Improved Denoising Diffusion Probabilistic Models (Supplementary)

Anonymous Authors[1]

1. Hyperparameters

For all of our experiments, we use a UNet model architecture[1] similar to that used by Ho et al. (2020). We changed the attention layers to use multi-head attention (Vaswani et al., 2017), and opted to use four attention heads rather than one (while keeping the same total number of channels). We employ attention not only at the 16x16 resolution, but also at the 8x8 resolution. Additionally, we changed the way the model conditions on t. In particular, instead of computing a conditioning vector v and injecting it into hidden state h as GroupNorm$(h + v)$, we compute conditioning vectors w and b and inject them into the hidden state as GroupNorm$(h)(w + 1) + b$. We found in preliminary experiments on ImageNet 64×64 that these modifications slightly improved FID.

For ImageNet 64×64 the architecture we use is described as follows. The downsampling stack performs four steps of downsampling, each with three residual blocks (He et al., 2015). The upsampling stack is setup as a mirror image of the downsampling stack. From highest to lowest resolution, the UNet stages use $[C, 2C, 3C, 4C]$ channels, respectively. In our ImageNet 64×64 ablations, we set $C = 128$, but we experiment with scaling C in a later section. We estimate that, with $C = 128$, our model is comprised of 120M parameters and requires roughly 39 billion FLOPs in the forward pass.

For our CIFAR-10 experiments, we use a smaller model with three resblocks per downsampling stage and layer widths $[C, 2C, 2C, 2C]$ with $C = 128$. We swept over dropout values $\{0.1, 0.2, 0.3\}$ and found that 0.1 worked best for the linear schedule while 0.3 worked best for our cosine schedule. We expand upon this in Section 6.

We use Adam (Kingma & Ba, 2014) for all of our experiments. For most experiments, we use a batch size of 128, a learning rate of 10^{-4}, and an exponential moving average (EMA) over model parameters with a rate of 0.9999. For our scaling experiments, we vary the learning rate to accomodate for different model sizes. For our larger class-conditional ImageNet 64×64 experiments, we scaled up the batch size to 2048 for faster training on more GPUs.

λ	Stop-gradient	NLL	FID
0.001	yes	3.62	28.04
0.01	yes	3.62	27.36
0.001	no	3.62	30.89
0.01	no	3.63	34.35

Table 1. Ablating hyper-parameters of the L_{hybrid} objective on ImageNet 64×64. All models were trained for 200K iterations.

When using the linear noise schedule from Ho et al. (2020), we linearly interpolate from $\beta_1 = 0.0001/4$ to $\beta_{4000} = 0.02/4$ to preserve the shape of $\bar{\alpha}_t$ for the $T = 4000$ schedule.

When computing FID we produce 50K samples from our models, except for unconditional ImageNet 64×64 where we produce 10K samples. Using only 10K samples biases the FID to be higher, but requires much less compute for sampling and helps do large ablations. Since we mainly use FID for relative comparisons on unconditional ImageNet 64×64, this bias is acceptable. For computing the reference distribution statistics we follow prior work (Ho et al., 2020; Brock et al., 2018) and use the full training set for CIFAR-10 and ImageNet, and 50K training samples for LSUN. Note that unconditional ImageNet 64×64 models are trained and evaluated using the official ImageNet-64 dataset (van den Oord et al., 2016), whereas for class conditional ImageNet 64×64 and 256×256 we center crop and area downsample images (Brock et al., 2018).

In Table 1 we ablate the two major choices in our L_{hybrid} objective: the L_{vlb} weight λ, and the stop-gradient after μ_θ when computing L_{vlb}. We find that the stop-gradient improves sample quality and reduces sensitivity to λ.

2. Fast Sampling on LSUN 256×256

To test the effectiveness of our L_{hybrid} models on a high-resolution domain, we trained both L_{hybrid} and L_{simple} models on the LSUN bedroom (Yu et al., 2015) dataset. We

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1] In initial experiments, we found that a ResNet-style architecture with no downsampling achieved better log-likelihoods but worse FIDs than the UNet architecture.

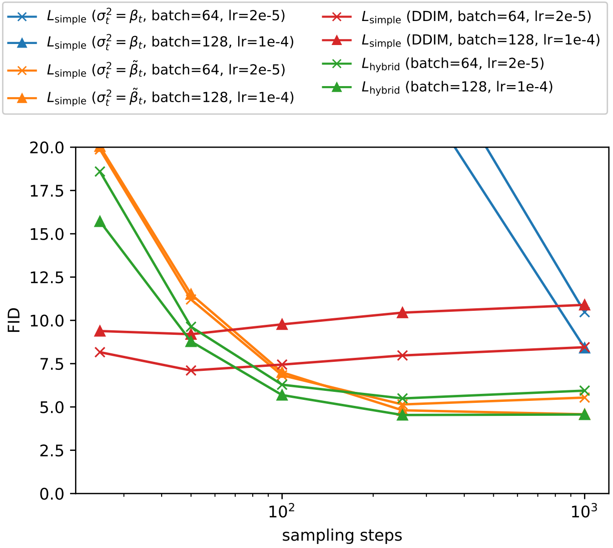

Figure 1. FID vs. number of sampling steps from an LSUN 256 × 256 bedroom model.

MODEL	FID
VQ-VAE-2 ((Razavi et al., 2019), two-stage)	38.1
Improved Diffusion (ours, single-stage)	31.5
Improved Diffusion (ours, two-stage)	**12.3**
BigGAN (Brock et al., 2018)	7.7
BigGAN-deep (Brock et al., 2018)	**7.0**

Table 2. Sample quality comparison on class conditional ImageNet 256 × 256. BigGAN FIDs are reported for the truncation that results in the best FID.

train two models: one with batch size 64 and learning rate 2×10^{-5} as in Ho et al. (2020), and another with a larger batch size 128 and learning rate 10^{-4}. All models were trained with 153.6M examples, which is 2.4M training iterations with batch size 64.

Our results are displayed in Figure 1. We find that DDIM outperforms our L_{hybrid} model when using fewer than 50 diffusion steps, while our L_{hybrid} model outperforms DDIM with more than 50 diffusion steps. Interestingly, we note that DDIM benefits from a smaller learning rate and batch size, whereas our method is able to take advantage of a larger learning rate and batch size.

3. Sample Quality on ImageNet 256 × 256

We trained two models on class conditional ImageNet 256 × 256. The first is a usual diffusion model that directly models the 256 × 256 images. The second model reduces compute by chaining a pretrained 64 × 64 model $p(x_{64}|y)$ with another upsampling diffusion model $p(x_{256}|x_{64}, y)$ to upsample images to 256 × 256. For the upsampling model, the downsampled image x_{64} is passed as extra conditioning input to the UNet. This is similar to VQ-VAE-2 (Razavi et al., 2019), which uses two stages of priors at different latent resolutions to more efficiently learn global and local features. The linear schedule worked better for 256 × 256 images, so we used that for these results. Table 2 summarizes our results. For VQ-VAE-2, we use the FIDs reported in (Ravuri & Vinyals, 2019). Diffusion models still obtain the best FIDs for a likelihood-based model, and close the gap to GANs considerably.

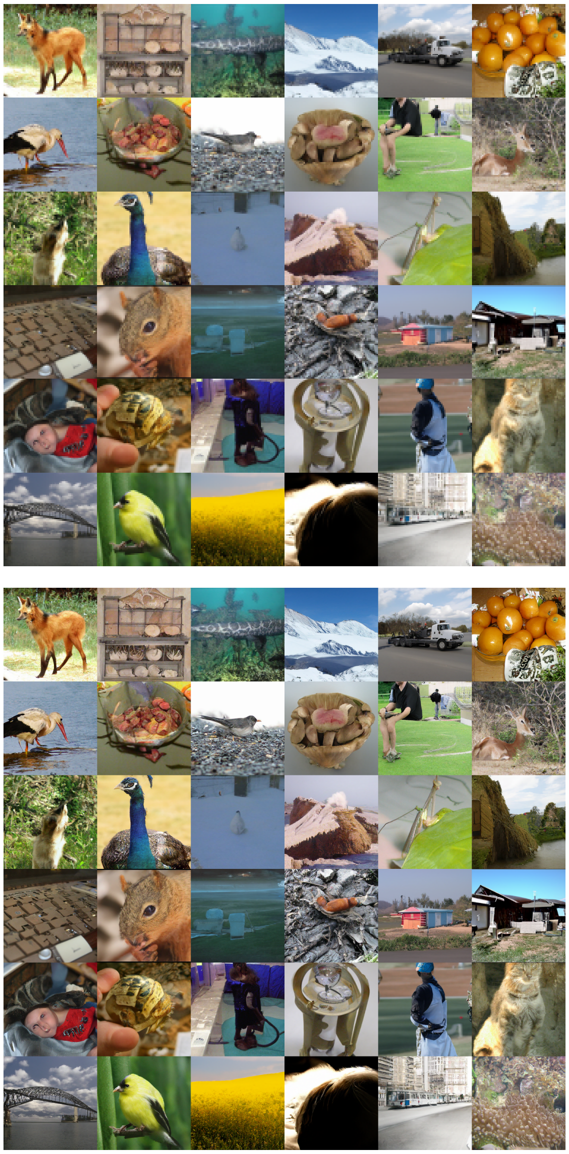

Figure 2. Random samples from two-stage class conditional ImageNet 256 × 256 model. On top are random samples from the 64 × 64 model (FID 2.92), whereas on bottom are the results after upsampling them to 256 × 256 (FID 12.3). Each model uses 250 sampling steps.

4. Combining L_{hybrid} and L_{vlb} Models

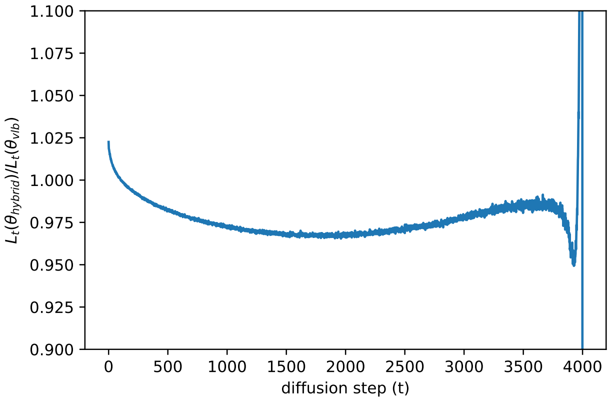

Figure 3. The ratio between VLB terms for each diffusion step of θ_{hybrid} and θ_{vlb}. Values less than 1.0 indicate that θ_{hybrid} is ”better” than θ_{vlb} for that timestep of the diffusion process.

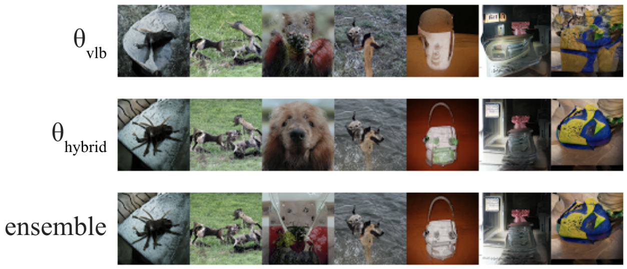

Figure 4. Samples from θ_{vlb} and θ_{hybrid}, as well as an ensemble produced by using θ_{vlb} for the first and last 100 diffusion steps. For these samples, the seed was fixed, allowing a direct comparison between models.

To understand the trade-off between L_{hybrid} and L_{vlb}, we show in Figure 3 that the model resulting from L_{vlb} (referred to as θ_{vlb}) is better at the start and end of the diffusion process, while the model resulting from L_{hybrid} (referred to as θ_{hybrid}) is better throughout the middle of the diffusion process. This suggests that θ_{vlb} is focusing more on imperceptible details, hence the lower sample quality.

Given the above observation, we performed an experiment on ImageNet 64×64 to combine the two models by constructing an ensemble that uses θ_{hybrid} for $t \in [100, T-100)$ and θ_{vlb} elsewhere. We found that this model achieved an FID of **19.9** and an NLL of **3.52 bits/dim**. This is only slightly worse than θ_{hybrid} in terms of FID, while being better than both models in terms of NLL.

5. Log-likelihood with Fewer Diffusion Steps

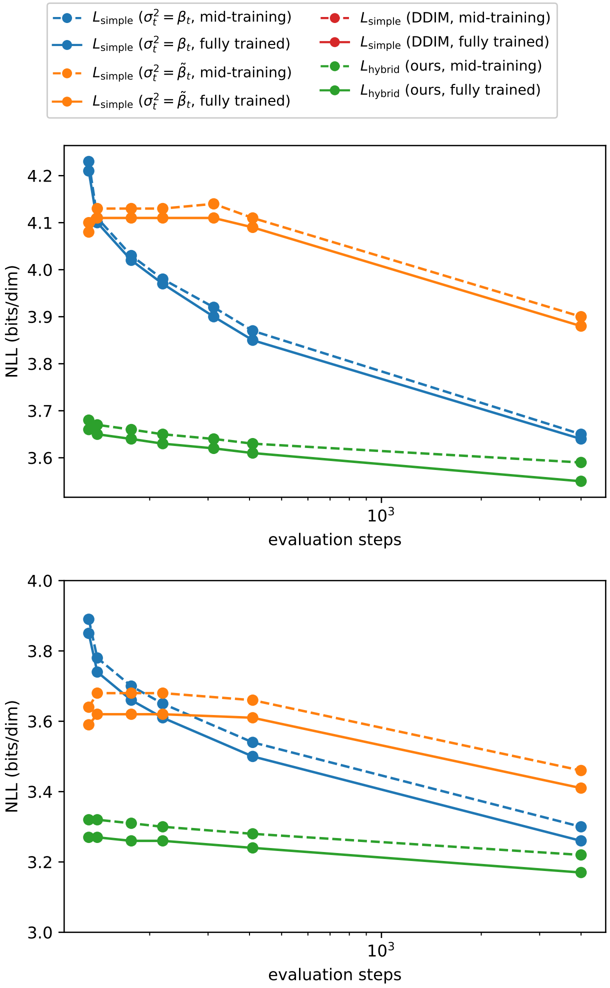

Figure 5. NLL versus number of evaluation steps, for models trained on ImageNet 64×64 (top) and CIFAR-10 (bottom). All models were trained with 4000 diffusion steps.

Figures 5 plots negative log-likelihood as a function of number of sampling steps for both ImageNet 64×64 and CIFAR-10. In initial experiments, we found that although constant striding did not significantly affect FID, it drastically reduced log-likelihood. To address this, we use a strided subset of timesteps as for FID, but we also include every t from 1 to T/K. This requires T/K extra evaluation steps, but greatly improves log-likelihood compared to the uniformly strided schedule. We did not attempt to calculate NLL using DDIM, since Song et al. (2020) does not present NLL results or a simple way of estimating likelihood under DDIM.

6. Overfitting on CIFAR-10

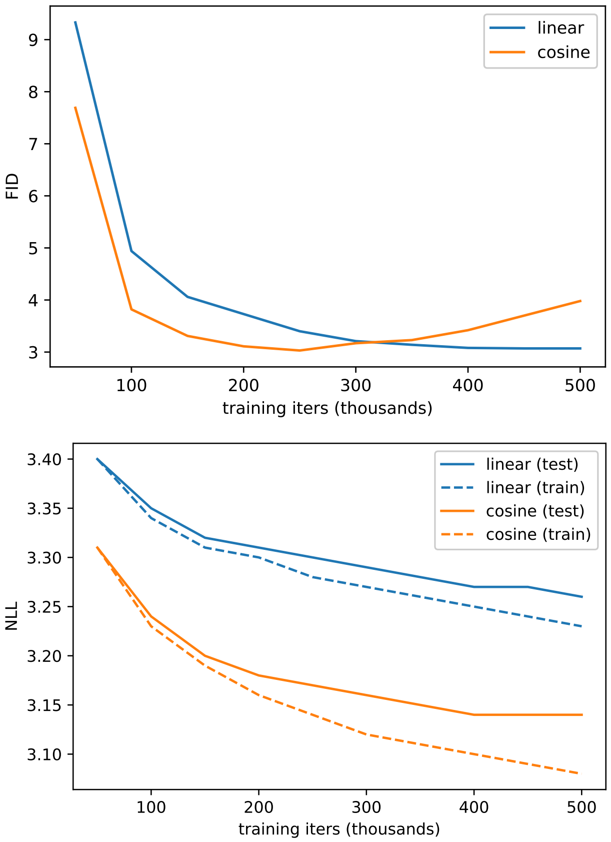

Figure 6. FID (top) and NLL (bottom) over the course of training for two CIFAR-10 models, both with dropout 0.1. The model trained with the linear schedule learns more slowly, but does not overfit as quickly. When too much overfitting occurs, we observed overfitting artifacts similar to those from Salimans et al. (2017), which is reflected by increasing FID.

All of our CIFAR-10 models experienced overfitting. This tended to hurt FID rather than producing exact training images, similar to observations from Salimans et al. (2017) and Ho et al. (2020). Before overfitting, all of our models tended to reach similar optimal FID at some point during training. Holding dropout constant, we found that models trained with our cosine schedule tended to reach optimal performance (and then overfit) more quickly than those trained with the linear schedule (Figure 6). In our experiments, we corrected for this difference by using more dropout for our cosine models than the linear models. We suspect that the overfitting from the cosine schedule is either due to 1) less noise in the cosine schedule providing less regularization, or 2) the cosine schedule making optimization, and thus overfitting, easier.

7. Early stopping for FID

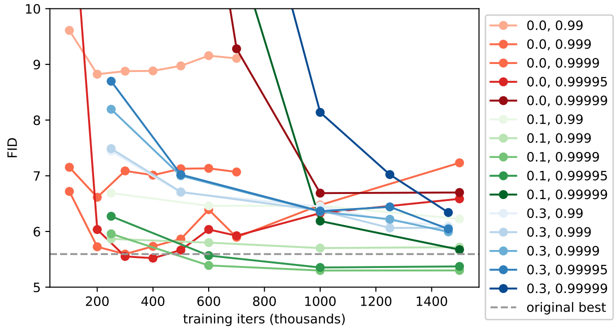

Figure 7. A sweep of dropout and EMA hyperparameters on class conditional ImageNet-64.

Like on CIFAR-10, we surprisingly observed overfitting on class-conditional ImageNet 64×64, despite it being a much larger and more diverse dataset. The main observable result of this overfitting was that FID started becoming worse over the course of training. We initially tried a sweep (Figure 7) over the EMA hyperparameter to make sure it was well tuned, and found that 0.9999 and 0.99995 worked best. We then tried runs with dropout 0.1 and 0.3, and found that models with a small amount of dropout improved the best attainable FID but took longer to get to the same performance and still eventually overfit. We concluded that the best way to train, given what we know, is to early stop and instead increase model size if we want to use additional training compute.

8. Samples with Varying Steps and Objectives

Figures 8 through 13 show unconditional ImageNet 64×64 samples as we reduce number of sampling steps for an L_{hybrid} model with $4K$ diffusion steps trained for 1.5M training iterations.

Figures 14 through 19 show unconditional CIFAR-10 samples as we reduce number of sampling steps for an L_{hybrid} model with $4K$ diffusion steps trained for 500K training iterations.

Figures 20 and 21 highlight the difference in sample quality between models trained with L_{hybrid} and L_{vlb}.

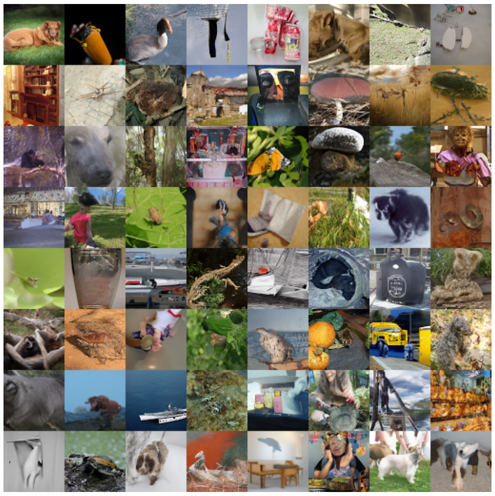

Figure 8. 50 sampling steps on unconditional ImageNet 64 × 64

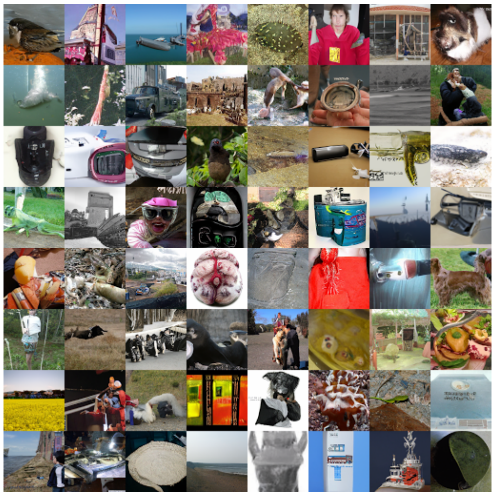

Figure 11. 400 sampling steps on unconditional ImageNet 64 × 64

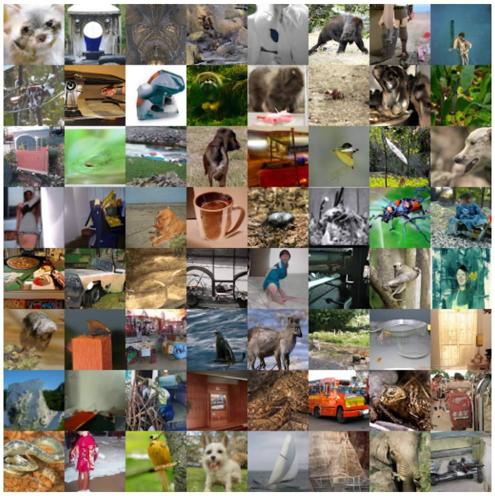

Figure 9. 100 sampling steps on unconditional ImageNet 64 × 64

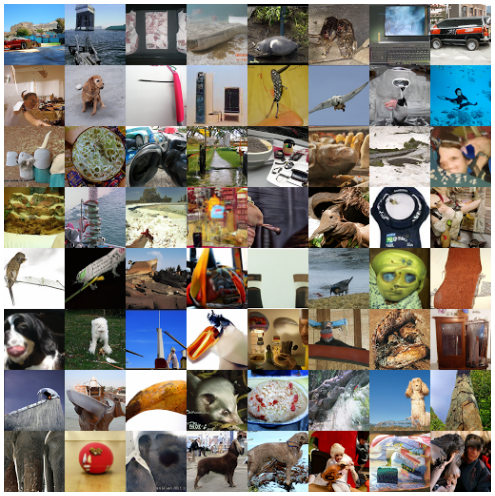

Figure 12. 1000 sampling steps on unconditional ImageNet 64×64

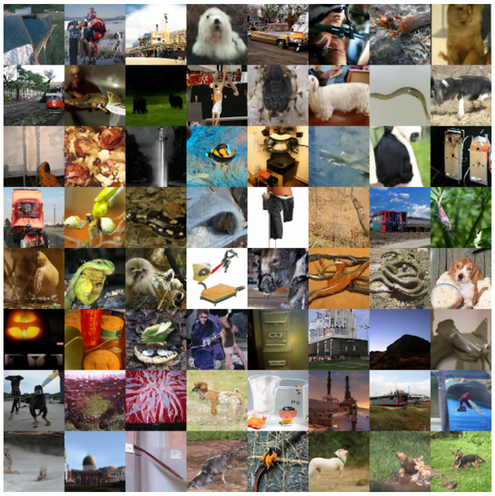

Figure 10. 200 sampling steps on unconditional ImageNet 64 × 64

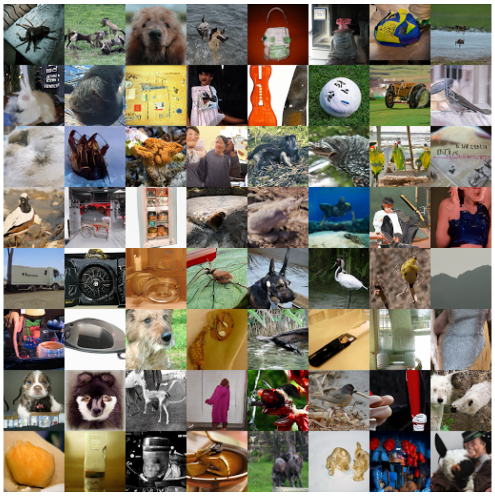

Figure 13. 4K sampling steps on unconditional ImageNet 64 × 64.

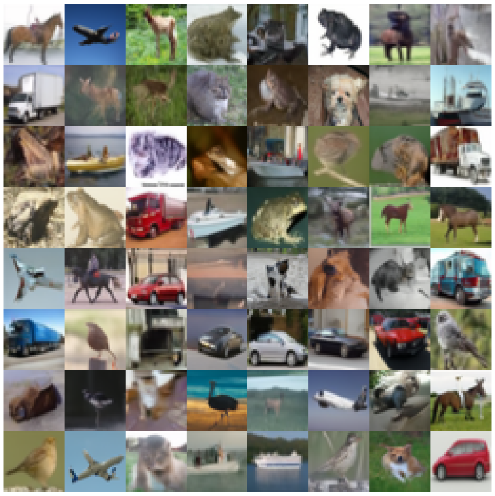

Figure 14. 50 sampling steps on unconditional CIFAR-10

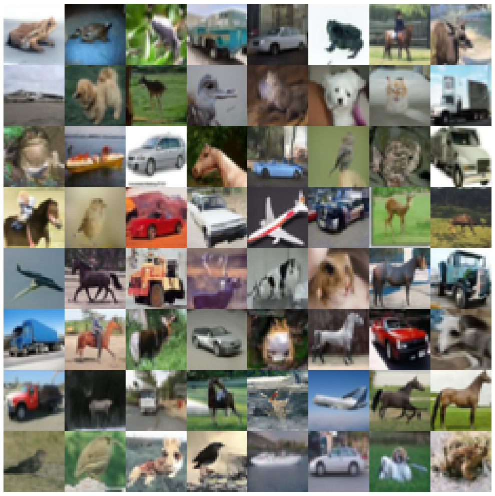

Figure 15. 100 sampling steps on unconditional CIFAR-10

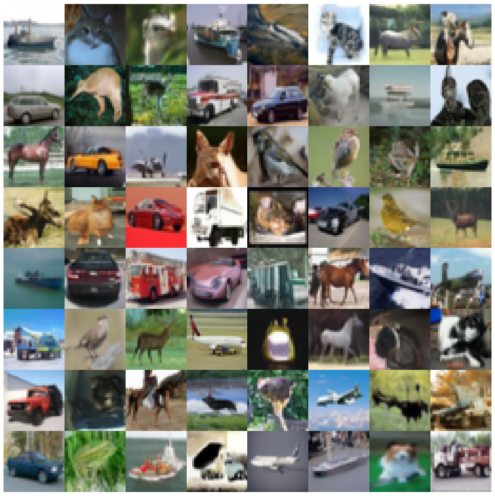

Figure 16. 200 sampling steps on unconditional CIFAR-10

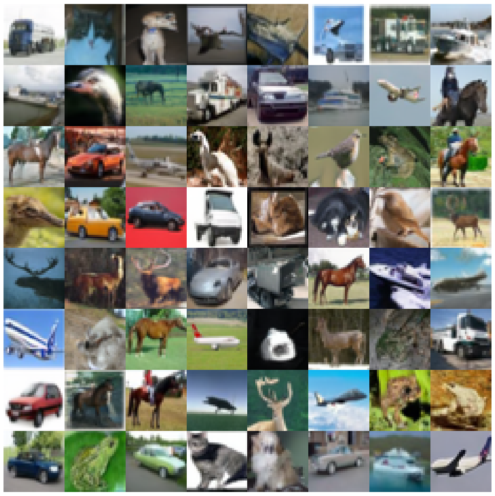

Figure 17. 400 sampling steps on unconditional CIFAR-10

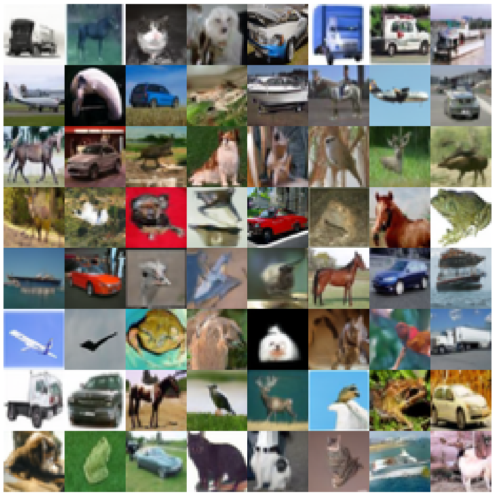

Figure 18. 1000 sampling steps on unconditional CIFAR-10

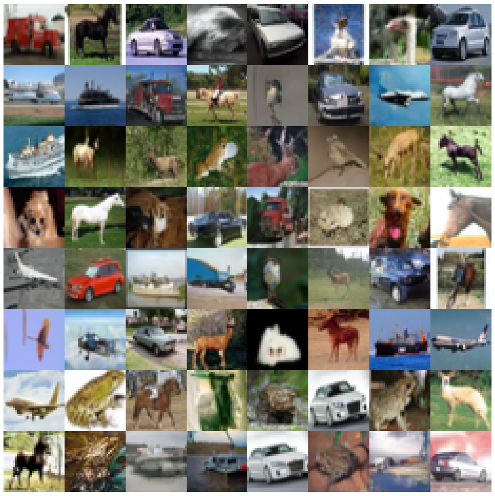

Figure 19. 4000 sampling steps on unconditional CIFAR-10

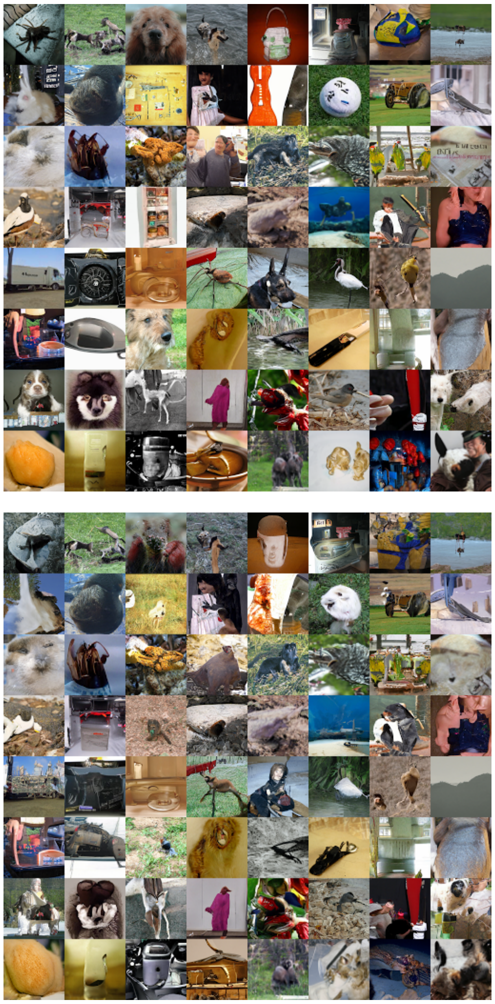

Figure 20. Unconditional ImageNet 64 × 64 samples generated from L_{hybrid} (top) and L_{vlb} (bottom) models using the exact same random noise. Both models were trained for 1.5M iterations.

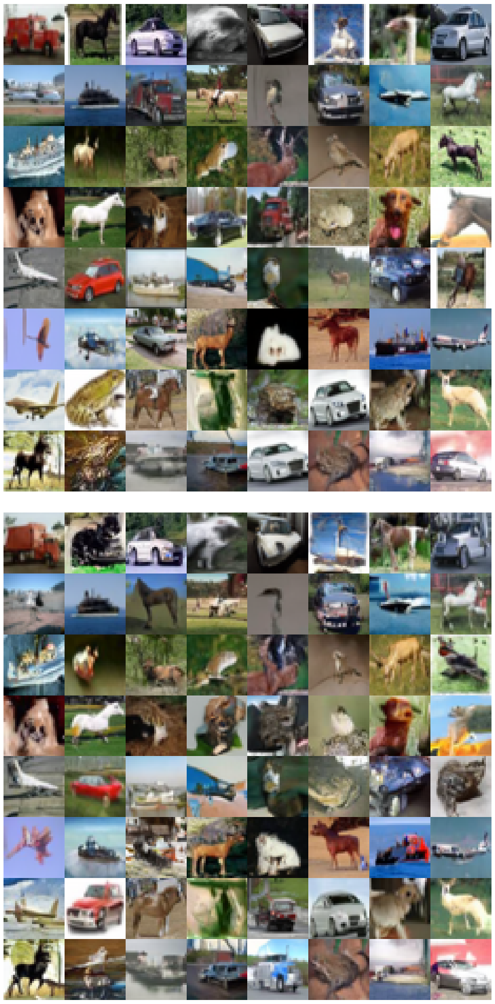

Figure 21. Unconditional CIFAR-10 samples generated from L_{hybrid} (top) and L_{vlb} (bottom) models using the exact same random noise. Both models were trained for 500K iterations.

References

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014.

Ravuri, S. and Vinyals, O. Classification accuracy score for conditional generative models. *arXiv preprint arXiv:1905.10887*, 2019.

Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2, 2019.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications, 2017.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models, 2020.

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks, 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2015.