ADVERSARIAL SCORE MATCHING AND IMPROVED SAMPLING FOR IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Denoising Score Matching with Annealed Langevin Sampling (DSM-ALS) has recently found success in generative modeling. The approach works by first training a neural network to estimate the score of a distribution, and then using Langevin dynamics to sample from the data distribution assumed by the score network. Despite the convincing visual quality of samples, this method appears to perform worse than Generative Adversarial Networks (GANs) under the Fréchet Inception Distance, a standard metric for generative models. We show that this apparent gap vanishes when *denoising* the final Langevin samples using the score network. In addition, we propose two improvements to DSM-ALS: 1) Consistent Annealed Sampling as a more stable alternative to Annealed Langevin Sampling, and 2) a hybrid training formulation, composed of both Denoising Score Matching and adversarial objectives. By combining these two techniques and exploring different network architectures, we elevate score matching methods and obtain results competitive with state-of-the-art image generation on CIFAR-10.

1 INTRODUCTION

Song and Ermon (2019) recently proposed a novel method of generating samples from a target distribution through a combination of Denoising Score Matching (DSM) (Hyvärinen, 2005; Vincent, 2011; Raphan and Simoncelli, 2011) and Annealed Langevin Sampling (ALS) (Welling and Teh, 2011; Roberts et al., 1996). Since convergence to the distribution is guaranteed by the ALS, their approach (DSM-ALS) produces high-quality samples and guarantees high diversity. Though, this comes at the cost of requiring an iterative process during sampling, contrary to other generative methods.

Song and Ermon (2020) further improved their approach by increasing the stability of score matching training and proposing theoretically sound choices of hyperparameters. They also scaled their approach to higher-resolution images and showed that DSM-ALS is competitive with other generative models. Song and Ermon (2020) observed that the images produced by their improved model were more visually appealing than the ones from their original work; however, the reported Fréchet Inception Distance (FID) (Heusel et al., 2017) did not correlate with this improvement.

Although DSM-ALS is gaining traction, Generative adversarial networks (GANs) (Goodfellow et al., 2014) remain the leading approach to generative modeling. GANs are a very popular class of generative models; they have been successfully applied to image generation (Brock et al., 2018; Karras et al., 2017; 2019; 2020) and have subsequently spawned a wealth of variants (Radford et al., 2015a; Miyato et al., 2018; Jolicoeur-Martineau, 2018; Zhang et al., 2019). The idea behind this method is to train a Discriminator (D) to correctly distinguish real samples from fake samples generated by a second agent, known as the Generator (G). GANs excel at generating high-quality samples as the discriminator captures features that make an image plausible, while the generator learns to emulate them.

Still, GANs often have trouble producing data from all possible modes, which limits the diversity of the generated samples. A wide variety of tricks have been developed to address this issue in GANs (Kodali et al., 2017; Gulrajani et al., 2017; Arjovsky et al., 2017; Miyato et al., 2018; Jolicoeur-Martineau and Mitliagkas, 2019), though it remains an issue to this day. DSM-ALS, on the other hand, does not suffer from that problem since ALS allows for sampling from the full distribution captured by the score network. Nevertheless, the perceptual quality of DSM-ALS higher-resolution

images has so far been inferior to that of GAN-generated images. Generative modeling has since seen some incredible work from Ho et al. (2020), who achieved exceptionally low (better) FID on image generation tasks. Their approach showcased a diffusion-based method (Sohl-Dickstein et al., 2015; Goyal et al., 2017) that shares close ties with DSM-ALS, and additionally proposed a convincing network architecture derived from Salimans et al. (2017).

In this paper, after introducing the necessary technical background in the next section, we build upon the work of Song and Ermon (2020) and propose improvements based on theoretical analyses both at training and sampling time. Our contributions are as follows:

- We propose Consistent Annealed Sampling (CAS) as a more stable alternative to ALS, correcting inconsistencies relating to the scaling of the added noise;
- We show how to recover the *expected denoised sample* (EDS) and demonstrate its unequivocal benefits *w.r.t* the FID. Notably, we show how to resolve the mismatch observed in DSM-ALS between the visual quality of generated images and its high (worse) FID;
- We propose to further exploit the EDS through a hybrid objective function, combining GAN and Denoising Score Matching objectives, thereby encouraging the EDS of the score network to be as realistic as possible.

In addition, we show that the network architecture used used by Ho et al. (2020) significantly improves sample quality over the RefineNet (Lin et al., 2017a) architecture used by Song and Ermon (2020). In an ablation study performed on CIFAR-10 and LSUN-church, we demonstrate how these contributions bring DSM-ALS in range of the state-of-the-art for image generation tasks *w.r.t.* the FID. The code to replicate our experiments is publicly available at **[Available in supplementary material]**.

2 Background

2.1 Denoising Score Matching

Denoising Score Matching (DSM) (Hyvärinen, 2005) consists of training a score network to approximate the gradient of the log density of a certain distribution ($\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$), referred to as the score function. This is achieved by training the network to approximate a noisy surrogate of p at multiple levels of Gaussian noise corruption (Vincent, 2011). The score network s, parametrized by θ and conditioned on the noise level σ, is tasked to minimize the following loss:

$$\frac{1}{2}\,\mathbb{E}_{p(\tilde{\boldsymbol{x}},\boldsymbol{x},\sigma)}\left[\left\|\sigma s_\theta(\tilde{\boldsymbol{x}},\sigma)+\frac{\tilde{\boldsymbol{x}}-\boldsymbol{x}}{\sigma}\right\|_2^2\right], \tag{1}$$

where $p(\tilde{\boldsymbol{x}},\boldsymbol{x},\sigma)=q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})p(\boldsymbol{x})p(\sigma)$. We define further $q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})=\mathcal{N}(\tilde{\boldsymbol{x}}|\boldsymbol{x},\sigma^2\boldsymbol{I})$ the corrupted data distribution, $p(\boldsymbol{x})$ the training data distribution, and $p(\sigma)$ the uniform distribution over a set $\{\sigma_i\}$ corresponding to different levels of noise. In practice, this set is defined as a geometric progression between σ_1 and σ_L (with L chosen according to some computational budget):

$$\{\sigma_i\}_{i=1}^L=\left\{\gamma^i\sigma_1 \;\middle|\; i\in\{0,\dots,L-1\},\gamma\triangleq\frac{\sigma_2}{\sigma_1}=\dots=\left(\frac{\sigma_L}{\sigma_1}\right)^{\frac{1}{L-1}}<1\right\}. \tag{2}$$

Rather than having to learn a different score function for every σ_i, one can train an unconditional score network by defining $s_\theta(\tilde{\boldsymbol{x}},\sigma_i)=s_\theta(\tilde{\boldsymbol{x}})/\sigma_i$, and then minimizing Eq. 1. While unconditional networks are less heavy computationally, it remains an open question whether conditioning helps performance. Li et al. (2019) and Song and Ermon (2020) found that the unconditional network produced better samples, while Ho et al. (2020) obtained better results than both of them using a conditional network. Additionally, the denoising autoencoder described in Lim et al. (2020) gives evidence supporting the benefits of conditioning when the noise becomes small (also see App. D and E for a theoretical discussion of the difference). While our experiments are conducted with unconditional networks, we believe our techniques can be straightforwardly applied to conditional networks; we leave that extension for future work.

2.2 ANNEALED LANGEVIN SAMPLING

Given a score function, one can use Langevin dynamics (or Langevin sampling) (Welling and Teh, 2011) to sample from the corresponding probability distribution. In practice, the score function is generally unknown and estimated through a score network trained to minimize Eq. 1. Song and Ermon (2019) showed that Langevin sampling has trouble exploring the full support of the distribution when the modes are too far apart and proposed Annealed Langevin Sampling (ALS) as a solution. ALS starts sampling with a large noise level and progressively anneals it down to a value close to 0, ensuring both proper mode coverage and convergence to the data distribution. Its precise description is shown in Algorithm 1.

Algorithm 1 Annealed Langevin Sampling

Require: $s_\theta, \{\sigma_i\}_{i=1}^L, \epsilon, n_\sigma$.

Initialize $\boldsymbol{x}$
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon\, \sigma_i^2/\sigma_L^2$
  for $t \leftarrow 1$ to n_σ **do**
    Draw $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + \alpha_i s_\theta(\boldsymbol{x}, \sigma_i) + \sqrt{2\alpha_i}\boldsymbol{z}$
return $\boldsymbol{x}$

Algorithm 2 Consistent Annealed Sampling

Require: $s_\theta, \{\sigma_i\}_{i=1}^L, \gamma, \epsilon, \sigma_{L+1} = 0$

Initialize $\boldsymbol{x}$
$\beta \leftarrow \sqrt{1 - (1 - \epsilon/\sigma_L^2)^2 / \gamma^2}$
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon\, \sigma_i^2/\sigma_L^2$
  Draw $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + \alpha_i s_\theta(\boldsymbol{x}, \sigma_i) + \beta\sigma_{i+1}\boldsymbol{z}$
return $\boldsymbol{x}$

2.3 EXPECTED DENOISED SAMPLE (EDS)

A little known fact from Bayesian literature is that one can recover a denoised sample from the score function using the Empirical Bayes mean (Robbins, 1955; Miyasawa, 1961; Raphan and Simoncelli, 2011):

$$s^*(\tilde{\boldsymbol{x}}, \sigma) = \frac{H^*(\tilde{\boldsymbol{x}}, \sigma) - \tilde{\boldsymbol{x}}}{\sigma^2}, \tag{3}$$

where $H^*(\tilde{\boldsymbol{x}}, \sigma) \triangleq \mathbb{E}_{\boldsymbol{x}\sim q_\sigma(\boldsymbol{x}|\tilde{\boldsymbol{x}})}[\boldsymbol{x}]$ is the expected denoised sample given a noisy sample (or Empirical Bayes mean), conditioned on the noise level. A different way of reaching the same result is through the closed-form of the optimal score function, as presented in Appendix D. The corresponding result for unconditional score function is presented in Appendix E for completeness.

The EDS corresponds to the expected real image given a corrupted image; it can be thought of as what the score network believes to be the true image concealed within the noisy input. It has also been suggested that denoising the samples (i.e., taking the EDS) at the end of the Langevin sampling improves their quality (Saremi and Hyvarinen, 2019; Li et al., 2019; Kadkhodaie and Simoncelli, 2020). In Section 4, we provide further evidence that denoising the final Langevin sample brings it closer to the assumed data manifold. In particular, we show that the Fréchet Inception Distance (FID) consistently decreases (improves) after denoising. Finally, in Section 5, we build a hybrid training objective using the properties of the EDS discussed above.

3 CONSISTENT SCALING OF THE NOISE

In this section, we present inconsistencies in ALS relating to the noise scaling and introduce Consistent Annealed Sampling (CAS) as an alternative.

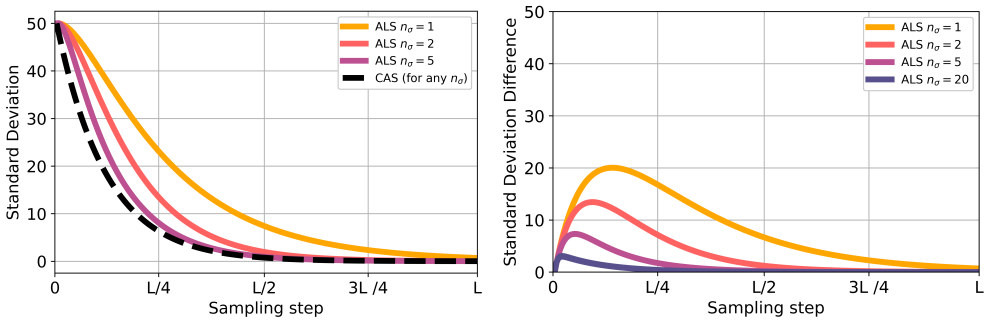
3.1 INCONSISTENCIES IN ALS

One can think of the ALS algorithm as a sequential series of Langevin Dynamics (inner loop in Algorithm 1) for decreasing levels of noise (outer loop). If allowed an infinite number of steps n_σ, the sampling process will properly produce samples from the data distribution.

In ALS, the score network is conditioned on geometrically decreasing noise (σ_i). In the unconditional case, this corresponds to dividing the score network by the noise level (*i.e.,* $s_\theta(\tilde{\boldsymbol{x}}, \sigma_i) = s_\theta(\tilde{\boldsymbol{x}})/\sigma_i$). Thus, in both conditional and unconditional cases, we make the assumption that the noise of the

sample at step i will be of variance σ_i^2, an assumption upon which the quality of the estimation of the score depends. While choosing a geometric progression of noise levels seems like a reasonable (though arbitrary) schedule to follow, we show that ALS does not ensure such schedule.

Assume we have the true score function s^* and begin sampling using a real image with some added zero-centered Gaussian noise of standard deviation $\sigma_0 = 50$. In Figure 1a, we illustrate how the intensity of the noise in the sample evolves through ALS and CAS, our proposed sampling, for a given sampling step size ϵ and a geometric schedule in this idealized scenario. We note that, although a large n_σ approaches the real geometric curve, it will only reach it at the limit ($n_\sigma \to \infty$ and $\epsilon \to 0$). Most importantly, Figure 1b highlights how even when the annealing process does converge, the progression of the noise is never truly geometric; we prove this formally in Proposition 1.



(a) Standard deviation of the noise in the image

(b) Difference between the standard deviation from ALS and CAS

Figure 1: Standard deviation during idealized sampling using a perfect score function s^*. The black curve in (a) corresponds to the true geometric progression, as demonstrated in Proposition 2.

Proposition 1. *Let s^* be the optimal score function from Eq. 3. Following the sampling described in Algorithm 1, the variance of the noise component in the sample $\boldsymbol{x}$ will remain greater than σ_t^2 at every step t.*

The proof is presented in Appendix F. In particular, for $n_\sigma < \infty$, sampling has not fully converged and the remaining noise is carried over to the next iteration of Langevin Sampling. It also follows that for any s_θ different from the optimal s^*, the actual noise at every iteration is expected to be even higher than for the best possible score function s^*.

3.2 Algorithm

We propose Consistent Annealed Sampling (CAS) as a sampling method that ensures the noise level will follow a prescribed schedule for any sampling step size ϵ and number of steps L. Algorithm 2 illustrates the process for a geometric schedule. Note that for a different schedule, β will instead depend on the step t, as in the general case, γ_t is defined as σ_{t+1}/σ_t.

Proposition 2. *Let s^* be the optimal score function from Eq. 3. Following the sampling described in Algorithm 2, the variance of the noise component in the sample $\boldsymbol{x}$ will consistently be equal to σ_t^2 at every step t.*

The proof is presented in Appendix G. Importantly, Proposition 2 holds no matter how many steps L we take to decrease the noise geometrically. For ALS, n_σ corresponds to the number of steps per level of noise. It plays a similar role in CAS: we simply dilate the geometric series of noise levels used during training by a factor of n_σ, such that $L_{\text{sampling}} = (L_{\text{training}} - 1)n_\sigma + 1$. Note that the proposition only holds when the initial sample is a corrupted image ($i.e.,\ \ \boldsymbol{x}_0 = \mathcal{I} + \sigma_0 \boldsymbol{z}_0$). However, by defining σ_0 as the maximum Euclidean distance between all pairs of training data points (Song and Ermon, 2020), the noise becomes in practice much greater than the true image; sampling with pure noise initialization ($i.e.,\ \ \boldsymbol{x}_0 = \sigma_0 \boldsymbol{z}_t$) becomes indistinguishable from sampling with data initialization.

4 BENEFITS OF THE EDS ON SYNTHETIC DATA AND IMAGE GENERATION

As previously mentioned, it has been suggested that one can obtain better samples (closer to the assumed data manifold) by taking the EDS of the last Langevin sample. We provide further evidence of this with synthetic data and standard image datasets.

It can first be observed that the sampling steps correspond to an interpolation between the previous point and the EDS, followed by the addition of noise.

Proposition 3. *Given a noise-conditional score function, the update rules from Algorithm 1 and Algorithm 2 are respectively equivalent to the following update rules:*

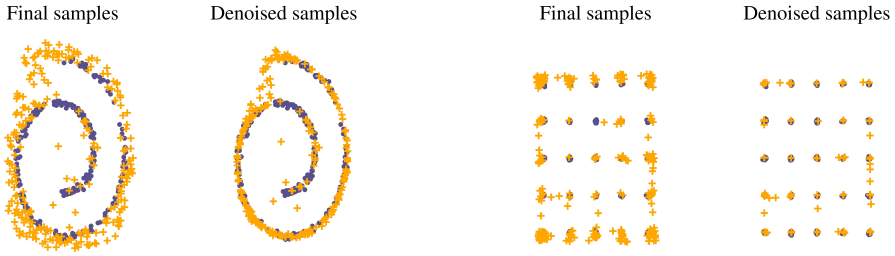
$$\boldsymbol{x} \leftarrow (1-\eta)\boldsymbol{x} + \eta H(\boldsymbol{x}, \sigma_i) + \sqrt{2\eta}\sigma_i \boldsymbol{z} \qquad \text{for } \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}) \text{ and } \eta = \frac{\epsilon}{\sigma_L^2}$$

$$\boldsymbol{x} \leftarrow (1-\eta)\boldsymbol{x} + \eta H(\boldsymbol{x}, \sigma_i) + \beta\sigma_{i+1} \boldsymbol{z}$$

The demonstration is in Appendix H. This result is equally true for an unconditional score network, with the distinction that η would no longer be independent of σ_i but rather linearly proportional to it.

Intuitively, this implies that the sampling steps slowly move the current sample towards a moving target (the EDS). If the sampling behaves appropriately, we expect the final sample $\boldsymbol{x}$ to be very close to the EDS, *i.e.,* $\boldsymbol{x} \approx H(\boldsymbol{x}, \sigma_L)$. However, if the sampling step size is inappropriate, or if the EDS does not stabilize to a fixed point near the end of the sampling, these two quantities may be arbitrarily far from one another. As we will show, the FIDs from Song and Ermon (2020) suffer from such distance.

The equivalence showed in Proposition 3 suggests instead to take the expected denoised sample at the end of the Langevin sampling as the final sample; this would be equivalent to the update rule $\boldsymbol{x} \leftarrow H(\boldsymbol{x}, \sigma_L)$ at the last step. Synthetic 2D examples shown in Figure 2 demonstrate the immediate benefits of this technique.



(a) Swiss Roll dataset (b) 25 Gaussians dataset

Figure 2: Langevin sampling on synthetic 2D experiments. Circles are real data points, crosses are generated data points. On both datasets, taking the EDS brings the samples much closer to the real data manifold.

We train a score network on CIFAR-10 (Krizhevsky et al., 2009) and report the FID from both ALS and CAS as a function of the sampling step size and of denoising in Figure 3. The first observation to be made is just how critical denoising is to the FID score for ALS, even as its effect cannot be perceived by the human eye. For CAS, we note that the score remains small for a much wider range of sampling step sizes when denoising. Alternatively, the sampling step size must be very carefully tuned to obtain results close to the optimal.

Figure 3 also shows that, with CAS, the FID of the final sample is approximately equal to the FID of the denoised samples for small sampling step sizes. Furthermore, we see a smaller gap in FID between denoised and non-denoised for larger sampling step sizes than ALS. This suggests that consistent sampling is resulting in the final sample being closer to the assumed data manifold (i.e., $\boldsymbol{x} \approx H_\theta(\boldsymbol{x}, \sigma_L)$).

Interestingly, when Song and Ermon (2020) improved their score matching method, they could not explain why the FID of their new model did not improve even though the generated images looked

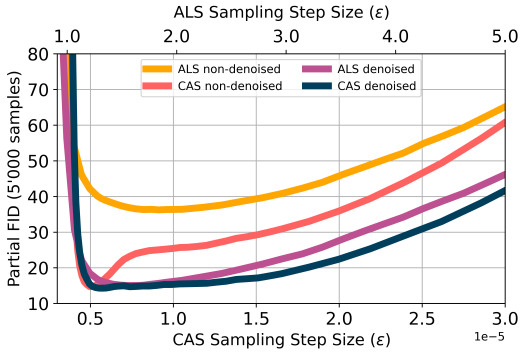


Figure 3: Partial estimate of FID (lower is better) as a function of the sampling step size on CIFAR-10, with $n_\sigma = 1$. The interactions between *consistent sampling* and denoising are shown.

better visually. To resolve that matter, they proposed the use of a new metric (Zhou et al., 2019) that did not have this issue. As shown in Figure 3, denoising resolves this mismatch.

5 Adversarial Formulation

The score network is trained to recover an uncorrupted image from a noisy input minimizing the l_2 distance between the two. However, it is well known from the image restoration literature that l_2 does not correlate well with human perception of image quality (Zhang et al., 2012; Zhao et al., 2016). One way to take advantage of the EDS would be to encourage the score network to produce an EDS that is more realistic from the perspective of a discriminator. Intuitively, this would incentivize the score network to produce more discernible features at inference time.

We propose to do so by training the score network to simultaneously minimize the score-matching loss function and maximize the probability of denoised samples being perceived as real by a discriminator. We use alternating gradient descent to sequentially train a discriminator for a determined number of steps at every score function update.

In our experiments, we selected the Least Squares GAN (LSGAN) (Mao et al., 2017) formulation as it performed best (see Appendix B for details). For an unconditional score network, the objective functions are as follows:

$$\max_{\phi} \; \mathbb{E}_{p(\boldsymbol{x})}\left[(D_\phi(\boldsymbol{x}) - 1)^2\right] + \mathbb{E}_{p(\tilde{\boldsymbol{x}},\boldsymbol{x},\sigma)}\left[(D_\phi(H_\theta(\tilde{\boldsymbol{x}},\sigma) + 1)^2\right] \tag{4}$$

$$\min_{\theta} \; \mathbb{E}_{p(\tilde{\boldsymbol{x}},\boldsymbol{x},\sigma)}\left[(D_\phi(H_\theta(\tilde{\boldsymbol{x}},\sigma)) - 1)^2 + \frac{\lambda}{2}\left\|\sigma s_\theta(\tilde{\boldsymbol{x}},\sigma) + \frac{\tilde{\boldsymbol{x}} - \boldsymbol{x}}{\sigma}\right\|_2^2\right], \tag{5}$$

where $H_\theta(\tilde{\boldsymbol{x}},\sigma) = s_\theta(\tilde{\boldsymbol{x}},\sigma)\sigma^2 + \tilde{\boldsymbol{x}}$ is the EDS derived from the score network. Eq. 4 is the objective function of the LSGAN discriminator, while Eq. 5 is the adversarial objective function of the score network derived from Eq. 1 and from the LSGAN objective function.

We note the similarities between these objective functions and those of an LSGAN adversarial autoencoder (Makhzani et al., 2015; Tolstikhin et al., 2017; Tran et al., 2018), with the distinction of using a denoising autoencoder H as opposed to a standard autoencoder.

As GANs favor quality over diversity, there is a concern that this hybrid objective function might decrease the diversity of samples produced by the ALS. In Section 6.1, we first study image generation improvements brought by this method and then address the diversity concerns with experiments on the 3-StackedMNIST (Metz et al., 2016) dataset in Section 6.2.

6 EXPERIMENTS

6.1 ABLATION STUDY

We ran experiments on CIFAR-10 (Krizhevsky et al., 2009) and LSUN-churches (Yu et al., 2015) with the score network architecture used by Song and Ermon (2020). We also ran similar experiments with an unconditional version of the network architecture by Ho et al. (2020), given that their approach is similar to Song and Ermon (2019) and they obtain very small FIDs. For the hybrid adversarial score matching approach, we used an unconditional BigGAN discriminator (Brock et al., 2018). We compared three factors in an ablation study: adversarial training, Consistent Annealed Sampling and denoising.

Details on how the experiments were conducted are found in Appendix B. Unsuccessful experiments with large images are also discussed in Appendix C. See also Appendix I for a discussion pertaining to the use of the Inception Score (Heusel et al., 2017), a popular metric for generative models.

Results for CIFAR-10 and LSUN-churches with Song and Ermon (2019) score network architecture are respectively shown in Table 1 and 2. Results for CIFAR-10 with Ho et al. (2020) score network architecture are shown in Table 3.

Table 1: [Non-denoised / Denoised FID] from 10k samples on CIFAR-10 (32x32) with Song and Ermon (2019) score network architecture

Sampling	Non-adversarial	Adversarial
non-consistent ($n_\sigma = 1$)	36.3 / 13.3	33.9 / 12.5
non-consistent ($n_\sigma = 5$)	33.7 / 10.9	31.5 / **10.2**
consistent ($n_\sigma = 1$)	14.7 / 12.3	13.4 / 11.6
consistent ($n_\sigma = 5$)	12.7 / 11.2	11.4 / **10.2**

Table 2: [Non-denoised / Denoised FID] from 10k samples on LSUN-Churches (64x64) with Song and Ermon (2019) score network architecture

Sampling	Non-adversarial	Adversarial
non-consistent ($n_\sigma = 1$)	43.2 / 40.3	42.4 / 38.6
non-consistent ($n_\sigma = 5$)	42.0 / 39.2	41.6 / 37.9
consistent ($n_\sigma = 1$)	41.5 / 40.7	40.2 / 39.1
consistent ($n_\sigma = 5$)	39.5 / **39.1**	38.2 / **37.6**

Table 3: [Non-denoised / Denoised FID] from 10k samples on CIFAR-10 (32x32) with Ho et al. (2020) unconditional score network architecture

Sampling	Non-adversarial	Adversarial
non-consistent ($n_\sigma = 1$)	25.29 / 7.47	26.75 / 7.98
non-consistent ($n_\sigma = 5$)	20.02 / **5.63**	20.98 / 6.02
consistent ($n_\sigma = 1$)	7.78 / 7.11	8.08 / 7.49
consistent ($n_\sigma = 5$)	6.19 / 6.11	6.43 / 6.40

We always observe an improvement in FID from denoising and by increasing n_σ from 1 to 5. We observe an improvement from using the adversarial approach with Song and Ermon (2019) network architecture, but not with the Ho et al. (2020) network architecture. We hypothesize that this is a limitation of the architecture of the discriminator since, as far as we know, no variant of BigGAN achieves an FID smaller than 6. Nevertheless, it remains advantageous for more simple architectures, as shown in Table 1 and 2. We observe that consistent sampling outperforms non-consistent sampling on the CIFAR-10 task at $n_\sigma = 1$, the quickest way to sample.

We calculated the FID of the non-consistent non-adversarial denoised models from 50k samples in order to compare our method with the recent work from Ho et al. (2020). We obtained a score of 3.65 on the CIFAR-10 task when sharing their architecture; a score close to their reported 3.17. Although not explicit in their approach, Ho et al. (2020) denoised their final sample. This suggests that taking the EDS and using an architecture akin to theirs were the two main reasons for outperforming Song and Ermon (2020). Of note, our method only trains the score network for 300k iterations, while Ho et al. (2020) trained their networks for more than 1 million iterations to achieve similar results.

6.2 Non-adversarial and Adversarial score networks have equally high diversity

To assess the diversity of generated samples, we evaluate our models on the 3-Stacked MNIST generation task (Metz et al., 2016) (128k images of 28x28), consisting of numbers from the MNIST dataset (LeCun et al., 1998) superimposed on 3 different channels. We trained non-adversarial and adversarial score networks in the same way as the other models. The results are shown in Table 4.

We see that each of the 1000 modes is covered, though the KL divergence is still inferior to PACGAN (Lin et al., 2018), meaning that the mode proportions are not perfectly uniform. Blindness to mode proportions is thought to be a fundamental limitation of score-based methods (Wenliang, 2020). Nevertheless, these results confirm a full mode coverage on a task where most GANs struggle and, most importantly, that using a hybrid objective does not hurt the diversity of the generated samples.

3-Stacked MNIST		
	Modes (Max 1000)	KL
DCGAN (Radford et al., 2015b)	99.0	3.40
ALI (Dumoulin et al., 2016)	16.0	5.40
Unrolled GAN (Metz et al., 2016)	48.7	4.32
VEEGAN (Srivastava et al., 2017)	150.0	2.95
PacDCGAN2 (Lin et al., 2017b)	1000.0	0.06
WGAN-GP (Kumar et al., 2019; Gulrajani et al., 2017)	959.0	0.73
PresGAN (Dieng et al., 2019)	999.6	0.115
MEG (Kumar et al., 2019)	1000.0	0.03
Non-adversarial DSM (ours)	1000.0	1.36
Adversarial DSM (ours)	1000.0	1.36

Table 4: As in Lin et al. (2018), we generated 26k samples and evaluated the mode coverage and KL divergence based on the predicted modes from a pre-trained MNIST classifier.

7 Conclusion

We proposed Consistent Annealed Sampling as an alternative to Annealed Langevin Sampling, which ensures the expected geometric progression of the noise and brings the final samples closer to the data manifold. We showed how to extract the expected denoised sample and how to use it to further improve the final Langevin samples. We proposed a hybrid approach between GAN and score matching. With experiments on synthetic and standard image datasets; we showed that these approaches generally improved the quality/diversity of the generated samples.

We found equal diversity (coverage of all 1000 modes) for the adversarial and non-adversarial variant of the difficult StackedMNIST problem. Since we also observed better performance (from lower FIDs) in our other adversarial models trained on images, we conclude that making score matching adversarial increases the quality of the samples without decreasing diversity. These findings imply that score matching performs better than most GANs and on-par with state-of-the-art GANs. Furthermore, our results suggest that hybrid methods, combining multiple generative techniques together, are a very promising direction to pursue.

As future work, these models should be scaled to larger batch sizes on high-resolution images, since GANs have been shown to produce outstanding high-resolution images at very large batch sizes (2048 or more). We also plan to further study the theoretical properties of CAS by considering its corresponding stochastic differential equation.

REFERENCES

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11918–11930, 2019.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709, 2005.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Martin Raphan and Eero P Simoncelli. Least squares estimation without priors or supervision. *Neural computation*, 23(2):374–420, 2011.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

Gareth O Roberts, Richard L Tweedie, et al. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015a.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363, 2019.

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

Alexia Jolicoeur-Martineau and Ioannis Mitliagkas. Connections between support vector machines, wasserstein distance and gradient-penalty gans. *arXiv preprint arXiv:1910.06922*, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.

Anirudh Goyal Alias Parth Goyal, Nan Rosemary Ke, Surya Ganguli, and Yoshua Bengio. Variational walkback: Learning a transition operator as a stochastic recurrent net. In *Advances in Neural Information Processing Systems*, pages 4392–4402, 2017.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017a.

Zengyi Li, Yubei Chen, and Friedrich T Sommer. Learning energy-based models in high-dimensional spaces with multi-scale denoising score matching. *arXiv*, pages arXiv–1910, 2019.

Jae Hyun Lim, Aaron C. Courville, Christopher Joseph Pal, and Chin-Wei Huang. Ar-dae: Towards unbiased neural entropy gradient estimation. *ArXiv*, abs/2006.05164, 2020.

Herbert Robbins. *An empirical Bayes approach to statistics*. Office of Scientific Research, US Air Force, 1955.

Koichi Miyasawa. An empirical bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist*, 38(181-188):1–2, 1961.

Saeed Saremi and Aapo Hyvarinen. Neural empirical bayes. *Journal of Machine Learning Research*, 20:1–23, 2019.

Zahra Kadkhodaie and Eero P Simoncelli. Solving linear inverse problems using the prior implicit in a denoiser. *arXiv preprint arXiv:2007.13640*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Li F Fei-Fei, and Michael Bernstein. Hype: A benchmark for human eye perceptual evaluation of generative models. In *Advances in Neural Information Processing Systems*, pages 3449–3461, 2019.

Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. A comprehensive evaluation of full reference image quality assessment algorithms. In *2012 19th IEEE International Conference on Image Processing*, pages 1477–1480. IEEE, 2012.

Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE, 2017.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Ngoc-Trung Tran, Tuan-Anh Bui, and Ngai-Man Cheung. Dist-gan: An improved gan using distance constraints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 370–385, 2018.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In *Advances in neural information processing systems*, pages 1498–1507, 2018.

Li K Wenliang. Blindness of score-based methods to isolated components and mixing proportions. *arXiv preprint arXiv:2008.10087*, 2020.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015b.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. *arXiv preprint arXiv:1712.04086*, 2017b.

Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019.

Adji B Dieng, Francisco JR Ruiz, David M Blei, and Michalis K Titsias. Prescribed generative adversarial networks. *arXiv preprint arXiv:1910.04302*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

Alexia Jolicoeur-Martineau. On relativistic f-divergences. *arXiv preprint arXiv:1901.02474*, 2019.

Gauthier Gidel, Reyhane Askari Hemmat, Mohammad Pezeshki, Rémi Le Priol, Gabriel Huang, Simon Lacoste-Julien, and Ioannis Mitliagkas. Negative momentum for improved game dynamics. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1802–1811, 2019.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *CoRR*, abs/1811.12231, 2018. URL http://arxiv.org/abs/1811.12231.

Shane T. Barratt and Rishi Sharma. A note on the inception score. *ArXiv*, abs/1801.01973, 2018.

APPENDICES

A BROADER IMPACT

Unfortunately, these improvements in image generation come at a very high computational cost, meaning that the ability to generate high-resolution images is constrained by the availability of large computing resources (TPUs or clusters of 8+ GPUs). This is mainly due to the architectures used in this paper, while adding a discriminator further adds to the training computational load.

B EXPERIMENTS DETAILS

Table 5: Sampling learning rates (ϵ)

Network architecture	Dataset	consistent	n_σ	ϵ
Song and Ermon (2020)	CIFAR-10	No	1	1.8e-5
Song and Ermon (2020)	CIFAR-10	No	5	3.6e-6
Song and Ermon (2020)	CIFAR-10	Yes	1	5.6e-6
Song and Ermon (2020)	CIFAR-10	Yes	5	1.1e-6
Song and Ermon (2020)	LSUN-Churches	No	1	4.85e-6
Song and Ermon (2020)	LSUN-Churches	No	5	9.7e-7
Song and Ermon (2020)	LSUN-Churches	Yes	1	2.8e-6
Song and Ermon (2020)	LSUN-Churches	Yes	5	4.5e-7
Ho et al. (2020)	CIFAR-10	No	1	1.6e-5
Ho et al. (2020)	CIFAR-10	No	5	4.0e-6
Ho et al. (2020)	CIFAR-10	Yes	1	5.45e-6
Ho et al. (2020)	CIFAR-10	Yes	5	1.05e-6
Song and Ermon (2020)	3-StackedMNIST	Yes	1	5.0e-6

Following the recommendations from Song and Ermon (2020), we chose $\sigma_1 = 50$ and $L = 232$ on CIFAR-10, and $\sigma_1 = 140$, $L = 788$ on LSUN-Churches, both with $\sigma_L = 0.01$. We used a batch size of 128 in all models. We first swept summarily the training checkpoint (saved at every 2.5k iterations), the Exponential Moving Average (EMA) coefficient from {.999, .9999}, and then swept over the sampling step size ϵ with approximately 2 significant number precision. The values reported in Table 1 correspond to the sampling step size that minimized the denoised FID for every n_σ (See Table 5). We used the same sampling step sizes for adversarial and non-adversarial. Empirically, the optimal sampling step size is found for a certain n_σ and is extrapolated to other precision levels by solving $\beta_{n'_\sigma} = \beta_{n_\sigma}\sqrt{n_\sigma / n'_\sigma}$ for the consistent algorithm. In the non-consistent algorithm, we found the best sampling step size at $n_\sigma = 1$ and divided by 5 to obtain a starting point to find the optimal value at $n_\sigma = 5$. The best EMA values were found to be .9999 in CIFAR-10 and .999 in LSUN-churches. The number of score network training iterations was 300k on CIFAR-10 and 200k on LSUN-Churches.

Of note, Song and Ermon (2020) used non-denoised non-consistent sampling with $n_\sigma = 5$ and $n_\sigma = 4$ for CIFAR-10 and LSUN-Churches respectively. However, they did not use the same learning rates (we tuned ours more precisely) and they used bigger images for LSUN-Churches.

Regarding the adversarial approach, we swept for the GAN loss function, the number of discriminator steps per score network steps ($n_D \in \{1, 2\}$), the Adam optimizer (Kingma and Ba, 2014) parameters, and the hyperparameter λ (see Eq. 5) based on quick experiments on CIFAR-10. LSGAN (Mao et al., 2017) yielded the best FID scores among all other loss functions considered, namely the original GAN loss function (Goodfellow et al., 2014), HingeGAN (Lim and Ye, 2017), as well as their relativistic counterparts (Jolicoeur-Martineau, 2018; 2019). Note that the saturating variant (see Goodfellow et al. (2014)) on LSGAN worked as well as its non-saturating version; we did not use it for simplicity. Following the trend towards zero or negative momentum (Gidel et al., 2019), we used the Adam optimizer with hyperparameters $(\beta_1, \beta_2) = (-.5, .9)$ for the discriminator and $(\beta_1, \beta_2) =$

$(0, .9)$ for the score network. These values were found by sweeping over $\beta_1 \in \{.9, .5, 0, -.5\}$ and $\beta_2 \in \{.9, .999\}$. We found the simple setting $\lambda = 1$ to perform comparatively better than more complex weighting schemes. We used $n_D = 1$ on the experiments with Song and Ermon (2020) network architecture and $n_D = 2$ on the experiments with Ho et al. (2020) network architecture.

The 3-Stacked MNIST experiment was conducted with an arbitrary EMA of .999. The sampling step size was broadly swept upon. Following (Song and Ermon, 2020) hyperparameter recommendations, we obtained $\sigma_1 = 50$, $L = 200$.

For the synthetic 2D experiments, we used Langevin sampling with $n_\sigma = 10$, $\epsilon = .2$, and $\sigma = .1$.

C SUPPLEMENTARY EXPERIMENTS

Due to limited computing resources (4 V100 GPUs), the training of models on FFHQ (70k images) in 256x256 (Karras et al., 2019), with the same setting as previously done by Song and Ermon (2020), was impossible. Using a reduced model yielded very poor results. The adversarial version performed worse than the other; we suspect this was the case due to the mini-batch of size 32, our computational limit, while the BigGAN architecture normally assumes very large batch sizes of 2048 when working with images of that size (256x256 or higher).

D OPTIMAL CONDITIONAL SCORE FUNCTION

Recall the loss from Eq. 1

$$\mathcal{L}[s] = \frac{1}{2L} \sum_{i=1}^{L} \mathbb{E}_{\tilde{\boldsymbol{x}} \sim q_{\sigma_i}(\tilde{\boldsymbol{x}}|\boldsymbol{x}), \boldsymbol{x} \sim p(\boldsymbol{x})} \left[\| s(\tilde{\boldsymbol{x}}, \sigma_i) - \nabla_{\tilde{\boldsymbol{x}}} \log q_{\sigma_i}(\tilde{\boldsymbol{x}}) \|_2^2 \right]$$

Then, the minimizer s^* of $\mathcal{L}[s]$ would be such that, $s^*(\tilde{\boldsymbol{x}}, \sigma_i) = \nabla_{\tilde{\boldsymbol{x}}} \log q_{\sigma_i}(\tilde{\boldsymbol{x}}) \quad \forall i \in \{1, ..., L\}$.

$$\begin{aligned}
\nabla_{\tilde{\boldsymbol{x}}} \log q_\sigma(\tilde{\boldsymbol{x}}) &= \frac{1}{q_\sigma(\tilde{\boldsymbol{x}})} \nabla_{\tilde{\boldsymbol{x}}} \int p(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{1}{q_\sigma(\tilde{\boldsymbol{x}})} \int p(\boldsymbol{x}) \nabla_{\tilde{\boldsymbol{x}}} q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{1}{q_\sigma(\tilde{\boldsymbol{x}})} \int p(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \nabla_{\tilde{\boldsymbol{x}}} \log q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{1}{q_\sigma(\tilde{\boldsymbol{x}})} \int p(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \frac{\boldsymbol{x} - \tilde{\boldsymbol{x}}}{\sigma^2} d\boldsymbol{x} \\
&= \frac{1}{q_\sigma(\tilde{\boldsymbol{x}})} \frac{1}{\sigma^2} \left(\int q_\sigma(\boldsymbol{x}|\tilde{\boldsymbol{x}}) q_\sigma(\tilde{\boldsymbol{x}}) \boldsymbol{x} d\boldsymbol{x} - \tilde{\boldsymbol{x}} q_\sigma(\tilde{\boldsymbol{x}}) \right) \\
&= \frac{\mathbb{E}_{\boldsymbol{x} \sim q_\sigma(\boldsymbol{x}|\tilde{\boldsymbol{x}})}[\boldsymbol{x}] - \tilde{\boldsymbol{x}}}{\sigma^2}
\end{aligned}$$

making use of the fact that $q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \mathcal{N}(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2 \boldsymbol{I})$, $q_\sigma(\tilde{\boldsymbol{x}}) \triangleq \int p(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$ and that $q_\sigma(\boldsymbol{x}|\tilde{\boldsymbol{x}}) = \frac{p(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})}{q_\sigma(\tilde{\boldsymbol{x}})}$

E OPTIMAL UNCONDITIONAL SCORE FUNCTION

As the explicit optimal score function is obtained in Appendix D for the conditional case, a similar result can be obtained for the unconditional case. Recall the loss from Eq. 1

$$\mathcal{L}[s] = \frac{1}{2L}\sum_{i=1}^{L}\mathbb{E}_{\tilde{\boldsymbol{x}}\sim q_{\sigma_i}(\tilde{\boldsymbol{x}}|\boldsymbol{x}),\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\left\|s(\tilde{\boldsymbol{x}}) + \frac{(\tilde{\boldsymbol{x}}-\boldsymbol{x})}{\sigma_i}\right\|_2^2\right] \tag{6}$$

$$= \frac{1}{2}\mathbb{E}_{\tilde{\boldsymbol{x}}\sim q_{\sigma}(\tilde{\boldsymbol{x}}|\boldsymbol{x}),\boldsymbol{x}\sim p(\boldsymbol{x}),\sigma\sim p(\sigma)}\left[\left\|s(\tilde{\boldsymbol{x}}) + \frac{(\tilde{\boldsymbol{x}}-\boldsymbol{x})}{\sigma}\right\|_2^2\right], \tag{7}$$

where $p(\sigma)$ is chosen to be a discrete uniform distribution over a specific set of values. We use this expectation formulation over σ to obtain a more general result; the choice of $p(\sigma)$ is not important for this derivation.

Solving with calculus of variations, we get:

$$\begin{aligned}
&\frac{\partial\mathcal{L}}{\partial s} = \int\int q_\sigma(\tilde{\boldsymbol{x}},\boldsymbol{x},\sigma)\left(s(\tilde{\boldsymbol{x}}) + \frac{\tilde{\boldsymbol{x}}-\boldsymbol{x}}{\sigma}\right)d\boldsymbol{x}d\sigma = 0\\
&\iff s(\tilde{\boldsymbol{x}})q(\tilde{\boldsymbol{x}}) = \int\int q_\sigma(\tilde{\boldsymbol{x}},\boldsymbol{x})p(\sigma)\left(\frac{\boldsymbol{x}-\tilde{\boldsymbol{x}}}{\sigma}\right)d\boldsymbol{x}d\sigma\\
&\iff s(\tilde{\boldsymbol{x}})q(\tilde{\boldsymbol{x}}) = \mathbb{E}_{\sigma\sim p(\sigma)}\left[\int q_\sigma(\boldsymbol{x}|\tilde{\boldsymbol{x}})q(\tilde{\boldsymbol{x}})\left(\frac{\boldsymbol{x}-\tilde{\boldsymbol{x}}}{\sigma}\right)d\boldsymbol{x}\right]\\
&\iff s(\tilde{\boldsymbol{x}}) = \mathbb{E}_{\sigma\sim p(\sigma)}\left[\int q_\sigma(\boldsymbol{x}|\tilde{\boldsymbol{x}})\left(\frac{\boldsymbol{x}-\tilde{\boldsymbol{x}}}{\sigma}\right)d\boldsymbol{x}\right]\\
&\iff s(\tilde{\boldsymbol{x}}) = \mathbb{E}_{\sigma\sim p(\sigma)}\left[\frac{\mathbb{E}_{\boldsymbol{x}\sim q_\sigma(\boldsymbol{x}|\tilde{\boldsymbol{x}})}[\boldsymbol{x}]-\tilde{\boldsymbol{x}}}{\sigma}\right]\\
&\iff s(\tilde{\boldsymbol{x}}) = \mathbb{E}_{\sigma\sim p(\sigma)}\left[s^*(\tilde{\boldsymbol{x}},\sigma)\right]
\end{aligned}$$

Specifically, for our discrete choice of noise levels, we have that the critical point is achieved when

$$s(\tilde{\boldsymbol{x}}) = \frac{1}{L}\sum_{i=1}^{L}\left(\frac{\mathbb{E}_{\boldsymbol{x}\sim q_{\sigma_i}(\boldsymbol{x}|\tilde{\boldsymbol{x}})}[\boldsymbol{x}]-\tilde{\boldsymbol{x}}}{\sigma_i}\right).$$

Let us now consider the scenario where the data distribution is simply a Dirac in $\boldsymbol{x_0}$: $p(\boldsymbol{x}) = \delta_{\boldsymbol{x_0}}$. In that case, the EDS is trivial: $H^*(\tilde{\boldsymbol{x}},\sigma) \triangleq \mathbb{E}_{\boldsymbol{x}\sim q_\sigma(\boldsymbol{x}|\tilde{\boldsymbol{x}})}[\boldsymbol{x}] = \boldsymbol{x_0}$. This gives an interesting insight into the unconditional score network. Letting $\frac{1}{\sigma} = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{\sigma_i}$, we get:

$$s(\tilde{\boldsymbol{x}}) = \frac{\boldsymbol{x}_0-\tilde{\boldsymbol{x}}}{\sigma} \implies s(\tilde{\boldsymbol{x}},\sigma_i) = \frac{s(\tilde{\boldsymbol{x}})}{\sigma_i} = \frac{\boldsymbol{x}_0-\tilde{\boldsymbol{x}}}{\sigma\sigma_i}.$$

This should be compared to the true value of the conditional network $s(\tilde{\boldsymbol{x}},\sigma_i) = \frac{\boldsymbol{x}_0-\tilde{\boldsymbol{x}}}{\sigma_i^2}$. We see that in the denominator of the unconditional score network, a noise level is replaced by the harmonic mean of noise levels. In particular, for σ_i far from its mean, the approximation will get inaccurate:

- For large noise values, the unconditional score network will overestimate the true score function, leading to a larger effective step size during sampling.
- For small noise values, the unconditional score network will underestimate the true score function, leading to samples not diffusing as much as they should.

F ALS NON-GEOMETRIC PROOF

Proposition 1. *Let s^* be the optimal score function from Eq. 3. Following the sampling described in Algorithm 1, the variance of the noise component in the sample $\boldsymbol{x}$ will remain greater than σ_t^2 at every step t.*

Proof. Assume at the start of an iteration of Langevin Sampling that the point $\boldsymbol{x}$ is comprised of an *image component* and of a *noise component* denoted $v_0\boldsymbol{z}_0$ for $\boldsymbol{z}_0 \sim \mathcal{N}(0,\boldsymbol{I})$. We assume from the

proposition statement that the Langevin Sampling is performed at the level of variance σ_t^2, meaning the update rule is as follows:

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \eta\sigma_t^2 s^*(\boldsymbol{x}, \sigma_t) + \sigma_t\sqrt{2\eta}\boldsymbol{z}$$

for $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$ and $0 < \eta < 1$. From Eq. 3, we get:

$$\begin{aligned}\boldsymbol{x} \leftarrow &\boldsymbol{x} + \eta(\mathbb{E}_{\boldsymbol{x}' \sim q_{v_0}(\boldsymbol{x}'|\boldsymbol{x})}[\boldsymbol{x}'] - \boldsymbol{x}) + \sigma_t\sqrt{2\eta}\boldsymbol{z} \\ =&(1-\eta)\boldsymbol{x} + \eta\mathbb{E}_{\boldsymbol{x}' \sim q_{v_0}(\boldsymbol{x}'|\boldsymbol{x})}[\boldsymbol{x}'] + \sigma_t\sqrt{2\eta}\boldsymbol{z}.\end{aligned}$$

The noise component of $(1-\eta)\boldsymbol{x}$ and $\sigma_t\sqrt{2\eta}\boldsymbol{z}$ can then be summed as:

$$(1-\eta)v_0\boldsymbol{z}_0 + \sigma_t\sqrt{2\eta}\boldsymbol{z}.$$

Making use of the fact that both sources of noise follow independent normal distributions, the sum will be normally distributed, centered and of variance v_1^2, with

$$v_1^2 = v_0^2(1-\eta)^2 + 2\eta\sigma_t^2.$$

Applying the same steps allows us to examine the variance after multiple Langevin Sampling iterations

$$\begin{aligned}v_2^2 =& v_1^2(1-\eta)^2 + 2\eta\sigma_t^2 \\ =& v_0^2(1-\eta)^4 + 2\eta\sigma_t^2 + 2\eta\sigma_t^2(1-\eta)^2 \\ &\vdots \\ v_n^2 =& v_0^2(1-\eta)^{2n} + 2\eta\sigma_t^2\sum_{i=0}^{n-1}(1-\eta)^{2i} \\ =& v_0^2(1-\eta)^{2n} + \frac{2\sigma_t^2}{2-\eta}(1-(1-\eta)^{2n}).\end{aligned}$$

From there, the two following statements can be obtained:

$$\begin{aligned}&(1)\quad \lim_{n\to\infty} v_n = \sigma_t\sqrt{\frac{2}{2-\eta}} > \sigma_t \\ &(2)\quad \frac{dv_n}{dn} < 0 \iff \eta < 2 - \frac{2\sigma_t^2}{v_0^2}.\end{aligned}$$

From these observations, we understand that v_n^2 is monotonically decreasing (under conditions generally respected in practice) but converges to a point superior to σ_t^2 after an infinite number of Langevin Sampling steps. We then conclude that for all n, the variance of the noise component in the sample will always exceed σ_t^2. We also note that this will be true across the full sampling, at every step t.

In the particular case where σ_t corresponds to a geometrically decreasing series, it means that even given an optimal score function, the standard deviation of the noise component cannot follow its prescribed schedule.

□

G CAS GEOMETRIC PROOF

Proposition 2. *Let s^* be the optimal score function from Eq. 3. Following the sampling described in Algorithm 2, the variance of the noise component in the sample $\boldsymbol{x}$ will consistently be equal to σ_t^2 at every step t.*

Proof. Let us first define β to be equal to $\sqrt{1-(1-\eta)^2/\gamma^2}$ with $\eta = \epsilon/\sigma_L^2$ and γ defined as Eq. 2. At the start of Algorithm 2, assume that $\boldsymbol{x}$ is comprised of an *image component* and a *noise component* denoted $\sigma_0\boldsymbol{z}_0$, where $\boldsymbol{z}_0 \sim \mathcal{N}(0, \boldsymbol{I})$ and $\sigma_0 = \sigma_1/\gamma$. We will proceed by induction to show that the noise component at step t will be a Gaussian of variance σ_t^2 for every t.

The first induction step is trivial. Assume the noise component of $\boldsymbol{x}_t$ to be $\sigma_t \boldsymbol{z}_t$, where $\boldsymbol{z}_t \sim \mathcal{N}(0, \boldsymbol{I})$. Following Algorithm 2, the update step will be:

$$\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t + \eta \sigma_t^2 s^*(\boldsymbol{x}_t, \sigma_t) + \sigma_{t+1} \beta \boldsymbol{z},$$

with $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$ and $0 < \eta < 1$. From Eq. 3, we get

$$\begin{aligned}\boldsymbol{x}_{t+1} \leftarrow & \boldsymbol{x}_t + \eta(\mathbb{E}_{\boldsymbol{x} \sim q_{\sigma_t}(\boldsymbol{x}|\boldsymbol{x}_t)}[\boldsymbol{x}] - \boldsymbol{x}_t) + \sigma_{t+1} \beta \boldsymbol{z} \\ = & (1-\eta)\boldsymbol{x}_t + \eta \mathbb{E}_{\boldsymbol{x} \sim q_{\sigma_t}(\boldsymbol{x}|\boldsymbol{x}_t)}[\boldsymbol{x}] + \sigma_{t+1} \beta \boldsymbol{z}.\end{aligned}$$

The noise component from $(1-\eta)\boldsymbol{x}_t$ and $\sigma_{t+1}\beta\boldsymbol{z}$ can then be summed as:

$$\sigma_t (1-\eta) \boldsymbol{z}_t + \sigma_{t+1} \beta \boldsymbol{z}.$$

Making use of the fact that both sources of noise follow independent normal distributions, the sum will be normally distributed, centered and of variance:

$$\sigma_t^2 (1-\eta)^2 + \sigma_{t+1}^2 \beta^2 = \sigma_{t+1}^2 \left[\left(\frac{1-\eta}{\gamma} \right)^2 + \beta^2 \right] = \sigma_{t+1}^2.$$

By induction, the noise component of the sample $\boldsymbol{x}$ will follow a Gaussian distribution of variance $\sigma_i^2 \ \forall i \in \{0, ..., L\}$. In the particular case where σ_i corresponds to a geometrically decreasing series, it means that given an optimal score function, the standard deviation of the noise component will follow its prescribed schedule. □

H UPDATE RULE

Proposition 3. *Given a noise-conditional score function, the update rules from Algorithm 1 and Algorithm 2 are equivalent to the respective following update rules:*

$$\boldsymbol{x} \leftarrow (1-\eta)\boldsymbol{x} + \eta H(\boldsymbol{x}, \sigma_i) + \sqrt{2\eta}\sigma_i \boldsymbol{z} \qquad \text{for } \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}) \text{ and } \eta = \frac{\epsilon}{\sigma_L^2}$$

$$\boldsymbol{x} \leftarrow (1-\eta)\boldsymbol{x} + \eta H(\boldsymbol{x}, \sigma_i) + \beta \sigma_{i+1} \boldsymbol{z}$$

Proof. Recall from Algorithm 1 that $\alpha_i = \epsilon \frac{\sigma_i^2}{\sigma_L^2} = \eta \sigma_i^2$. Then, the update rule is as follows:

$$\begin{aligned}\boldsymbol{x} &\leftarrow \boldsymbol{x} + \alpha_i s(\boldsymbol{x}, \sigma_i) + \sqrt{2\alpha_i}\, \boldsymbol{z} \\ &= \boldsymbol{x} + \eta \sigma_i^2 \left(\frac{H(\boldsymbol{x}) - \boldsymbol{x}}{\sigma_i^2} \right) + \sqrt{2\alpha_i}\, \boldsymbol{z} \\ &= (1-\eta)\boldsymbol{x} + \eta H(\boldsymbol{x}) + \sqrt{2\eta}\sigma_i \boldsymbol{z}\end{aligned}$$

The same thing can be proven for Algorithm 2 in the very same way. □

I INCEPTION SCORE (IS)

While the FID is improved by applying the EDS to image samples, the Inception Score is not. Convolutional neural networks suffer from texture bias (Geirhos et al., 2018). Since the IS is built upon convolution layers, this flaw is also strongly present in the metric. Designed to answer the question of how easy it is to recover the class of an image, it tends to bias towards within-class texture similarity (Barratt and Sharma, 2018).

Since we denoise the final image, we are evaluating the expected lower level of details across all classes. Therefore, the denoiser will confound the textures used by the IS to distinguish between classes, invariably worsening the score. Since the FID has already been shown to be more consistent with the level of noise than the IS (Heusel et al., 2017), and since ALS methods are particularly prone to inject class-specific imperceptible noise, we would recommend against its use to compare within and between score matching models.

J UNCURATED SAMPLES

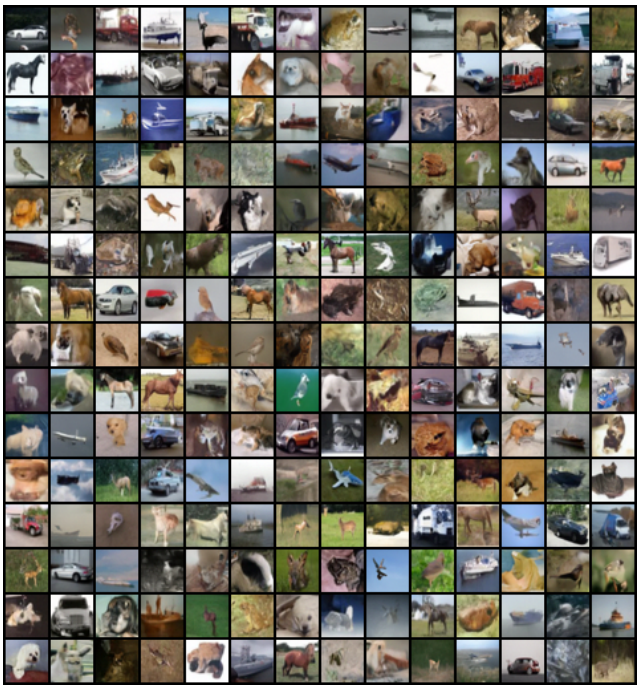


Figure 4: CIFAR-10 Non-adversarial non-consistent $n_\sigma = 1$

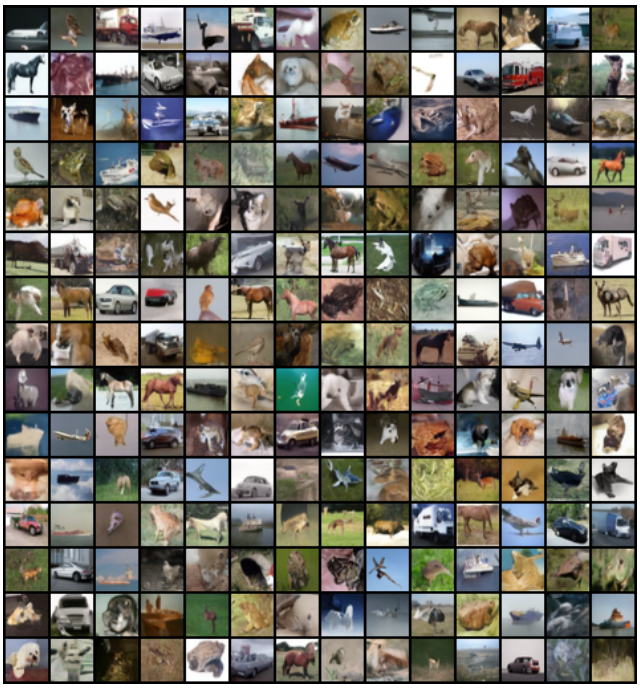


Figure 5: CIFAR-10 Adversarial non-consistent $n_\sigma = 1$

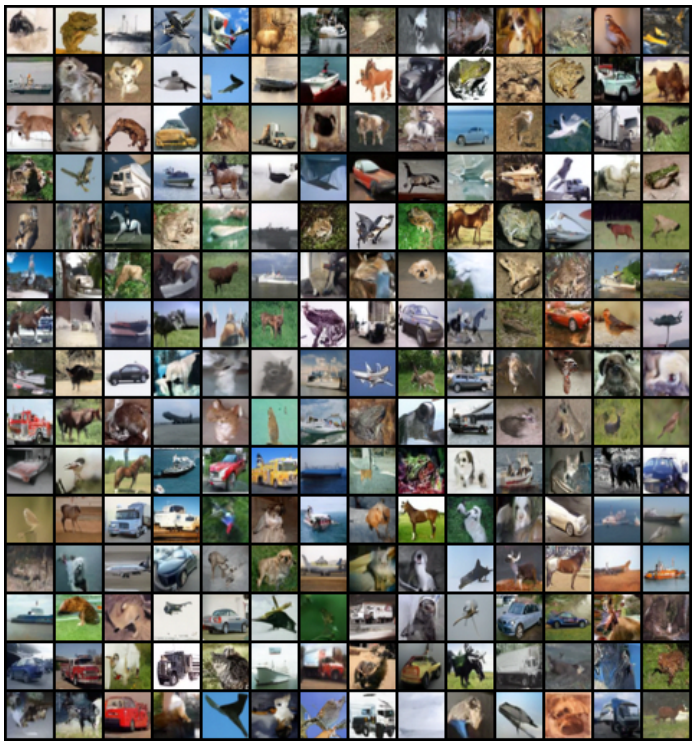


Figure 6: CIFAR-10 Non-adversarial non-consistent $n_\sigma = 5$

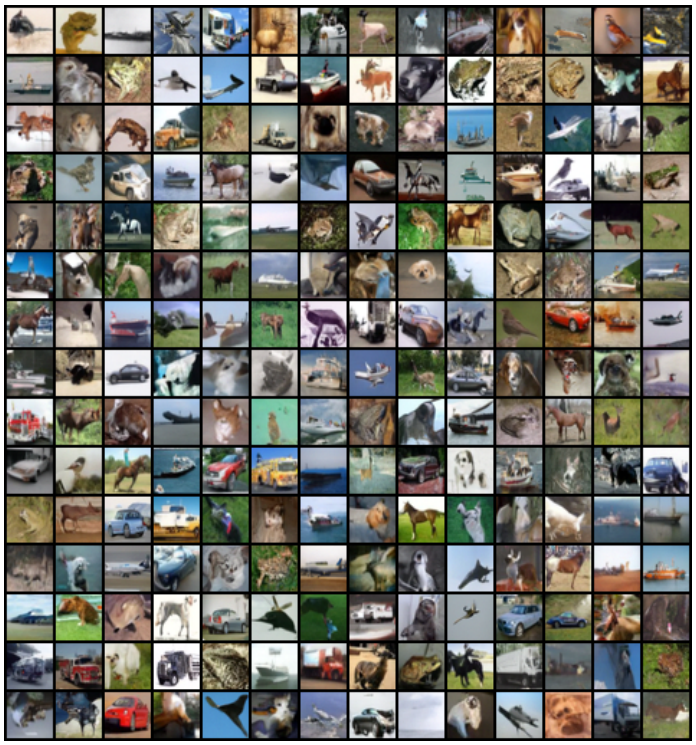


Figure 7: CIFAR-10 Adversarial non-consistent $n_\sigma = 5$

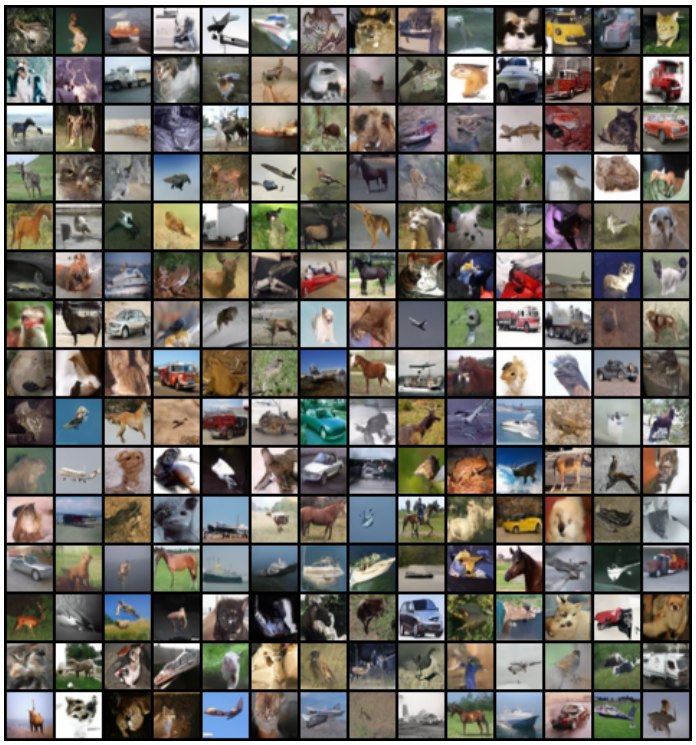


Figure 8: CIFAR-10 Non-adversarial consistent $n_\sigma = 1$

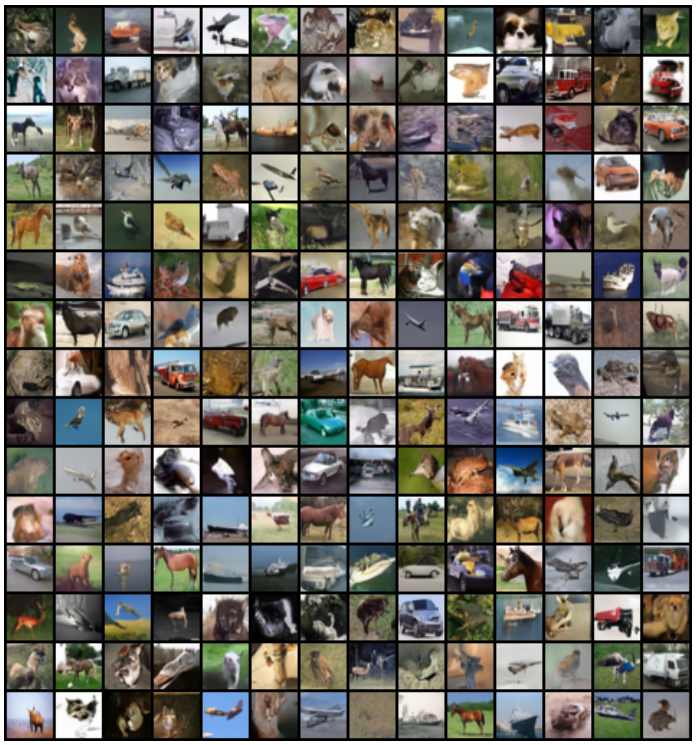


Figure 9: CIFAR-10 Adversarial consistent $n_\sigma = 1$

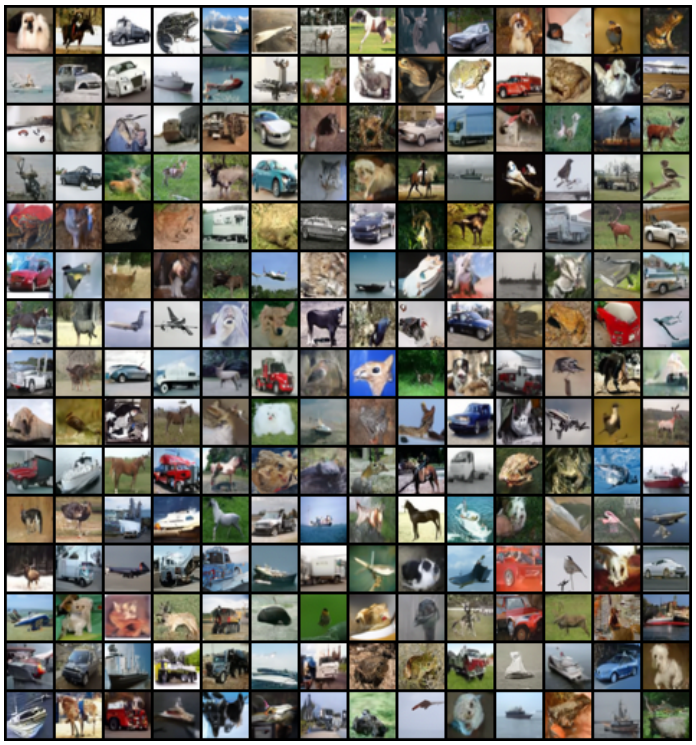


Figure 10: CIFAR-10 Non-adversarial consistent $n_\sigma = 5$

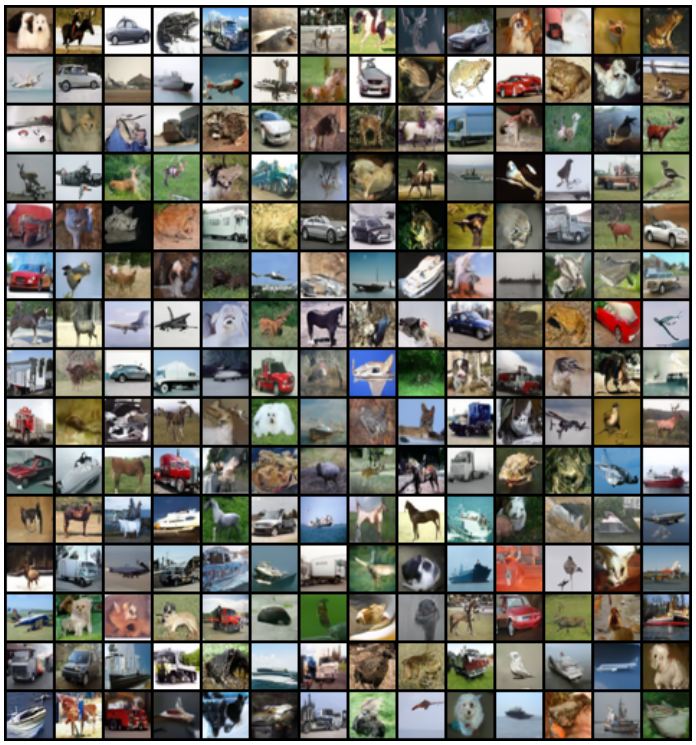


Figure 11: CIFAR-10 Adversarial consistent $n_\sigma = 5$

Figure 12: LSUN-Churches Non-adversarial non-consistent $n_\sigma = 1$

Figure 13: LSUN-Churches Adversarial non-consistent $n_\sigma = 1$

Figure 14: LSUN-Churches Non-adversarial non-consistent $n_\sigma = 5$

Figure 15: LSUN-Churches Adversarial non-consistent $n_\sigma = 5$

Figure 16: LSUN-Churches Non-adversarial consistent $n_\sigma = 1$

Figure 17: LSUN-Churches Adversarial consistent $n_\sigma = 1$

Figure 18: LSUN-Churches Non-adversarial consistent $n_\sigma = 5$

Figure 19: LSUN-Churches Adversarial consistent $n_\sigma = 5$

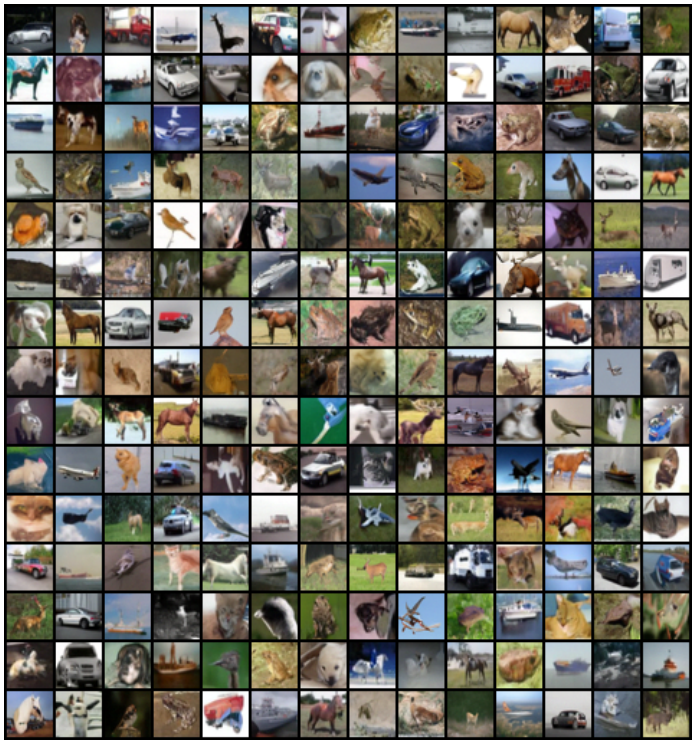


Figure 20: Ho et al. (2020) network architecture with CIFAR-10 Non-adversarial non-consistent $n_\sigma = 1$

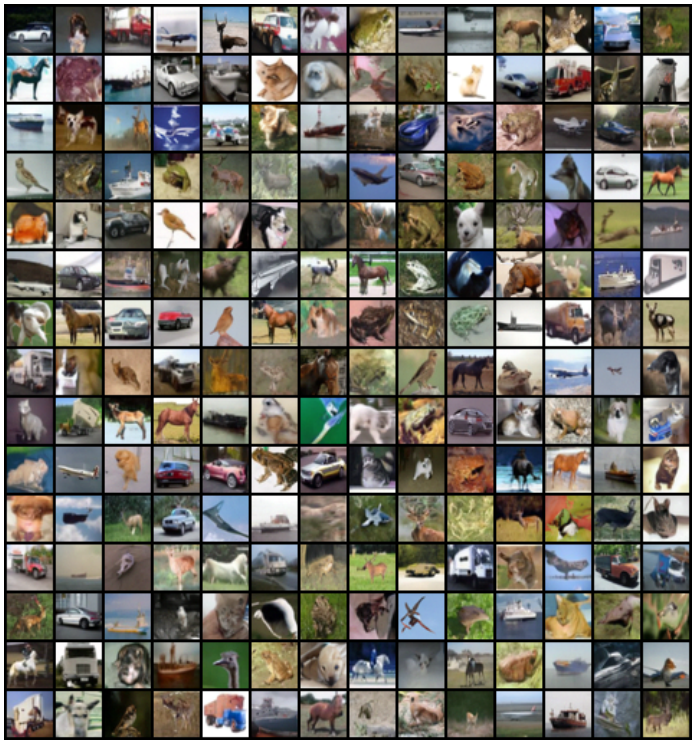


Figure 21: Ho et al. (2020) network architecture with CIFAR-10 Adversarial non-consistent $n_\sigma = 1$

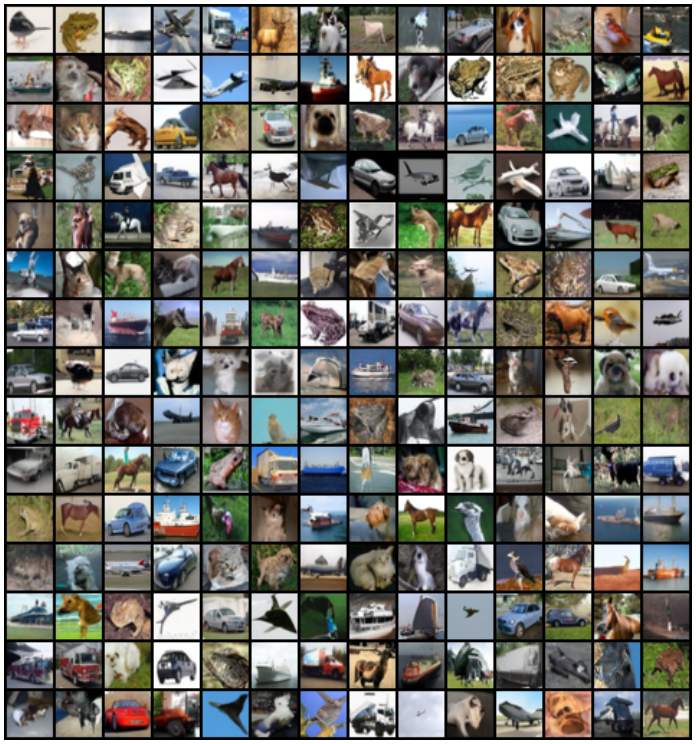


Figure 22: Ho et al. (2020) network architecture with CIFAR-10 Non-adversarial non-consistent $n_\sigma = 5$

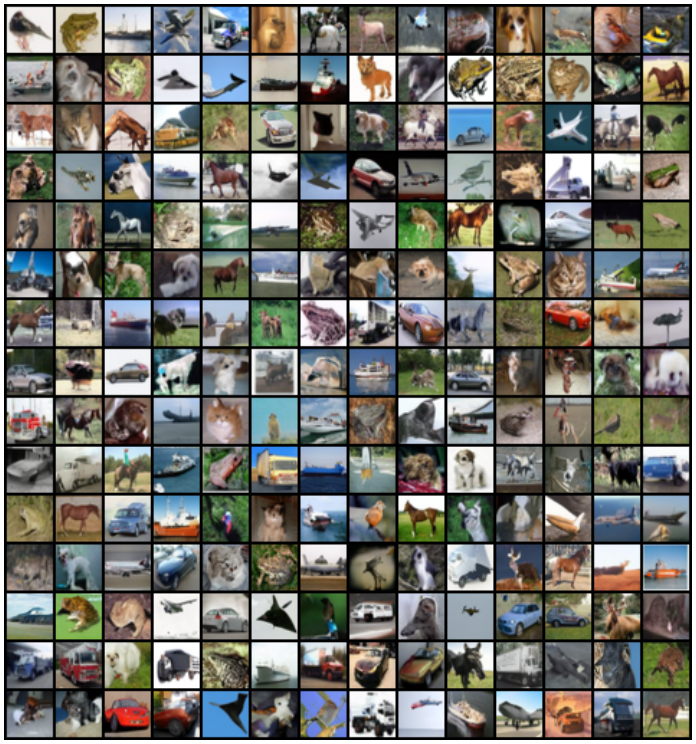


Figure 23: Ho et al. (2020) network architecture with CIFAR-10 Adversarial non-consistent $n_\sigma = 5$

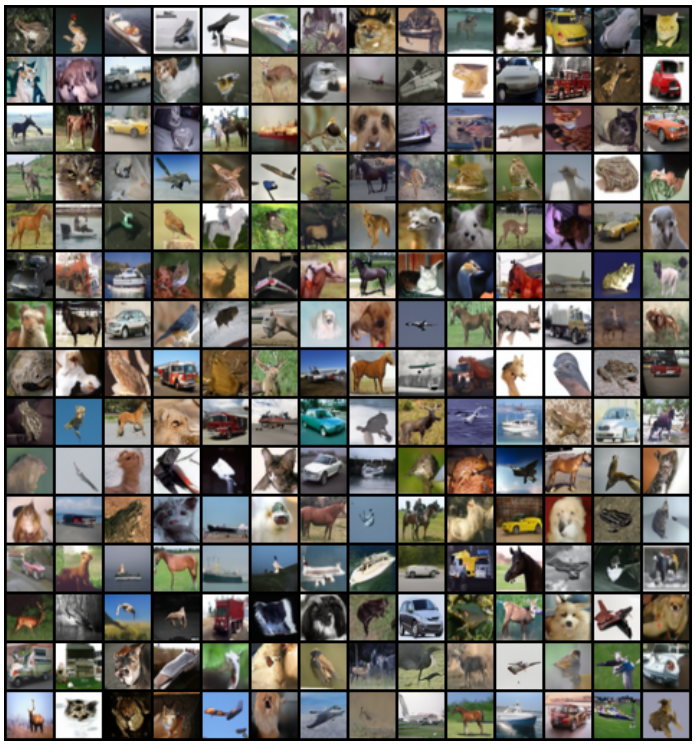


Figure 24: Ho et al. (2020) network architecture with CIFAR-10 Non-adversarial consistent $n_\sigma = 1$

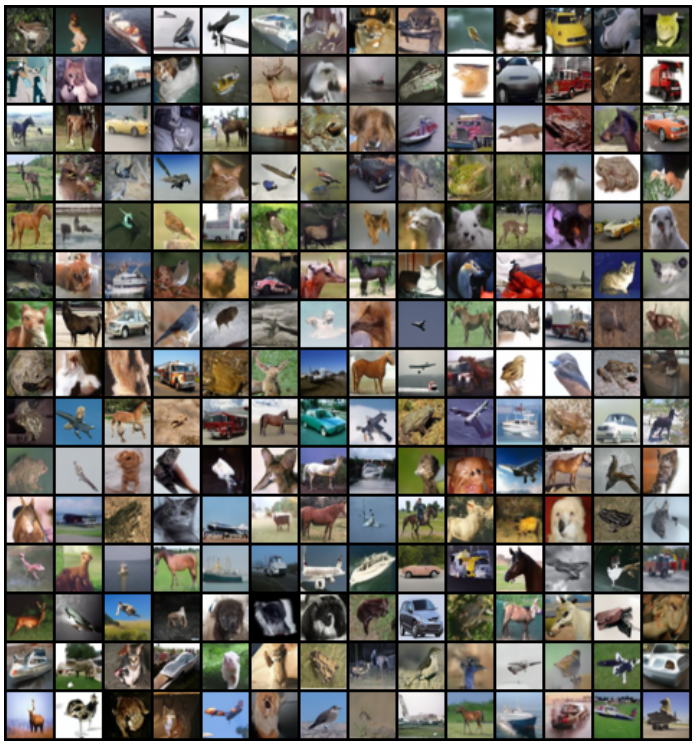


Figure 25: Ho et al. (2020) network architecture with CIFAR-10 Adversarial consistent $n_\sigma = 1$

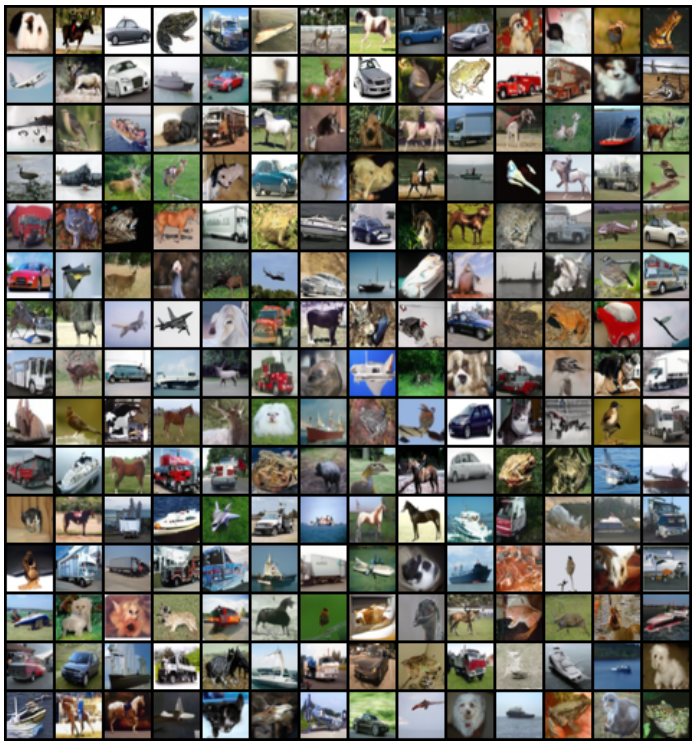


Figure 26: Ho et al. (2020) network architecture with CIFAR-10 Non-adversarial consistent $n_\sigma = 5$

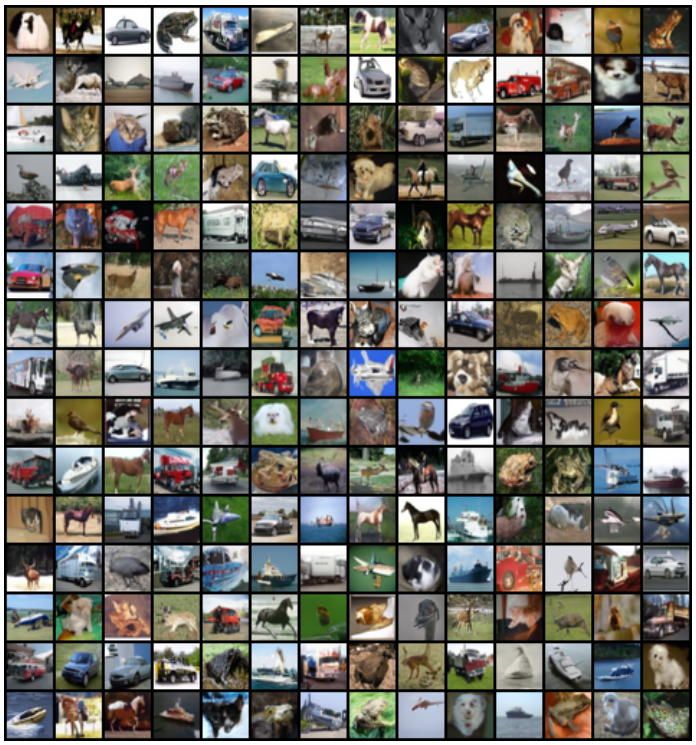


Figure 27: Ho et al. (2020) network architecture with CIFAR-10 Adversarial consistent $n_\sigma = 5$