling of the noise levels of the training data** The surface in the above plots represents the difference in performance between the a network trained with the adaptive strategy with sparse sampling and polynomial interpolation of the specification-loss landscape and the ideal networks, and the points represent the differences in performance between a network trained with uniform sampling of the noise levels of the training data and the ideal networks. Like the networks trained with adaptive noise level sampling and dense sampling, we still achieve performance that is uniformly worse than the ideal without severe failure modes.

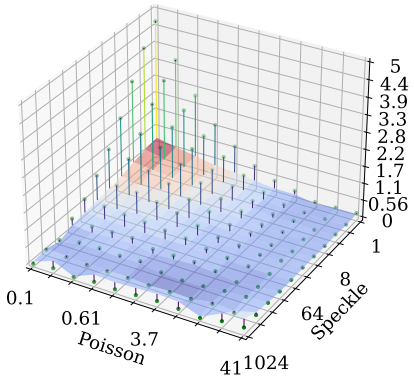

Figure 9: **Performance comparison between a network trained with adaptive training, using sparse sampling of the specification-loss landscape, and a network trained with uniform sampling of the noise levels of the training data** The surface in the above plots represents the difference in performance between the a network trained with the adaptive strategy with sparse sampling and polynomial interpolation of the specification-loss landscape and the ideal networks, and the points represent the differences in performance between a network trained with uniform sampling of the noise levels of the training data and the ideal networks. Like the networks trained with adaptive noise level sampling and dense sampling, we still achieve performance that is uniformly worse than the ideal without severe failure modes.

C Derivation of Dual Ascent iterations

We restate the derivation of the dual ascent algorithm to solve the optimization problem of equation 3 from Gnanasambandam & Chan (2020) in our notation here. First, we rewrite the optimization problem from Equation equation 3 as

$$\begin{aligned} \min_{f,t} \quad & t \\ \text{subject to} \quad & \mathcal{L}_f(\theta) - \mathcal{L}_{\text{ideal}}(\theta) \leq t, \forall \theta \in \Theta. \end{aligned}$$

Then the Lagrangian is defined as

$$L(f,t,\lambda) = t + \int_{\theta\in\Theta} \left\{\mathcal{L}_f(\theta) - \mathcal{L}_{\text{ideal}}(\theta) - t\right\} \lambda(\theta) d\theta$$

To get the dual function, we minimize over f and t:

$$\begin{aligned} g(\lambda) &= \inf_{f,t} L(f,t,\lambda) \\ &= \begin{cases} \inf_f \int (\mathcal{L}_f(\theta) - \mathcal{L}_{\text{ideal}}(\theta))\lambda(\theta) d\theta, & \text{if } \int \lambda(\theta) d\theta = 1 \\ -\infty, & \text{otherwise.} \end{cases} \end{aligned}$$

Then the dual problem is defined as

$$\begin{aligned} \lambda^* = \arg\max_{\lambda\geq 0} \inf_f & \left\{ \int (\mathcal{L}_f(\theta) - \mathcal{L}_{\text{ideal}}(\theta))\lambda(\theta) d\theta \right\} \\ \text{subject to } & \int \lambda(\sigma) d\sigma = 1. \end{aligned} \tag{9}$$

Then we can write down the dual ascent iterations as

$$f^{t+1} = \arg\min_{f\in\mathcal{F}} \left\{ \int_{\theta\in\Theta} \mathcal{L}_f(\theta)\lambda^t(\theta) d\theta \right\} \tag{10}$$

$$\lambda^{t+\frac{1}{2}} = \lambda^t + \gamma^t \left(\mathcal{L}_{f^{t+1}} - \mathcal{L}_{\text{ideal}}\right) \tag{11}$$

$$\lambda^{t+1} = \lambda^{t+\frac{1}{2}} / \int_{\theta\in\Theta} \lambda^{t+\frac{1}{2}}(\theta) d\theta, \tag{12}$$

Here, equation 10 solves the inner optimization of equation 9, fixing λ, equation 11 is a gradient ascent step for λ, and equation 12 ensures that the normalization constraint on λ is satisfied. Note that because we use PSNR constraints instead of MSE constraints, we use the step

$$\lambda^{t+\frac{1}{2}} = \lambda^t + \gamma^t \left(\frac{\mathcal{L}_{f^{t+1}}}{\mathcal{L}_{\text{ideal}}} - 1 \right) \tag{13}$$

in place of equation 11. Intuitively, the reason is because PSNR is the logarithm of the MSE loss, and a more uniform PSNR gap means that the ratio of the losses is closer to 1, which is where the $\frac{\mathcal{L}_{f^{t+1}}}{\mathcal{L}_{\text{ideal}}} - 1$ term comes from.

D Additional Quantitative Results

Tables 1, 2, 3, and 4 compare our adaptive training method, which uses an approximation of the loss-landscape, with "ideal" non-blind baselines, which are trained for specific noise parameters; with the densely sampled adaptive training procedure from Gnanasambandam & Chan (2020), which requires training specialized ideal baselines at all noise specifications beforehand; and with a network trained by uniformly sampling the specification-space. Both adaptive strategies approach the performance of the specialized networks and dramatically outperform the uniformly trained networks at certain problem specifications.

Table 5 shows that approximation of the specification-loss landscape allows us to reduce the computation time required for adaptive training by an order of magnitude.

Poisson	Gaussian	Ideal	Uniform	Adaptive-Dense	Adaptive-Sparse
0.1	0.02	35.3	31.4	34.5	34.6
0.1	0.66	22.0	22.0	21.9	21.9
41	0.02	22.9	22.9	22.9	22.9
41	0.66	21.4	21.3	21.3	21.3
2.0	0.11	27.1	26.6	27.0	27.0

Table 1: Quantitative comparison of methods on Poisson-Gaussian noise sampled at various levels, using PSNR (dB).

Speckle	Gaussian	Ideal	Uniform	Adaptive-Dense	Adaptive-Sparse
1.0	0.02	36.6	32.0	35.4	35.1
1.0	0.66	22.0	21.9	21.7	21.6
1024	0.02	23.1	23.1	22.6	22.4
1024	0.66	21.5	21.4	20.8	20.7
32	0.11	27.4	27.0	27.2	27.2

Table 2: Quantitative comparison of methods on Speckle-Gaussian noise sampled at various levels, using PSNR (dB).

Speckle	Poisson	Ideal	Uniform	Adaptive-Dense	Adaptive-Sparse
1.0	0.1	36.1	31.5	35.3	35.3
1.0	41	23.0	22.9	22.9	22.9
1024	0.1	23.0	23.1	23.0	22.9
1024	41	22.0	21.8	21.6	21.6
32	2.0	27.8	27.3	27.6	27.7

Table 3: Quantitative comparison of methods on Speckle-Poisson noise sampled at various levels, using PSNR (dB).

Speckle	Poisson	Gaussian	Ideal	Uniform	Adaptive-Sparse
1	0.1	0.02	34.7	29.3	33.4
1	0.1	0.66	22.0	21.9	21.5
1	41	0.02	23.0	22.9	22.7
1	41	0.66	21.4	21.3	20.8
1024	0.1	0.02	23.2	23.2	22.6
1024	0.1	0.66	21.5	21.4	20.7
1024	41	0.02	22.0	21.9	21.2
1024	41	0.66	21.0	20.8	20.0
64	2.3	0.54	25.8	25.5	25.5

Table 4: Quantitative comparison of methods on Speckle-Poisson-Gaussian noise sampled at various levels, using PSNR (dB).

	Uniform	Adaptive-Dense		Adaptive-Sparse	
		Baselines	Adaptive	Baselines	Adaptive
Poisson-Gauss	6hr 27min	36d 20hr	7hr 37min	3d 16hr	7hr 12min
Speckle-Gauss	7hr 48min	30d 22hr	5hr 48min	3d 1hr	5hr 46min
Speckle-Poisson	8hr 0min	27d 2hr	5hr 55min	2d 17hr	5hr 47min
Speckle-Poisson-Gauss	7hr 30min	X	X	34d 5hr	8hr 34min

Table 5: Comparing training times between the various methods. Note that for the adaptive training methods, there are two parts: training the ideal baseline networks to generate the specification-loss landscapes as well as the adaptive training itself.

E HIGH-DIMENSIONAL SPECIFICATION-SPACES

In this work, we demonstrated how our quadratic approximation of the specification-loss landscape allowed us to adaptively train a blind image denoiser with orders of magnitude less compute

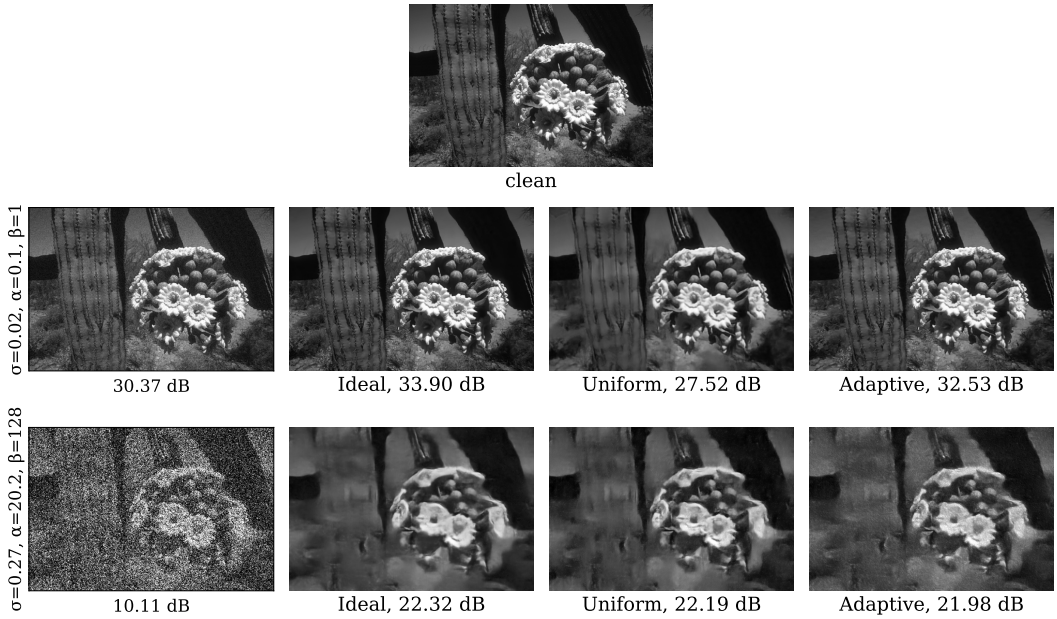

Figure 10: **Qualitative comparisons.** Comparison between the performance of the ideal, uniform-trained, and adaptive-trained denoisers on a sample image corrupted with a low amount of noise and corrupted with a high amount of noise. Our adaptive blind training strategy performs only marginally worse than an ideal, non-blind baseline when applied to "easy" problem specifications, and significantly better than the uniform baseline, while also being only marginally worse than an ideal baseline and uniform baseline under "hard" problem specifications.

than Gnanasambandam & Chan (2020). The key distinction between the two methods is that our approach only needs to sparsely sample the specification-loss landscape, in order to form a quadratic approximation of the landscape, whereas Gnanasambandam & Chan (2020) needs to evaluate this landscape at all specifications of interest.

As one increases the number of specifications, n, needed to describe this landscape ($n = 1$ for Gaussian noise, $n = 2$ for Poisson-Gaussian noise, $n = 3$ for Poisson-Gaussian-Speckle noise, ...), the number points needed to densely sample the landscape grows exponentially. Fortunately, the number of samples needed to fit a quadratic to this landscape only grows quadratically with n: The number of possible nonzero coefficients, i.e., unknowns, of a quadratic of n variables is $\binom{n+2}{2} = \frac{(n+1)(n+2)}{2}$ and thus one can uniquely specify this function from $\frac{(n+1)(n+2)}{2} + 1$ non-degenerate samples.

F Additional Qualitative Results

Additionally qualitative results are presented in Figure 10 and 11.

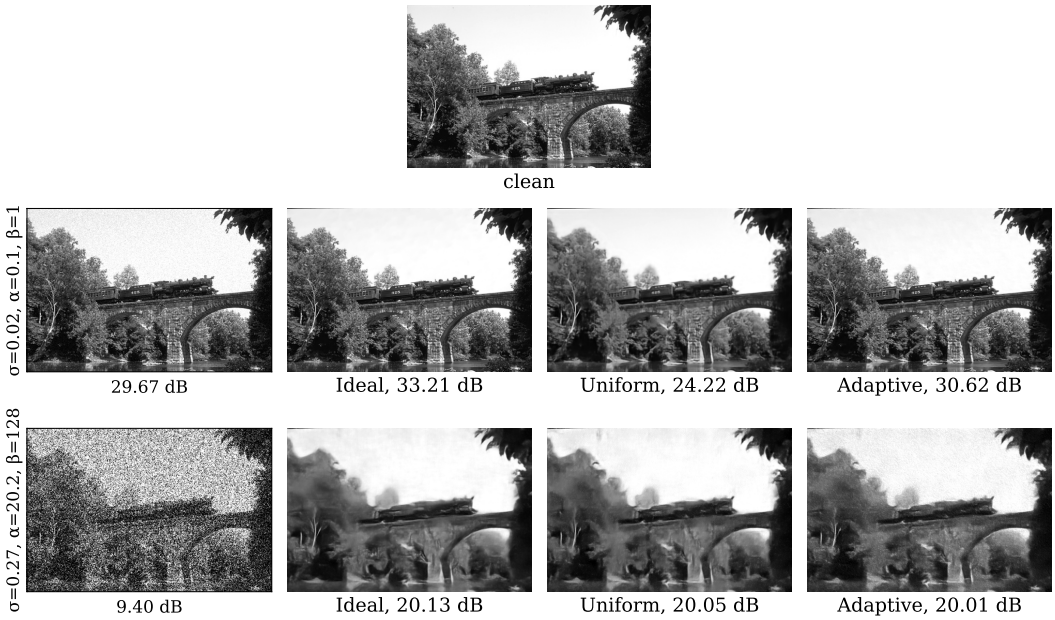

Figure 11: **Qualitative comparisons.** Comparison between the performance of the ideal, uniform-trained, and adaptive-trained denoisers on a sample image corrupted with a low amount of noise and corrupted with a high amount of noise. Our adaptive blind training strategy performs only marginally worse than an ideal, non-blind baseline when applied to "easy" problem specifications, and significantly better than the uniform baseline, while also being only marginally worse than an ideal baseline and uniform baseline under "hard" problem specifications.