Towards a Mechanistic Explanation of Diffusion Model Generalization

Matthew Niedoba[1,2], **Berend Zwartsenberg**[2], **Kevin Murphy**[3], **Frank Wood**[1,2,4]
[1]University of British Columbia, [2]Inverted AI
[3]Google Deepmind, [4]Alberta Machine Intelligence Institute
mniedoba@cs.ubc.ca

Abstract

We propose a mechanism for diffusion generalization based on local denoising operations. Through analysis of network and empirical denoisers, we identify local inductive biases in diffusion models. We demonstrate that local denoising operations can be used to approximate the optimal diffusion denoiser. Using a collection of patch-based, local empirical denoisers, we construct a denoiser which approximates the generalization behaviour of diffusion model denoisers over forward and reverse diffusion processes.

1 Introduction

Diffusion models (17; 6; 18) have been widely adopted and are the *de facto* standard for modelling image (14) and video (5) data due to their high sample quality and generalization abilities. With sufficient data, diffusion models produce samples which are distributionally similar to their training set, but are not exact copies of training data (21).

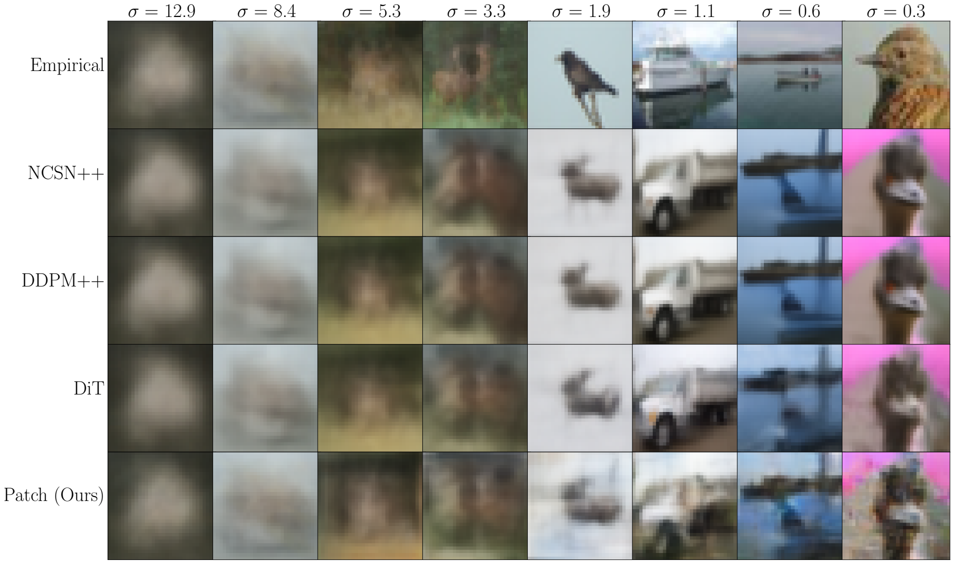

Figure 1: Empirical, diffusion model, and our patch-based denoiser outputs for shared $\mathbf{z}$ drawn from the reverse diffusion process. For large σ, network denoisers match the empirical denoiser. However, as σ decreases, the networks generalize and deviate from the empirical denoiser. Our method resembles the network outputs for all σ, offering a plausible mechanism for diffusion generalization.

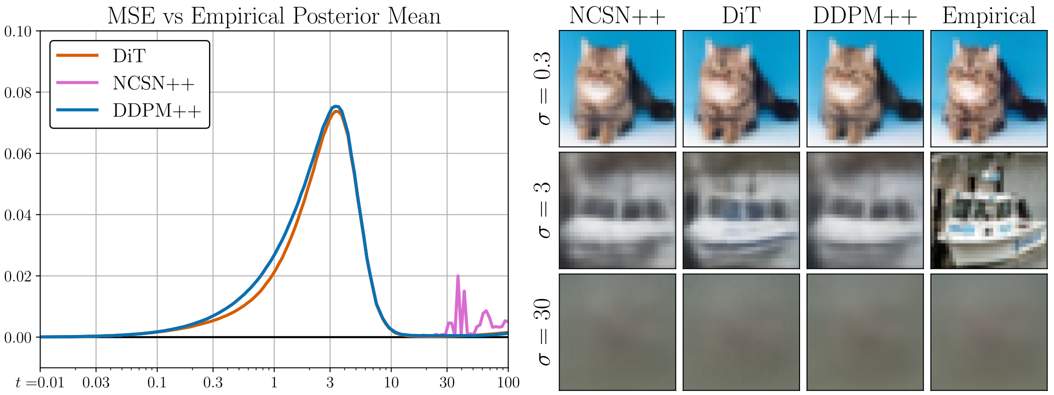

Figure 2: **Left**: Mean squared error between empirical and network denoisers for three architectures on CIFAR-10. **Right**: Comparison of network and empirical denoiser for a shared $\mathbf{z} \sim p_t(\mathbf{z}|\mathbf{x})$ at three σ values. Network estimators have low error for small and large σ, but large errors around $\sigma = 3$. At this point, each network varies substantially from the empirical denoiser *in the same way*.

This behaviour is remarkable, as linear increases in data dimensionality require exponentially more training samples to model the data density (1). Avoiding this curse of dimensionality necessitates inductive biases which enable generalization form sparse examples (4). Recently, research has found that diffusion models produce similar samples irrespective of architecture, optimization, or diffusion hyperparameters(21). This has been replicated even when networks are trained on disjoint subsets of the same dataset (7), suggesting inductive biases common to the structure of diffusion itself.

Generalization in diffusion model samples is the product of compounding generalizations over repeated denoising iterations of their sampling procedure. At each step, the mathematical construction of this procedure defines an optimal denoiser which is the unique minimizer of the training objective of diffusion models. However, the generative model which this empirical denoiser defines produces *no generalization* (11; 16). Thankfully, diffusion model denoisers do not perfectly estimate this empirical denoiser, as it is through approximation errors that diffusion generalization emerges (20).

In this work, we study identify diffusion model inductive biases through analysis of network denoiser approximation errors. Using this approach, we find that several network denoisers make similar errors, primarily over an intermediate portion of the diffusion process. Analyzing the response of network and empirical denoisers to input modifications within this region, and though gradient analysis, we find strong evidence of a shared local inductive bias common to all network denoisers.

From this conclusion, we hypothesize that diffusion models generalize through a local denoising mechanism which we approximate with a patch-based empirical denoiser. We show that for a majority of the diffusion process, the outputs of patch and full empirical denoisers are equivalent. Further, we show that in settings where network denoisers generalize, their output is approximated well by patch-based denoisers. Finally, through compositing our patch-based empirical denoisers, we are able to approximate the the output of network denoisers, providing strong evidence to support our diffusion generalization mechanism.

2 Related Work

Diffusion generalization Several prior works have studied diffusion generalization. Comparing a variety of networks, (21) finds diffusion models produce consistent samples despite differences in architecture and training. Both (11; 19) find that diffusion models only deviate from empirical denoisers for intermediate σ. Futher, (11) identify that errors in this region are primarily responsible for diffusion generalization. Similarly, (20) attributes diffusion generalization to "slight differences" between the network and empirical denoisers. (7) suggests that generalization stems from adaptive bases found in the eigenspace of diffusion models. However, they do not consider how trained models deviate from empirical denoisers. Similar to our work, Closed-form diffusion models (16) produces a mechanism for diffusion generalization by intentionally biasing the empirical denoiser with a spectral bias (13). However, they do not validate this inductive bias choice by studying network denoisers.

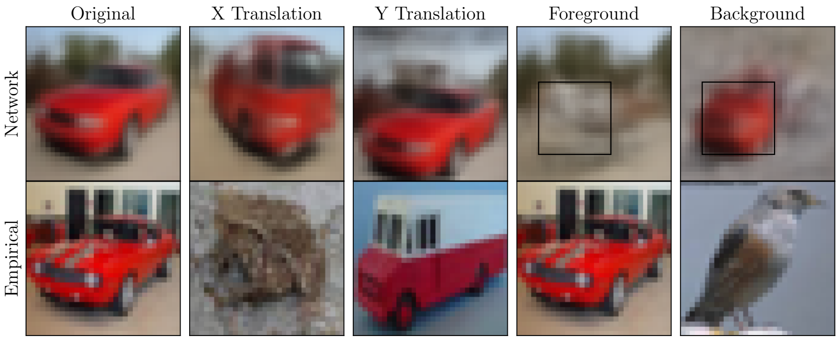

Figure 3: Effect of transformations to $\mathbf{z}$ on network and empirical denoiser outputs for $\sigma = 2$. The transformations are, from left to right, **(a)** no transformation, **(b)** a 5 pixel horizontal translation, **(c)** a 5 pixel vertical translation, **(d)** replacement of pixels of $\mathbf{z}$ within the black box with $\sigma = 2$ Gaussian noise, **(e)** replacement of pixels of $\mathbf{z}$ outside of the black box with $\sigma = 2$ Gaussian noise.

Patch Denoising Outside of the context of diffusion models, many methods exist which use patch priors to denoise images. Field of Experts (15) suggests modelling images as Markov Random Fields to learn a prior over patches. Using expected patch log likelihood (22) and half quadratic splitting methods (22; 3; 2), patch priors can be used to produce maximum a priori (MAP) estimates for denoising problems. Notably, the classical denoising problem is slightly different than the diffusion denoising problem which estimates an average over possible samples and not a MAP estimate.

3 Background

The basis of diffusion models is the diffusion process in which Gaussian noise is gradually added to a data distribution $p(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d$ in a so-called forward process. At any point $t \in (0, T]$, the diffusion process defines a marginal latent variable distribution $p_t(\mathbf{z}) = \int p_t(\mathbf{z}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$, $\mathbf{z} \in \mathbb{R}^d$ where $p_t(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}; s(t)\mathbf{x}, \sigma(t)^2\mathbf{I}\right)$ is the marginalization of the process' transition kernels up to time t. Functions $s(t)$ and $\sigma(t)$ are chosen such that $p_T(\mathbf{z}) \approx \pi(\mathbf{z}) = \mathcal{N}\left(\mathbf{z}; \mathbf{0}, \sigma(T)^2\mathbf{I}\right)$. We use the variance exploding formulation (18; 8) with $s(t) = 1$ and $\sigma(t) = t$ hereafter.

Diffusion models reproduce the data distribution $p(\mathbf{x})$ through a reverse diffusion process which is constructed to match the marginal latent distributions $p_t(\mathbf{z})$ for all t. The probability flow ODE (18; 8) is one way of describing this reverse diffusion process

$$d\mathbf{z} = -t\nabla_{\mathbf{z}} \log p_t(\mathbf{z})dt. \tag{1}$$

To draw samples from $p(\mathbf{x})$, diffusion models integrate Eq. (1), from $t = T$ to 0, with initial value $\mathbf{z} \sim \pi(\mathbf{z})$ using numerical integration. Critically, this sampling procedure requires repeated estimation of the score function $\nabla_{\mathbf{z}} \log p_t(\mathbf{z})$ which has the form

$$\nabla_{\mathbf{z}} \log p_t(\mathbf{z}) = \frac{\mathbb{E}\left[\mathbf{x}|\mathbf{z}, t\right] - \mathbf{z}}{t^2}. \tag{2}$$

From Eq. (2), estimating the score can be reduced to estimating the posterior mean $\mathbb{E}\left[\mathbf{x}|\mathbf{z}, t\right]$, an operation referred to a *denoising*. In practice, the analytic form of $p(\mathbf{x})$ is generally unknown, which prevents exact computation of the posterior $p_t(\mathbf{x}|\mathbf{z})$ and therefore $\mathbb{E}\left[\mathbf{x}|\mathbf{z}, t\right]$. Instead, diffusion models use deep neural networks to approximate the score function. Using an *empirical* data distribution $p_{\text{data}}(\mathbf{x}) = \frac{1}{N}\sum_{\mathbf{x}^{(i)} \in \mathcal{D}} \delta(\mathbf{x} - \mathbf{x}^{(i)})$, with dataset $\mathcal{D} = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)} \mid \mathbf{x}^{(i)} \sim p(\mathbf{x})\right\}$, diffusion models optimize a neural network denoiser $D_\theta(\mathbf{z}, t)$ by minimizing

$$\mathop{\mathbb{E}}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \mathbf{z} \sim p_t(\mathbf{z}|\mathbf{x}), t \sim p(t)} \left[\lambda(t) \left\|\mathbf{x} - D_\theta(\mathbf{z}, t)\right\|_2^2\right] \tag{3}$$

The minimizer of Eq. (3) is the *empirical* posterior mean which we also term the empirical denoiser

$$\mathbb{E}\left[\mathbf{x}^{(i)}|\mathbf{z}, t\right] = \sum_{\mathbf{x}^{(i)} \in \mathcal{D}} p_t(\mathbf{x}^{(i)}|\mathbf{z})\mathbf{x}^{(i)}, \quad p_t(\mathbf{x}^{(i)}|\mathbf{z}) = \frac{p_t\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}{\sum_{\mathbf{x}^{(j)} \in \mathcal{D}} p_t\left(\mathbf{z}|\mathbf{x}^{(j)}\right)}. \tag{4}$$

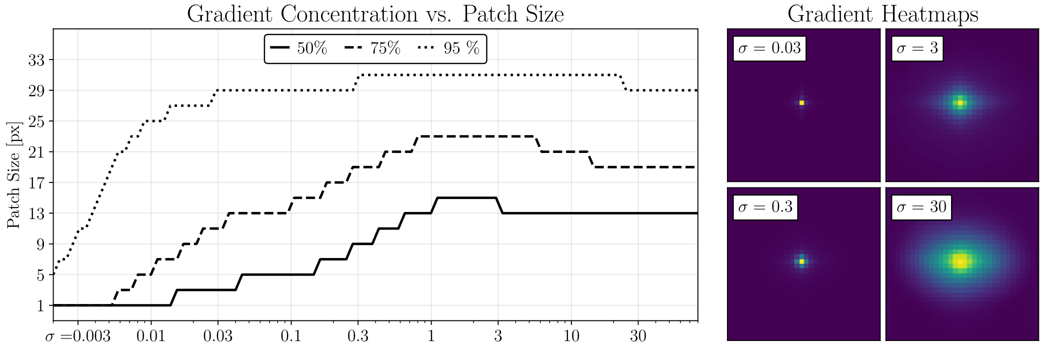

Figure 4: **Left**: Comparison of square patch sizes required to capture 50, 75, and 95% of the total absolute gradient of network denoiser output with respect to $\mathbf{z}$. **Right**: Illustration of average absolute gradient of a center pixel of the network denoiser output with respect to $\mathbf{z}$.

4 Inductive Biases of Network Denoisers

To understand the generalization of diffusion models, we must understand and characterize network denoiser approximation errors. To begin, we simply compare the denoiser outputs of three near SoTA diffusion models trained on CIFAR-10 (10) to the empirical denoiser of that dataset. We evaluate models parameterized by NCSN++ (18; 8), DDPM++(18; 8) and DiT (12) architectures over 150 values of $\sigma \in [0.01, 100]$, drawing 10,000 $\mathbf{z}$ samples from the forward process for each σ value.

Figure 2 plots the mean squared error (MSE) between network and empirical denoisers. We observe that across all architectures, denoisers have low bias for both small and large values of σ but substantial bias for $\sigma \in [0.3, 10]$. This region corresponds with the results of (11) who found that diffusion generalization can be attributed to this intermediate region.

Visualizations of the denoiser outputs are presented in the right portion of Fig. 2. We note that only the middle $\sigma = 3$ row has significant differences between the empirical and network denoisers. At this noise level, we observe that all three networks make *qualitatively similar approximation errors*, despite differences between the transformer architecture of DiT and the convolutional U-Net architectures of DDPM++ and NCSCN++. That denoiser outputs appear independent to network architecture suggests that diffusion model approximation errors are not random optimization artifacts, but the result of a shared inductive bias unrelated to the neural network architecture. This would also support the findings of (20) – that diffusion models produce similar samples when with the same $\mathbf{z} \sim \pi(\mathbf{z})$ despite changes in network architecture, training, or diffusion hyperparameters.

We next attempt to identify common diffusion inductive biases by comparing network and empirical denoisers' responses to input transformations. Figure 3 visualizes DDPM++ (18; 8) and empirical denoiser outputs after transforming the original $\mathbf{z} \sim p_t(\mathbf{z}|\mathbf{x})$. We observe that small changes to $\mathbf{z}$ can result in substantially different empirical posterior means. In contrast, the network outputs are remarkably consistent. In four of five cases, network denoisers still produce a red vehicle, including a near-identical car in response to the y-translation. In general, local changes in input produce local changes in the network denoiser output. For example, by replacing the contents of the black square with random noise, the network "deletes" the red car from the image while maintaining the background structure.

Figure 3, provides preliminary evidence that network denoisers are predominantly *locally* sensitive to changes in $\mathbf{z}$, while the empirical denoiser is *globally* sensitive to any such changes. This global sensitivity is justified by Eq. (4). Since $p_t(\mathbf{x}^{(i)}|\mathbf{z}) \propto p_t(\mathbf{z}|\mathbf{x}^{(i)})$ and $p_t(\mathbf{z}|\mathbf{x}^{(i)})$ is Gaussian, the posterior probability of any $\mathbf{x}^{(i)}$ is related to the ℓ_2 distance between $\mathbf{x}^{(i)}$ and $\mathbf{z}$ and therefore, the individual values of every pixel of $\mathbf{z}$.

We further investigate the local inductive bias of network denoisers by measuring the sensitivity of the DDPM++ denoiser to each input pixel through $|\nabla_{\mathbf{z}} D_\theta(\mathbf{z}, t)_{x,y}|$ – the absolute gradient of the network denoiser output at pixel (x, y) with respect to the input $\mathbf{z}$. In Fig. 4, we evaluate the network

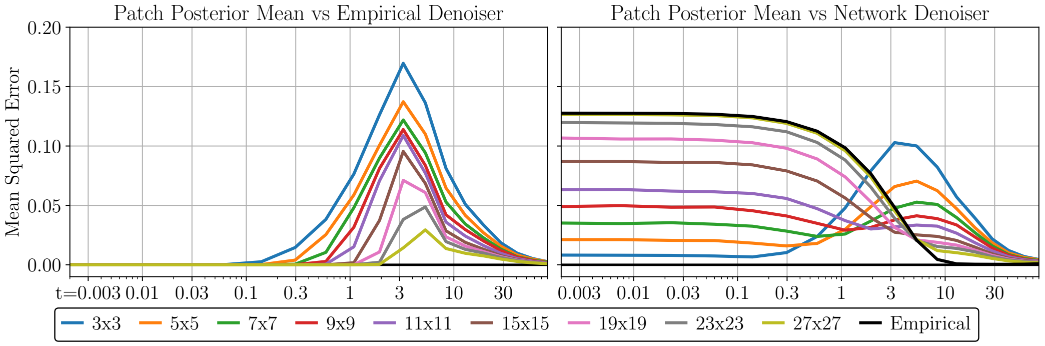

Figure 5: **Left**: Comparison of posterior means with varying patch sizes to the full 32x32 empirical mean on forward process samples. **Right**: Comparison of patch posterior means with varying patch sizes to DDPM++ denoiser on $\mathbf{z}$ drawn from the sampling procedure.

denoiser's input sensitivity over a range of σ values through the gradients of 10,000 $\mathbf{z} \sim p_t(\mathbf{z}|\mathbf{x})$ for each σ.

Figure 4 confirms that network denoisers demonstrate strong local inductive bias. The right portion of Fig. 4 visualizes the average absolute input gradient for pixel (15,15) of the input $\mathbf{z}$. At $\sigma = 0.03$, the input gradient is focused almost exclusively on the same spatial location as the output pixel. As σ increases, so does the area in which the input gradient is concentrated.

The local bias of network denoisers is reflected quantitatively in the left portion of Fig. 4 which plots the side length of a square patch required to capture a fixed percentage of the total $|\nabla_{\mathbf{z}} D_\theta\left(\mathbf{z}, t\right)_{i,j}|$ across varying σ. At all levels, there is strong evidence of local inductive bias, with the majority of the input gradient is concentrated within a local 15×15 pixel region around the output pixel. We also observe that the strength of the local bias is anti-correlated with σ. While a 13×13 patch is required to capture 50% of the input gradient at $\sigma = 30$, the same percentage can be accounted for in a 3×3 patch at $\sigma = 0.03$.

5 Patch Posterior Denoising

Section 4 presents strong evidence that diffusion models generalize through local inductive bias, irrespective of network architecture. However, Fig. 2 also shows that network denoisers accurately estimate the empirical posterior mean for small and large values of σ, despite the inherent global sensitivity of this quantity. To resolve this apparent contradiction, we hypothesize that diffusion models perform local operations whose combined result is generally equivalent to empirical denoising.

Specifically, we propose that diffusion models generalize through a local denoising operation. We can approximate such an operation with patch posterior denoising – estimating the empirical posterior mean of a cropped patch of the input. Formally, let $\mathbf{C}_{x,y,p} \in \{0,1\}^{3p^2 \times d}$ be a specific cropping matrix such that $\mathbf{C}_{x,y,p}\mathbf{x}^{(i)}$ is the $p \times p$ patch of $\mathbf{x}^{(i)}$ with upper left corner at pixel (x, y). For a patch size p, let $\mathcal{C}_p = \{\mathbf{C}_{x,y,p} \mid x \in \{0, \ldots, 32-p\}, y \in \{1, \ldots, 32-p\}\}$ be the set of all such cropping matrices. Then, for any such cropping matrix $\mathbf{C} \in \mathcal{C}_p$, which produces patches $\mathbf{x}^{(i)}_{\mathbf{C}} = \mathbf{C}\mathbf{x}^{(i)}$ and $\mathbf{z}_{\mathbf{C}} = \mathbf{C}\mathbf{z}$, we define the patch posterior mean with $p_t(\mathbf{z}_{\mathbf{C}} \mid \mathbf{x}^{(i)}_{\mathbf{C}}) = \mathcal{N}(\mathbf{z}_{\mathbf{C}}; \mathbf{x}^{(i)}_{\mathbf{C}}, \sigma(t)^2\mathbf{I})$ as

$$\mathbb{E}\left[\mathbf{x}^{(i)}_{\mathbf{C}} \mid \mathbf{z}_{\mathbf{C}}, t\right] = \sum_{\mathbf{x}^{(i)} \in \mathcal{D}} p_t\left(\mathbf{x}^{(i)}_{\mathbf{C}} \mid \mathbf{z}_{\mathbf{C}}\right) \mathbf{x}^{(i)}_{\mathbf{C}}, \quad p_t(\mathbf{x}^{(i)}_{\mathbf{C}} \mid \mathbf{z}_{\mathbf{C}}) = \frac{p_t\left(\mathbf{z}_{\mathbf{C}} \mid \mathbf{x}^{(i)}_{\mathbf{C}}\right)}{\sum_{\mathbf{x}^{(j)} \in \mathcal{D}} p_t\left(\mathbf{z}_{\mathbf{C}} \mid \mathbf{x}^{(j)}_{\mathbf{C}}\right)} \tag{5}$$

We have hypothesized that denoisers perform local operations which are equivalent to global posterior mean estimation. However, in general, $\mathbb{E}[\mathbf{x}^{(i)}_{\mathbf{C}} \mid \mathbf{z}_{\mathbf{C}}, t] \neq \mathbf{C}\mathbb{E}[\mathbf{x}^{(i)} \mid \mathbf{z}, t]$ because $p_t(\mathbf{x}^{(i)}_{\mathbf{C}} \mid \mathbf{z}_{\mathbf{C}}) \neq p_t(\mathbf{x}^{(i)} \mid \mathbf{z})$. Why then would we expect network denoisers to use local posterior mean estimates to estimate the global posterior mean?

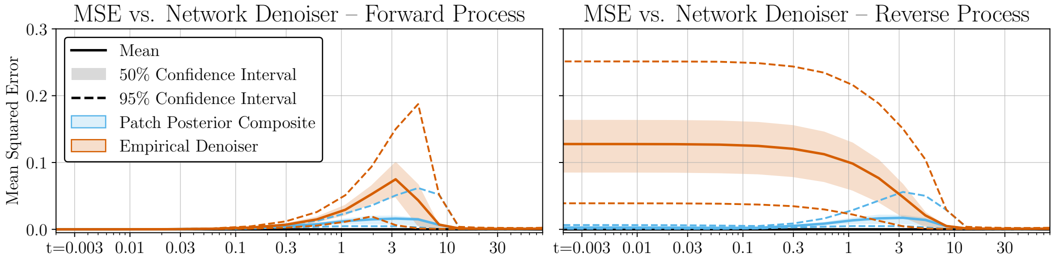

Figure 6: Mean squared error between patch posterior composite and DDPM++(18; 8) network denoiser over varying σ, with $\mathbf{z}$ sampled from the forward and reverse processes. Across all σ, the patch posterior composite is a better estimator of the network denoiser than the empirical denoiser.

Critically, there are two cases when equality does exist. For sufficiently small σ, $p_t(\mathbf{x}_{\mathbf{C}}^{(i)} \mid \mathbf{z}_{\mathbf{C}})$ and $p_t(\mathbf{x}^{(i)}|\mathbf{z})$ both approximate Dirac distributions on the same $\mathbf{x}^{(i)} \in \mathcal{D}$. Similarly, as σ becomes large, both posteriors will approach a uniform distribution over $\mathcal{D}$. We note that these two cases correspond to the regions of Fig. 2 in which network estimators accurately estimate the empirical posterior mean.

The left portion of Fig. 5 empirically confirms these cases of equality. It plots the MSE between $\mathbb{E}[\mathbf{x}_{\mathbf{C}}^{(i)} \mid \mathbf{z}_{\mathbf{C}}, t]$ and $\mathbf{C}\mathbb{E}[\mathbf{x}^{(i)} \mid \mathbf{z}, t]$, averaged across $\mathbf{C} \in \mathcal{C}_p$ and 10,000 $\mathbf{z} \sim p_t(\mathbf{z}, \mathbf{x})$ for varying σ and patch sizes p. For both $\sigma < 0.1$ and large σ, the two posterior means are similar, regardless of patch size. Additionally as σ increases, larger patch sizes are required to accurately estimate the empirical mean. This matches the correlation between local sensitivity and σ observed in Fig. 4.

Although local denoising approximates the empirical denoiser well for large and small σ in the forward process, this is not true of $\mathbf{z}$ sampled from the reverse process. If network denoisers approximate the empirical denoiser with local denoising, we would expect that they would also use this mechanism for the reverse process. In the right subplot of Fig. 5 we see that although local denoisers are poor estimators of the empirical score over the reverse process, for $\sigma < 5$, they are a better approximator of the network denoiser. Further, our results corroborate the anti-correlation of σ and the degree of local bias observed in Fig. 4. As σ increases, the network denoiser is best approximated by patch denoisers of increasingly large patch sizes.

The similarity between network denoisers and patch posterior means motivate us to attempt reproduction of the full network denoiser output through local patch posterior means. For a patch size p, we define our patch posterior composite as

$$D_p(\mathbf{z}, t) = \left(\sum_{\mathbf{C} \in \mathcal{C}_p} \mathbf{C}^\top \mathbf{C}\right)^{-1} \left(\sum_{\mathbf{C} \in \mathcal{C}_p} \mathbf{C}^\top \mathbb{E}\left[\mathbf{x}_{\mathbf{C}}^{(i)} \mid \mathbf{z}_{\mathbf{C}}, t\right]\right). \tag{6}$$

The patch posterior composite of Eq. (6) estimates the patch posterior mean of every overlapping $p \times p$ square patch of the input $\mathbf{z}$. The composite is produced by averaging each of these patch posterior means together, normalizng each pixel by it's total number of overlapping patches. A visualization of the patch posterior composite is provided in Fig. 7.

We evaluate the performance of our patch posterior composite in Fig. 6 which evaluates the MSE of our composite against the DDPM++ denoiser across forward and reverse process $\mathbf{z}$ samples for 18 σ values based on the Huen sampler of (8). For every σ, we use a fixed patch size determined by the minimum error of Fig. 5. We find that for both forward and reverse processes, our method is better than the empirical denoiser at estimating the network denoiser. This is also shown in Fig. 1 where despite its simplicity, we find that our method closely resembles the network denoiser outputs. We present additional composite examples in Appendix C.

6 Conclusions

Our work identifies a mechanism for the generalization behaviour of diffusion models through local denoising. We find network denoisers exhibit a local inductive bias whose strength is anti-correlated with σ. We outline how diffusion models can accurately estimate the empirical denoiser during training using local denoising operations. Using a patch-based approximation of these local operations we produce a composite denoiser which is a better estimator of network denoisers than the empirical denoiser, even when network denoisers exhibit generalization. Unfortunately, our composite is a crude approximation to the flexibility of deep neural networks. Integrating Eq. (1) using our patch-based composite does not produce realistic samples, as artifacts from the independent denoisers at each step compound to produce unrealistic samples. In future work, we aim to improve the quality of our composite by dynamically selecting patch size, or by using non-square local denoisers.

7 Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada CIFAR AI Chairs Program, Inverted AI, MITACS, and Google. This research was enabled in part by technical support and computational resources provided by the Digital Research Alliance of Canada Compute Canada (alliancecan.ca), the Advanced Research Computing at the University of British Columbia (arc.ubc.ca), and Amazon.

References

[1] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[2] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[3] Roy Friedman and Yair Weiss. Posterior sampling for image restoration using explicit patch priors. *arXiv preprint arXiv:2104.09895*, 2021.

[4] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068, 2022.

[5] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965, 2022.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[7] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. *arXiv preprint arXiv:2310.02557*, 2023.

[8] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[9] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[11] Matthew Niedoba, Dylan Green, Saeid Naderiparizi, Vasileios Lioutas, Jonathan Wilder Lavington, Xiaoxuan Liang, Yunpeng Liu, Ke Zhang, Setareh Dabiri, Adam Ścibior, et al. Nearest neighbour score estimators for diffusion generative models. *arXiv preprint arXiv:2402.08018*, 2024.

[12] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[13] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[15] Stefan Roth and Michael 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, pages 860–867. IEEE, 2005.

[16] Christopher Scarvelis, Haitz Sáez de Ocáriz Borde, and Justin Solomon. Closed-form diffusion models. *arXiv preprint arXiv:2310.12395*, 2023.

[17] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[18] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[19] Yilun Xu, Shangyuan Tong, and Tommi Jaakkola. Stable target field for reduced variance score estimation in diffusion models. *arXiv preprint arXiv:2302.00670*, 2023.

[20] Mingyang Yi, Jiacheng Sun, and Zhenguo Li. On the generalization of diffusion model. *arXiv preprint arXiv:2305.14712*, 2023.

[21] Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *Forty-first International Conference on Machine Learning*, 2023.

[22] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 international conference on computer vision*, pages 479–486. IEEE, 2011.

A Patch Composite Algorithm Details

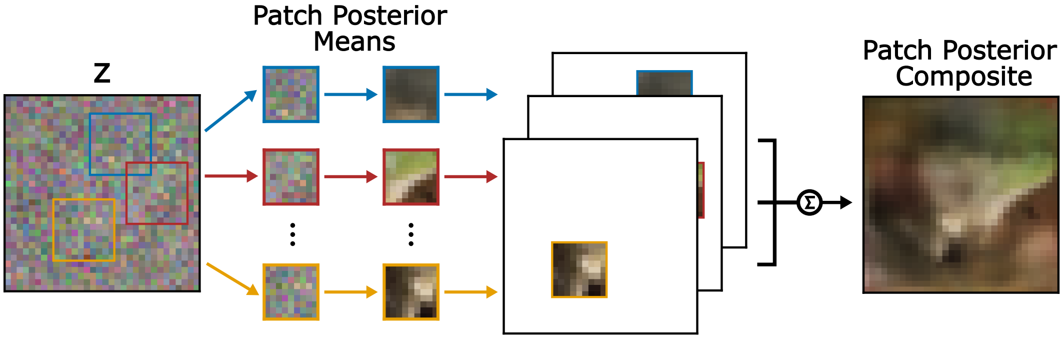

Figure 7: Visual explanation of the patch posterior composite presented in Eq. (6). To produce a composite, we decompose **z** into equal-sized overlapping square patches. For each patch, we compute the patch posterior mean via Eq. (5). Then, we composite the resulting means into one image and normalize each pixel by dividing by the number of patch posterior means which overlap that pixel.

We visually illustrate the patch posterior compoiste algorithm in Fig. 7. We produce composites using values based on the Heun ODE solver of (8) with $\rho = 7$, $n = 18$, $\sigma_min = 0.002$, $\sigma_max = 80$. For each σ, we use a fixed patch size, described in Table 1

σ	80	57	41	28	19	13	8.4	5.3	3.3	1.9	1.1	0.59	0.30	0.14	0.06	0.02	0.01	0.002
Patch Size	32	32	32	32	32	32	32	23	15	11	7	5	3	3	3	3	3	3

Table 1: Patch sizes used to produce patch posterior composites for varying σ.

B Architecture Details

For the NSCN++ and DDPM++ architectures, we utilize the pretrained unconditional model checkpoints provided by (8). For the Diffusion Transformer, we adopt the code of (12), utilizing the EDM preconditioning scheme and data augmentation pipeline. We trained a DiT-B/4 network from scratch on 200 million examples using the Adam (9) optimizer and the hyperparamters given in Table 2

Hyperparameter	**Value**
Batch Size	512
Learning Rate	0.0001
β_1	0.9
β_2	0.999
ϵ	1E-8
Patch Size	4
Heads	12
Hidden Size	768
Transformer Blocks	12
Dropout Ratio	0.12
Augmentation Rate	0.12

Table 2: Hyperparameters for DiT training on CIFAR-10

C Additional Patch Posterior Composite Examples

We present additional examples of denoiser outputs for the same σ values presented in Fig. 1. All **z** are drawn from the reverse process of the DDPM++ network denoiser.

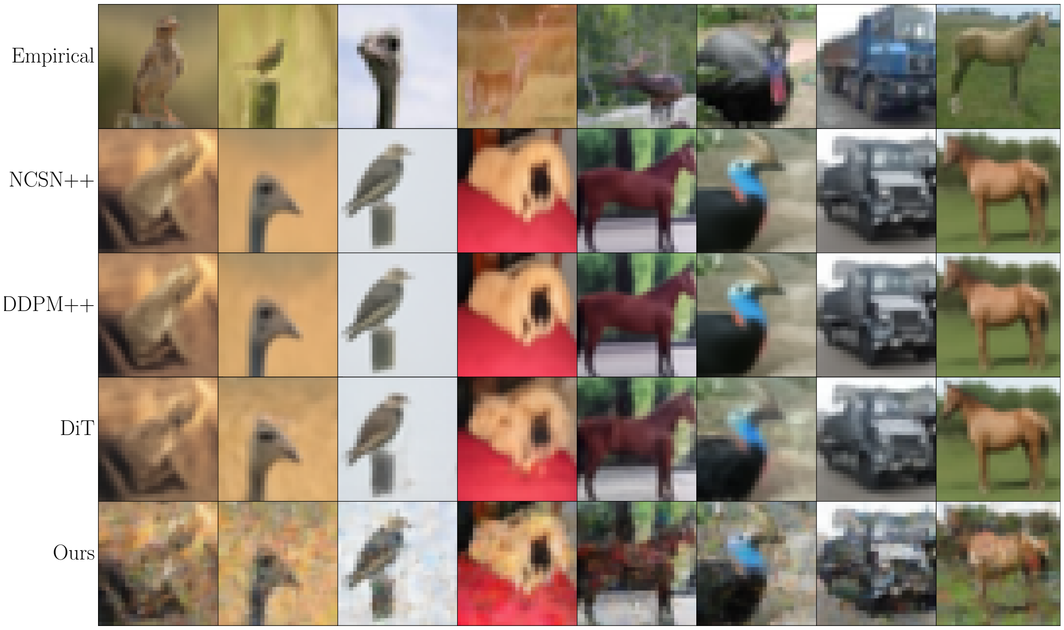

Figure 8: Additional denoiser outputs for $\sigma = 0.3$

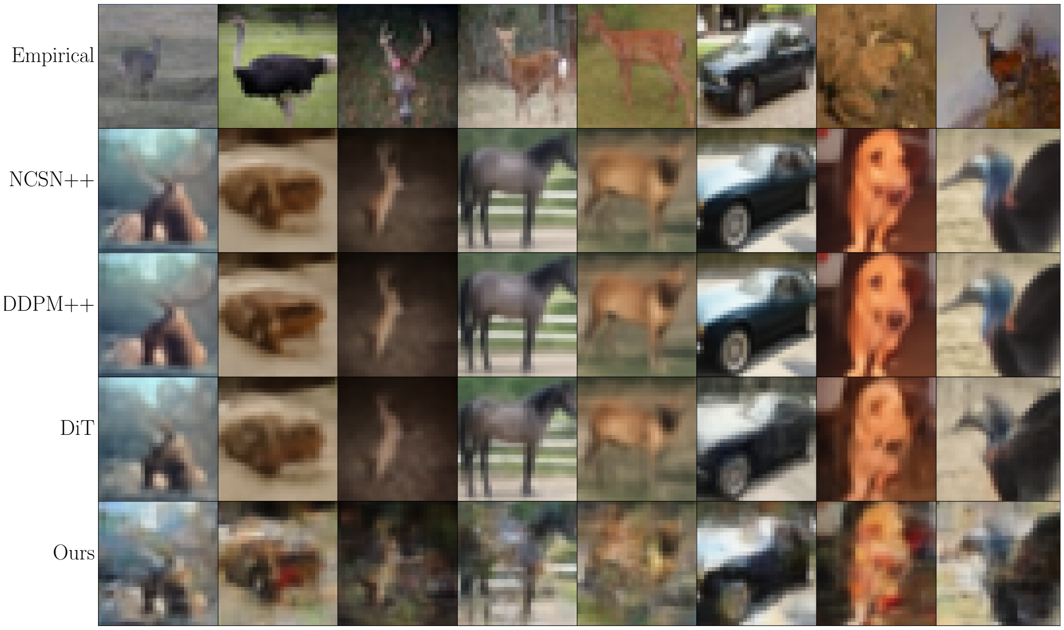

Figure 9: Additional denoiser outputs for $\sigma = 0.6$

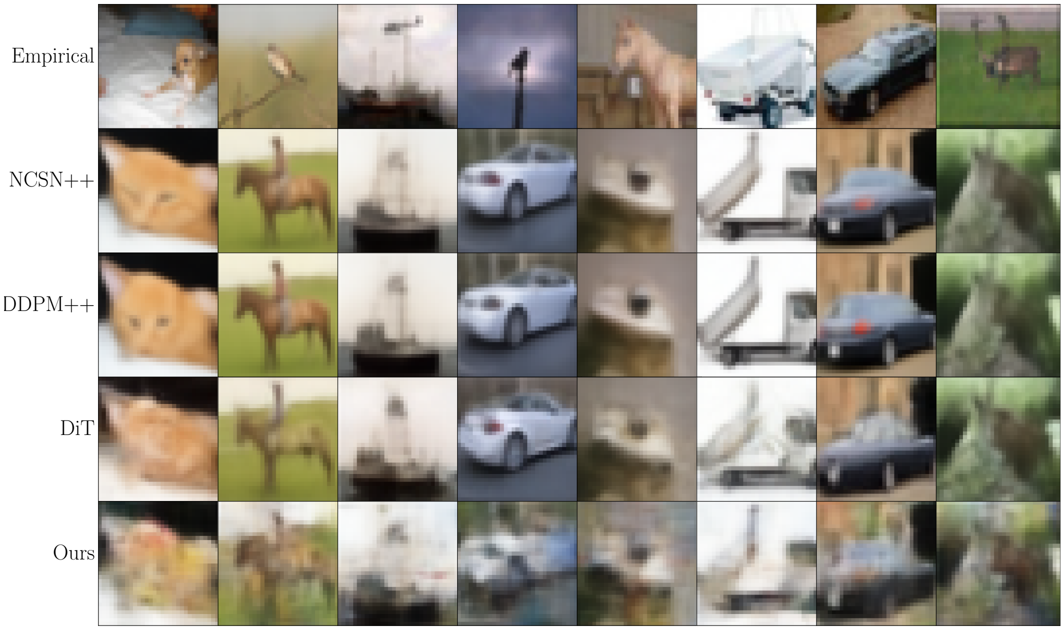

Figure 10: Additional denoiser outputs for $\sigma = 1.1$

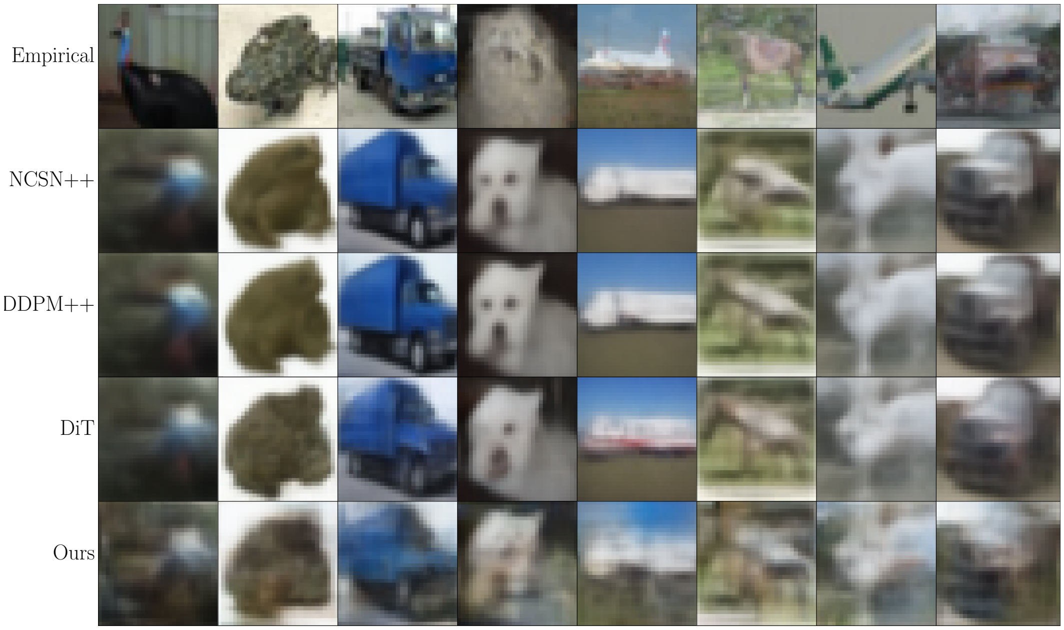

Figure 11: Additional denoiser outputs for $\sigma = 1.9$

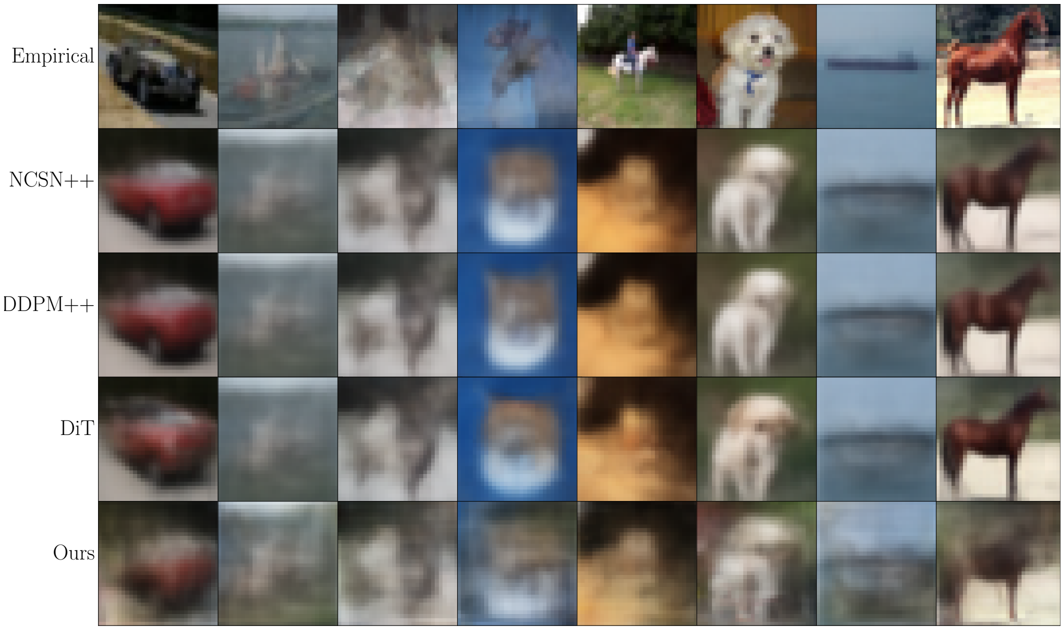

Figure 12: Additional denoiser outputs for $\sigma = 3.3$

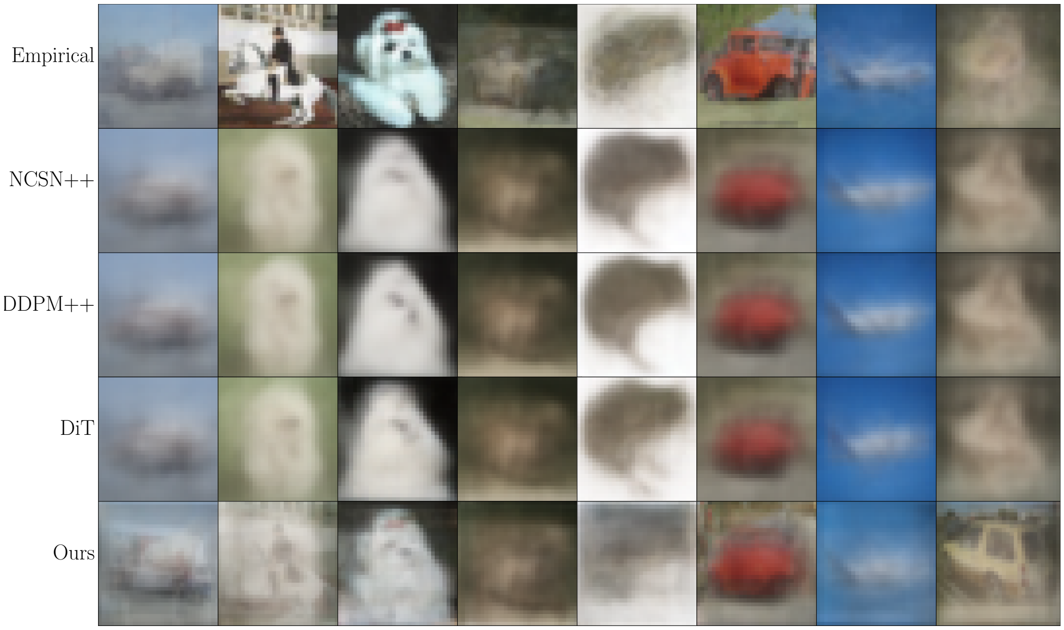

Figure 13: Additional denoiser outputs for $\sigma = 5.3$

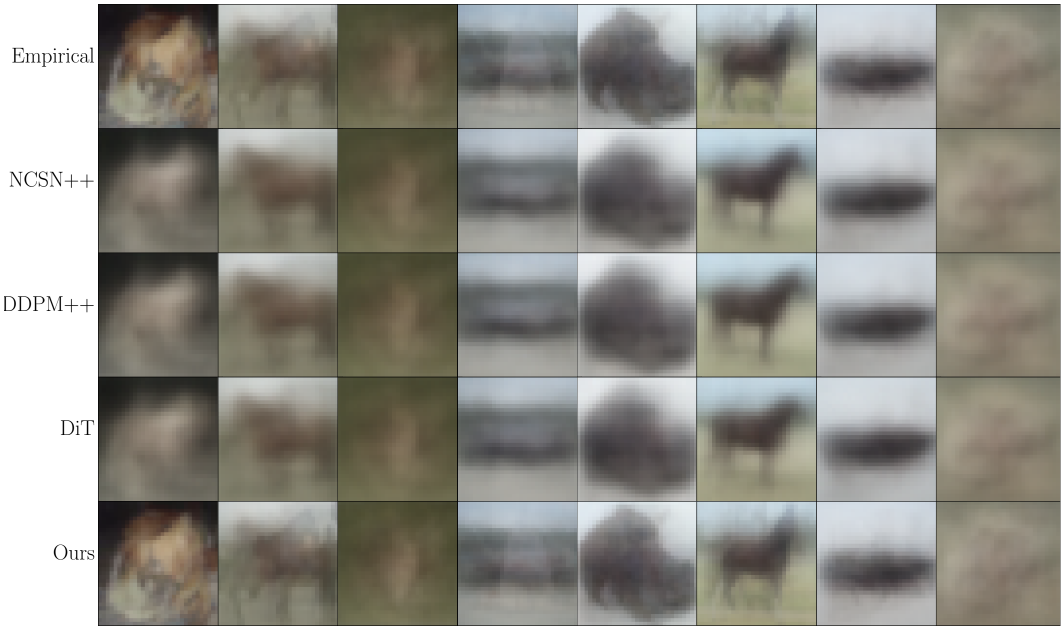

Figure 14: Additional denoiser outputs for $\sigma = 8.4$

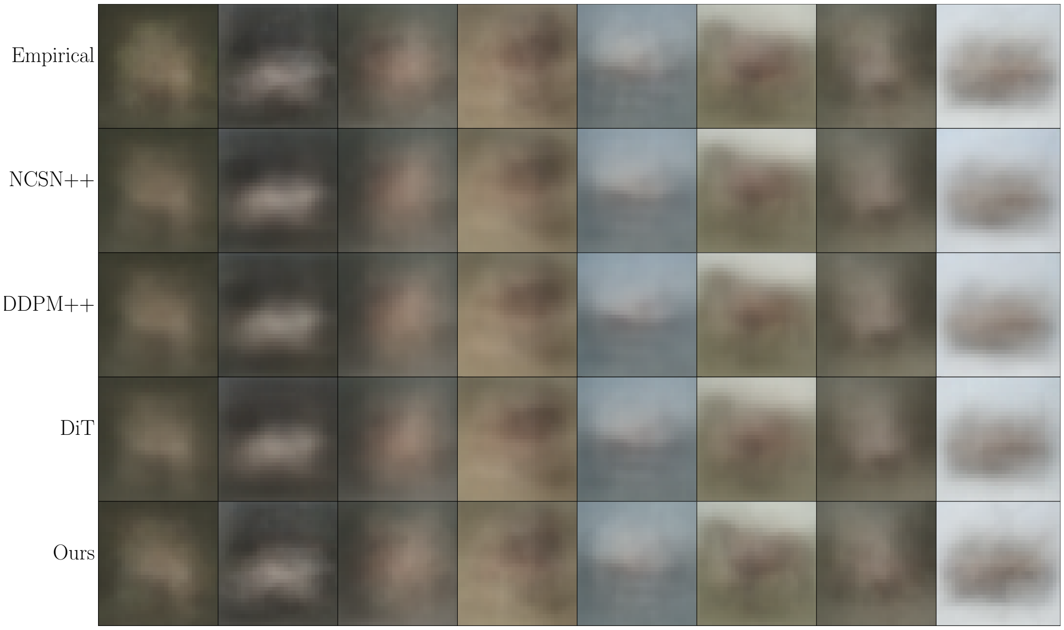

Figure 15: Additional denoiser outputs for $\sigma = 12.9$