A. PnF Sampling Details and Hyper-Parameters

We broadly adopt the geometric annealing schedule and hyper-parameters of annealed Langevin dynamics introduced in Song & Ermon (2019) and elaborated upon in Song & Ermon (2020). For both the PixelCNN++ and WaveNet models, we found that we needed additional intermediate noise levels to generate quality samples. We also found that good sample quality using these models required a smaller learning rate and mixing for more iterations than previous work (Song & Ermon, 2019; Jayaram & Thickstun, 2020). We speculate that the need for more levels of annealing and slower mixing could be a attributable to the autoregressive model parameterization, because also required a finer annealing and mixing schedule for the WaveNet models. Detailed hyper-parameters for the PixelCNN++ and WaveNet experiments are presented in Appendix B and C respectively.

Previous work makes the empirical observation that gradients of a noisy model $p_\sigma(\mathbf{x})$ are inverse-proportional to the variance of the noise: $\mathbb{E}\|\nabla_\mathbf{x} \log p_\sigma(\mathbf{x}^{(t)})\|^2 \propto 1/\sigma^2$ (Song & Ermon, 2019). This motivates the choice of learning rate $\eta \propto \sigma^2$. The empirical scale of the gradients is conjectured in Jayaram & Thickstun (2020) to be a consequence of "severe non-smoothness of the noiseless model $p(\mathbf{x})$." That work goes on to show that, if $p(\mathbf{x})$ were a Dirac spike, then exact inverse-proportionality of the gradients would hold in expectation. For the discretized autoregressive models discussed in this work, the noiseless distribution $p(\mathbf{x})$ is genuinely a mixture of Dirac spikes, and so the analysis in Jayaram & Thickstun (2020) applies without caveats to these models and justifies the choice $\eta \propto \sigma^2$ (the precise constant of proportionality remains application dependent, and discussed in the experimental details sections).

B. PixelCNN++ Experimental Details

Our visual sampling experiments are performed using a PixelCNN++ model $p(\mathbf{x})$ trained on CIFAR-10. Specifically, we used a public implementation of PixelCNN++ written by Lucas Caccia, available at:

`https://github.com/pclucas14/pixel-cnn-pp`.

We used the pre-trained weights for this model shared by Lucas Caccia (at the link above) with a reported test-set log loss of 2.95 bits per dimension. For the models $p_\sigma(\mathbf{x})$, we fine-tuned the pre-trained model for 10 epochs at each noise level σ^2. We adopt the geometric annealing schedule proposed in Song & Ermon (2019) for annealing σ^2, beginning at $\sigma_1 = 1.0$ and ending at $\sigma_L = 0.01$ using $L = 19$ noise levels. This is double the number of noise levels used in previous work (Song & Ermon, 2019; Jayaram & Thickstun, 2020). We also found that sample quality improved using a smaller learning rate and mixing for more iterations than reported in previous work. For conditional sampling tasks, we set $\delta = 3e-06$ and $T = 300$ in contrast to $\delta = 2e-05$ and $T = 100$ used in previous work. In wall-clock time, we find that conditional PixelCNN++ sampling tasks require approximately 60 minutes to generate a batch of 16 samples using a 1080Ti GPU.

C. WaveNet Experimental Details

Our audio sampling experiments are performed using a WaveNet model $p(\mathbf{x})$ trained on both the VCTK and Supra Piano datasets. We used the public implementation of Wavenet written by Ryuichi Yamamoto available at:

`https://github.com/r9y9/wavenet_vocoder`.

For all audio experiments, where data is encoded between $\{0..255\}$, we use $L = 15$ noise levels geometrically spaced between $\sigma = 175.9$ and $\sigma = 0.15$. The same noise levels are also used for the sampling speed and quality results presented in Figures 2 and 3. For all experiments, the number of Langevin steps per noise level is $T = 256$, except for Figure 2 where this parameter is varied to highlight changes in likelihood. The learning rate multiper $\delta = 0.05$ is used for all experiments. The Markov window w is based on the underlying architecture. When training the fine-tuned noise models, all training hyperparameters are kept the same as the original WaveNet implementation. For the WaveNet implementation used in this paper, this is $6,139$ samples which is roughly 0.3 seconds at a 22kHz sample rate Please refer to the WaveNet paper or the public WaveNet implementation for training details.

As discussed in the WaveNet paper (van den Oord et al., 2016a), 8-bit μ-law encoding results in a higher fidelity representation of audio than 8-bit linear encoding. For most experiments, the observation constraint $y = g(x)$ is still linear even under a μ-law encoding of x. However, for source separation, the constraint $y = x_1 + x_2$ is no longer linear under μ-law encoding. Consequently, we use an 8-bit linear encoding of x for source separation experiments to avoid a change of variables calculation. To facilitate a fair comparison, all ground truths and baselines shown in the demos use the corresponding μ-law or linear 8-bit encoding.

In the source separation experiments, all mixtures were created with 1/2 gain on each source component. Due to the natural variation of loudness in the training datasets, we find that our model generalizes to mixtures without exactly 1/2 gain on each source. The real life source separation result on the project website shows that we can separate a mixture in the wild when we have no information about the relative loudness of each component.

D. Additional PixelCNN++ Sampling Results

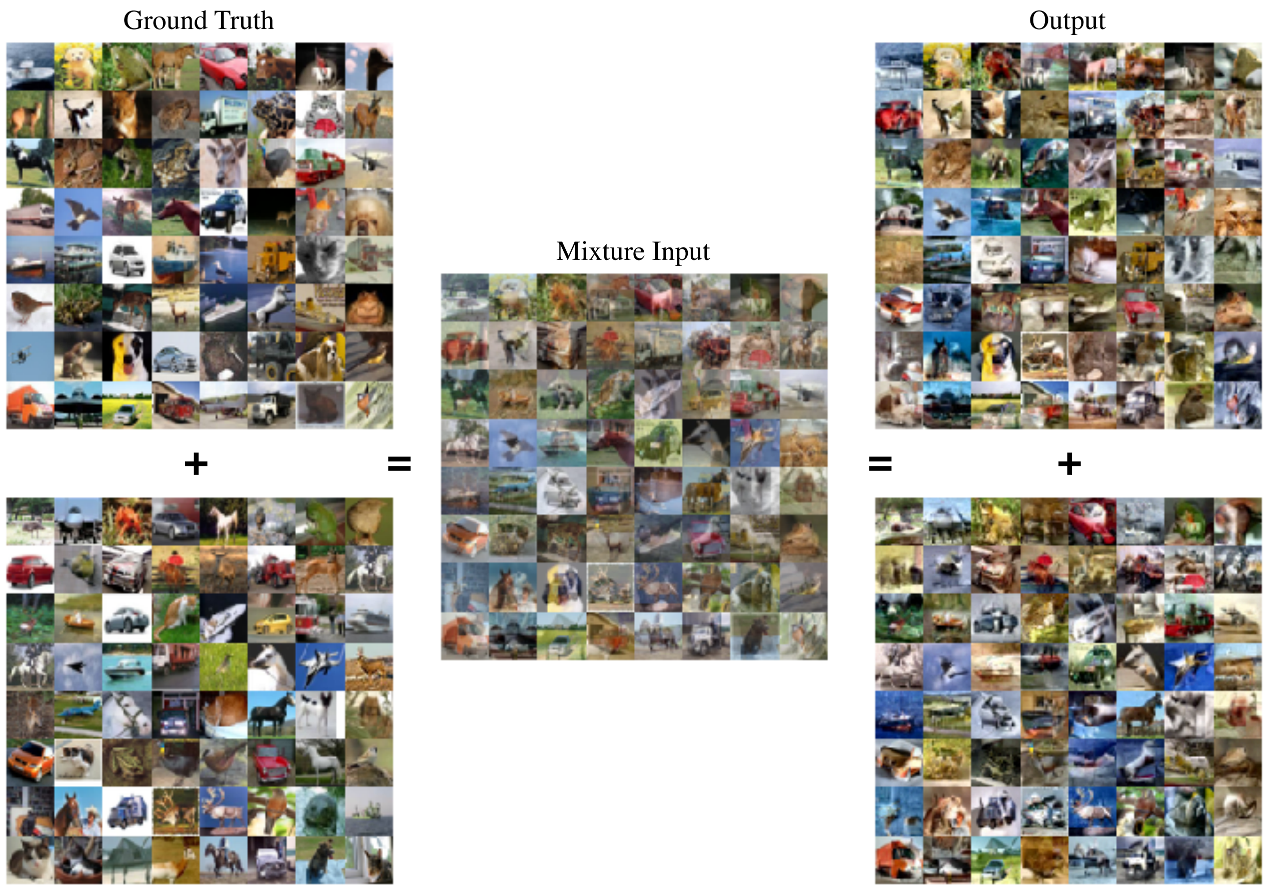

Figure 5. Additional uncurated results of PnF source separation (Section 4.4) for mixtures of CIFAR-10 test-set images using a PixelCNN++ prior trained on CIFAR-10.

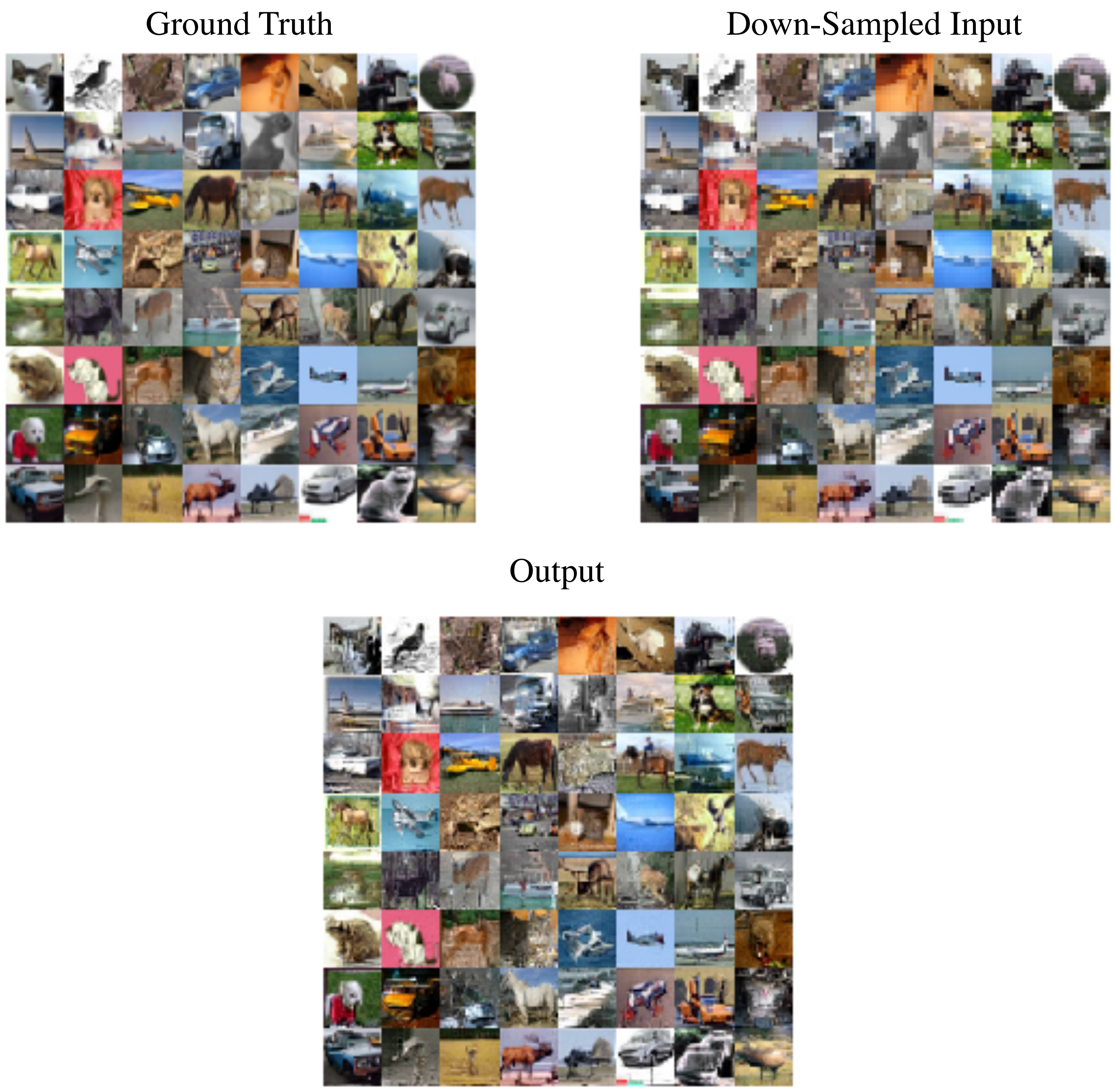

Figure 6. Additional uncurated results of PnF super-resolution (Section 4.5) applied to down-sampled CIFAR-10 test-set images using a PixelCNN++ prior trained on CIFAR-10.

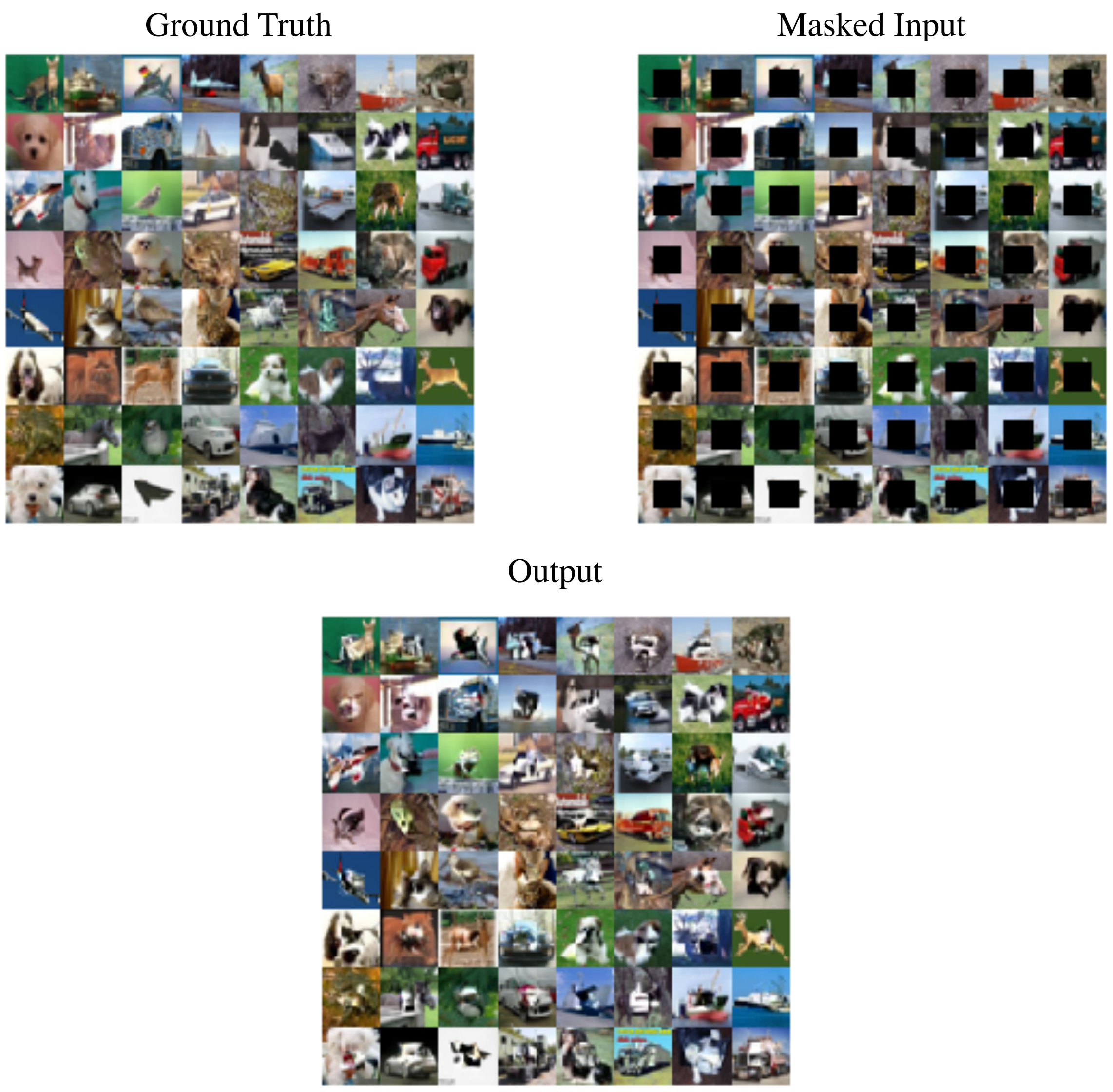

Figure 7. Additional uncurated results of PnF inpainting (Section 4.6) applied to masked CIFAR-10 test-set images using a PixelCNN++ prior trained on CIFAR-10.