Nested Diffusion Processes for Anytime Image Generation

Noam Elata [1] **Bahjat Kawar** [2] **Tomer Michaeli** [1] **Michael Elad** [2]

Abstract

Diffusion models are the current state-of-the-art in image generation, synthesizing high-quality images by breaking down the generation process into many fine-grained denoising steps. Despite their good performance, diffusion models are computationally expensive, requiring many neural function evaluations (NFEs). In this work, we propose an anytime diffusion-based method that can generate viable images when stopped at arbitrary times before completion. Using existing pretrained diffusion models, we show that the generation scheme can be recomposed as two nested diffusion processes, enabling fast iterative refinement of a generated image. We use this Nested Diffusion approach to peek into the generation process and enable flexible scheduling based on the instantaneous preference of the user. In experiments on ImageNet and Stable Diffusion-based text-to-image generation, we show, both qualitatively and quantitatively, that our method's intermediate generation quality greatly exceeds that of the original diffusion model, while the final slow generation result remains comparable.[1]

1. Introduction

The sampling process of modern diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019) can be computationally expensive (Salimans & Ho, 2022; Song et al., 2020; Lu et al., 2022), due to the large networks used and the iterative nature of the reverse diffusion process. During sampling, it is possible to monitor the diffusion models by examining the intermediate predictions, denoted as $\hat{\mathbf{x}}_0$, at various time steps. However, these predictions do not align with the learned image manifold and often exhibit a smooth or blurry appearance (Kawar et al., 2021). To address this issue, we propose Nested Diffusion, a novel technique that leverages a pretrained diffusion model to iteratively refine generated images, acting as an *anytime* generation algorithm. With Nested Diffusion, intermediate predictions are of better quality, and users have the ability to observe the generated image during the sampling process and can choose to terminate the generation if the intermediate result is satisfactory. Furthermore, in the setting where multiple images are generated concurrently, the user has the option to select the leading candidate and guide the sampling process towards the preferred image.

[1]Department of Electrical and Computer Engineering, Technion, Israel [2]Department of Computer Science, Technion, Israel. Correspondence to: Noam Elata <noamelata@campus.technion.ac.il>.

Accepted to ICML workshop on Structured Probabilistic Inference & Generative Modeling 2023, Honolulu, Hawaii, USA.

[1]Our code is available at https://github.com/noamelata/NestedDiffusion.

2. Preliminaries: Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019) are the state-of-the-art generative models (Dhariwal & Nichol, 2021), relying on the capabilities of deep neural networks (DNN) in removing Gaussian noise. The forward diffusion process is defined as a degradation of a data point $\mathbf{x}_0$ in a dataset $\mathcal{D}$ with accumulating Gaussian noise using a series of noise amplitudes β_t and $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$ for all timesteps $t = 1, \dots, T$. During training, the reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is learned by minimizing the evidence lower bound (ELBO) on the training dataset. The ELBO can be written as a sum of Kullback Leibler divergence terms between $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, which have a simple closed-form target when $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is modeled as a Gaussian distribution. The trained DNN gradually removes noise from a random initialization $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, sampling iteratively from the learned distributions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, and finally outputting a generated image $\mathbf{x}_0 \sim q(\mathbf{x}_0)$.

3. Nested Diffusion

3.1. Formulation

In DDPM (Ho et al., 2020), $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is assumed to follow a Gaussian distribution, with its mean defined using the expectation $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ yielded by the DNN, and its variance defined as a constant. Thus, we can sample from $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ in closed-form. However, we can reinterpret this sampling by marginalizing the distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

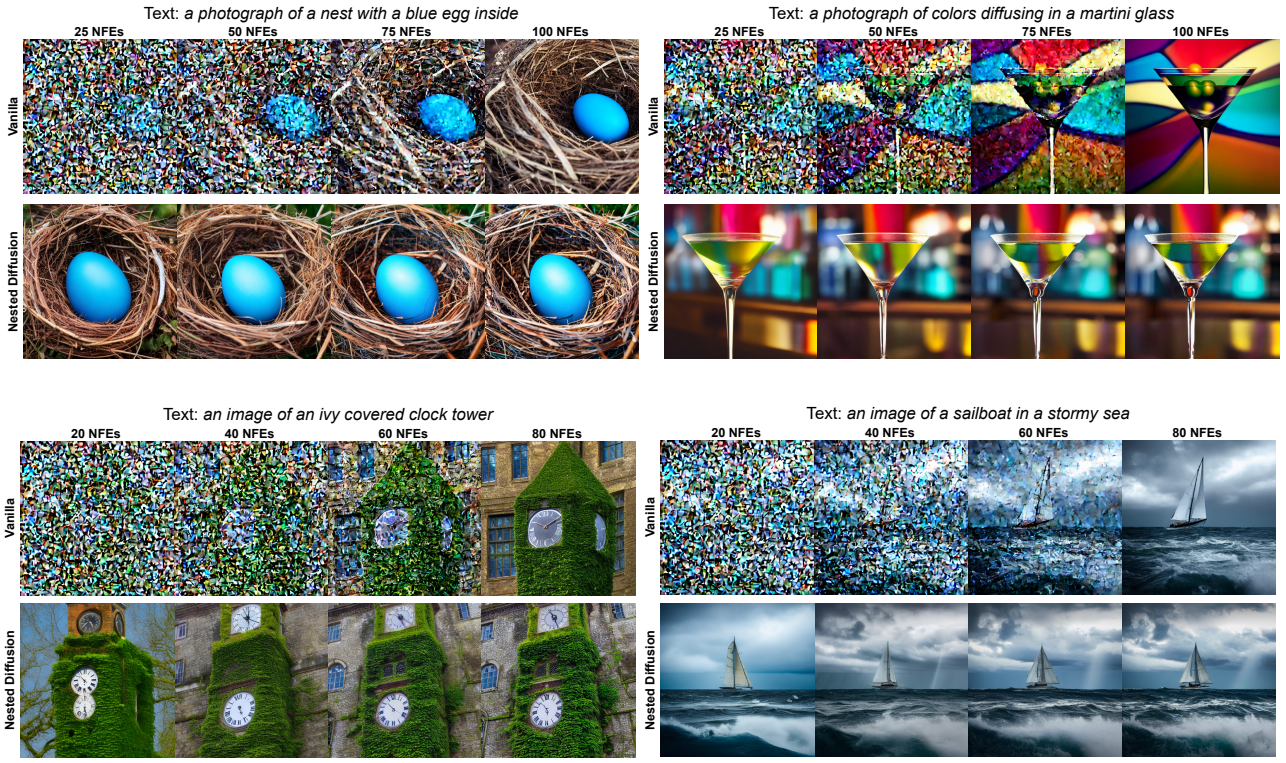

Figure 1. Results of intermediate predictions of Stable Diffusion from a reverse diffusion process of 100 steps (top) and 80 steps (bottom).

as a convolution of two others (Xiao et al., 2022) – the closed-form distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)$, and a DNN-based approximation $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$, as follows:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \int q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t) p_\theta(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0. \quad (1)$$

In DDPM, $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ would correspond to a Dirac delta function around the DNN-estimated $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$, and $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)$ would be a fixed Gaussian. More generally, sampling from the joint distribution $p_\theta(\mathbf{x}_{t-1}, \mathbf{x}_0|\mathbf{x}_t) = q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t) p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ can be done sequentially, by first sampling $\hat{\mathbf{x}}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ and then sampling $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\hat{\mathbf{x}}_0, \mathbf{x}_t)$, resulting in $\mathbf{x}_{t-1}$ that follows Equation 1. The generalized reverse diffusion process, following this interpretation, is presented in Algorithm 1.

Algorithm 1 Sampling from Reverse Diffusion Process

$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
for t in $\{\mathrm{T} \dots 1\}$ **do**
  $\hat{\mathbf{x}}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$
  $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\hat{\mathbf{x}}_0, \mathbf{x}_t)$
end for
return $\mathbf{x}_0$

Note that after training $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ for a certain $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)$, it is possible to utilize the same DNN model for different distributions q. For instance, DDIM (Song et al., 2020) utilizes a deterministic $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)$ (equivalent to a Dirac delta function) for faster generation. Interestingly, by sampling using Algorithm 1, the Gaussian assumption on $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is no longer required, and can be generalized beyond DDPM sampling. In this setting, $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ may be any learned distribution, and is not restricted to a delta function or a Gaussian form.

3.2. Method

We suggest that many valid choices of $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)$ and an accurate DNN-based approximation $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ can generate high quality samples using Equation 1 and Algorithm 1. This could allow us to harness many different generative models into the diffusion process, for instance as done with GANs (Goodfellow et al., 2014) by Xiao et al. (2022). In this section, we suggest that even complete diffusion processes may be used as a good approximation for $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

We propose a Nested Diffusion process, where an **outer diffusion** process would utilize the generative sampler $p_\psi(\mathbf{x}_0|\mathbf{x}_t)$ – itself an **inner diffusion** process. As shown in Algorithm 2, for each sampling step in the outer diffusion, the inner diffusion would use an unaltered (vanilla) diffusion model to generate a plausible image $\hat{\mathbf{x}}_0$, which would

Algorithm 2 Sampling from Nested Diffusion

Outer diffusion denoted in blue
Inner diffusion denoted in purple
$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
for t in $\{T \dots 1\}$ **do**
  $\mathbf{x}'_{T'} = \mathbf{x}_t$
  for t' in $\{T' \dots 1'\}$ **do**
    $\hat{\mathbf{x}}'_{0'} \sim p_\theta(\mathbf{x}'_{0'}|\mathbf{x}'_{t'})$
    $\mathbf{x}'_{t'-1} \sim q'(\mathbf{x}'_{t'-1}|\hat{\mathbf{x}}'_{0'}, \mathbf{x}'_{t'})$
  end for
  $\hat{\mathbf{x}}_0 = \mathbf{x}'_{0'}$
  $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\hat{\mathbf{x}}_0, \mathbf{x}_t)$
end for
return $\mathbf{x}_0$

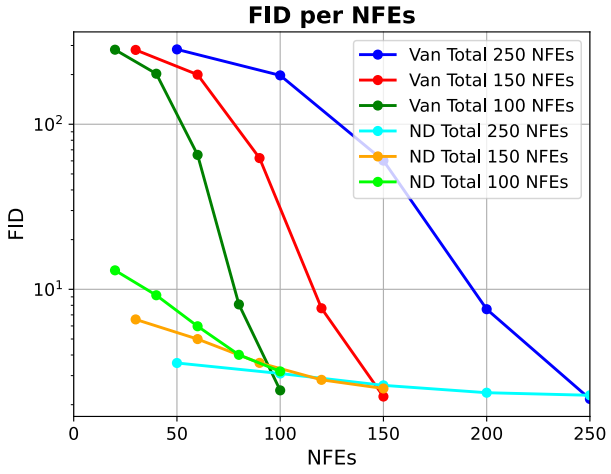

Figure 2. 50K FID evaluation of intermediate predictions from Nested (ND) and vanilla (Van) diffusion processes.

than be used to calculate $\mathbf{x}_{t-1}$ in the outer diffusion. We emphasize that only the inner diffusion uses a DNN. The inner diffusion becomes the outer diffusion's abstraction for a generative model.

Unlike vanilla diffusion processes, Nested Diffusion yields a more detailed $\hat{\mathbf{x}}_0$ at the termination of each outer step. This is because $\hat{\mathbf{x}}_0$ is a sample generated from the multi-step inner diffusion process, and not the mean yielded by a single denoising step. These $\hat{\mathbf{x}}_0$ estimations hint at the final algorithm result while being closer to the manifold. Using Nested Diffusion, the sampling process becomes an *anytime* algorithm, in which a valid image may be returned if the algorithm is terminated prematurely.

Nested Diffusion requires |outer steps| $\times$ |inner steps| NFEs for a complete image genration process. For a given number of NFEs, Nested Diffusion may support any ratio $R_{ND} = \frac{|\text{outer steps}|}{|\text{inner steps}|}$. This ratio represents a tradeoff between fast updates to the predicted image, and the intermediate image quality (see Appendix A). Additionally, the ratio influences the number of NFEs needed before Nested Diffusion produces its initial intermediate prediction, which occurs at the conclusion of the first inner process. In the extremes, where the number of either outer steps or inner steps is one, the process reverts to vanilla diffusion sampling.

The computation devoted to each outer step is not required to be the same – i.e. different ratio per outer step. As the number of inner steps corresponds to the number of NFEs, changing the length of each outer steps determines the amount of computations devoted to this step. In our experiments, we use the same number of inner steps for each outer step for simplicity. We hope that future work could fine-tune the inner step allocation for each outer step and achieve better results.

4. Experiments

We evaluate Nested Diffusion as an anytime image generator using a DiT model (Peebles & Xie, 2022) trained on 256×256-pixel ImageNet (Deng et al., 2009) images, as well as Stable Diffusion (Rombach et al., 2022) V1.5. To ensure a fair comparison, we compare Nested Diffusion against the unaltered sampling algorithm (vanilla) using the same models, hyperparameters, and total number of NFEs used.

All experiments use deterministic DDIM (Song et al., 2020) sampling for the outer diffusion. The inner diffusion hyperparameters are chosen according to the best practices of each model used.

4.1. Class-Conditional ImageNet Generation

The denoising DNN from DiT (Peebles & Xie, 2022) uses a VAE (Kingma & Welling, 2013) based architecture to decode generated latent samples (Vahdat et al., 2021; Rombach et al., 2022), thus enabling the appliciation of the diffusion model in a smaller latent space. The DNN is trained using Kullback Leibler divergence to yield both the mean and variance of a Gaussian distribution $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$[2]. In addition, the DNN has been trained with class-labels, using Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) to generate class-conditional samples. When using this DNN for Nested Diffusion, both the inner diffusion and the outer are conducted in the latent space. The variance prediction is used only in the inner diffusion, while the outer diffusion

[2]The model directly predicts the conditional mean of the Gaussian noise in $\mathbf{x}_t$ and the variance of $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, but we can use a change of variables to view these as the mean and variance of $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$, conforming with our notation.

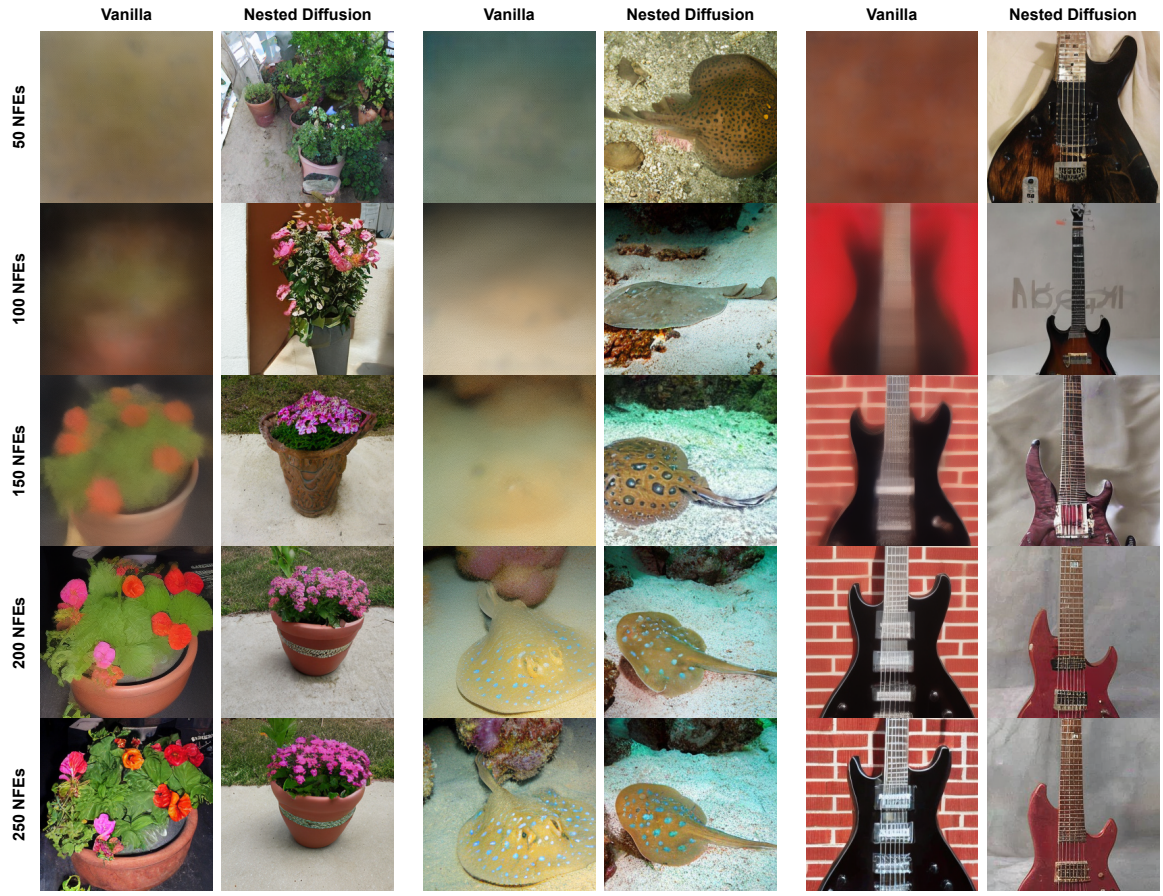

Figure 3. Samples of ImageNet generation, comparing vanilla diffusion model against Nested Diffusion.

remains deterministic DDIM sampling. CFG is regarded as part of the DNN, and therefore applied in the inner diffusion only. We set the CFG value to 1.5 similar to Peebles & Xie (2022). In Figure 3 we compare samples generated using 250 vanilla diffusion steps against Nested diffusion with 5 outer steps with 50 inner steps each (totaling 250 NFEs). The latents from the intermediate steps are decoded using the VAE decoder.

In Figure 2 we compare the FID (Heusel et al., 2017) of intermediate estimations of Nested Diffusion with the intermediate estimations of vanilla diffusion models, for the same number of NFEs[3]. We note that the intermediate FID scores for Nested Diffusion are much better than their vanilla counterparts, while the final result FID (without premature interruption) of Nested Diffusion is comparable to the vanilla diffusion. Exact FID values can be found in Table 1 in Appendix B.

[3]FID for vanilla diffusion DiT reflect results reproduced by us, which are slightly better than reported in the original paper (Peebles & Xie, 2022).

4.2. Text-to-Image Generation

Stable Diffusion is a large text-to-image model capable of generating photo-realistic images for any textual prompt (Rombach et al., 2022). We use Stable Diffusion to test Nested Diffusion for text-to-image generation. Similar to subsection 4.1, Stable Diffusion's process runs in a latent space, and uses CFG for text-conditional sampling. We implement Nested Diffusion using non-deterministic DDIM with $\eta = 0.85$ for the inner diffusion, and treat the CFG as we did in subsection 4.1, setting it to the default value of 7.5. In Figure 1, we present intermediate results from Nested Diffusion and compare them to their counter parts from vanilla Stable Diffusion, decoding intermediate latents using the VAE decoder. The Nested Diffusion sampling process previews satisfactory outputs, highly similar to the end result. The finer details in the images improve with the accumulation of more NFEs. Based on the figure, it is apparent that the intermediate latents obtained from the vanilla diffusion model do not correspond to valid latents. As a result, when these latents are decoded, they produce "fracture patterns" instead of natural-looking images. More examples for generated images can be found in Appendix B.

References

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

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

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations (ICLR)*, 2022.

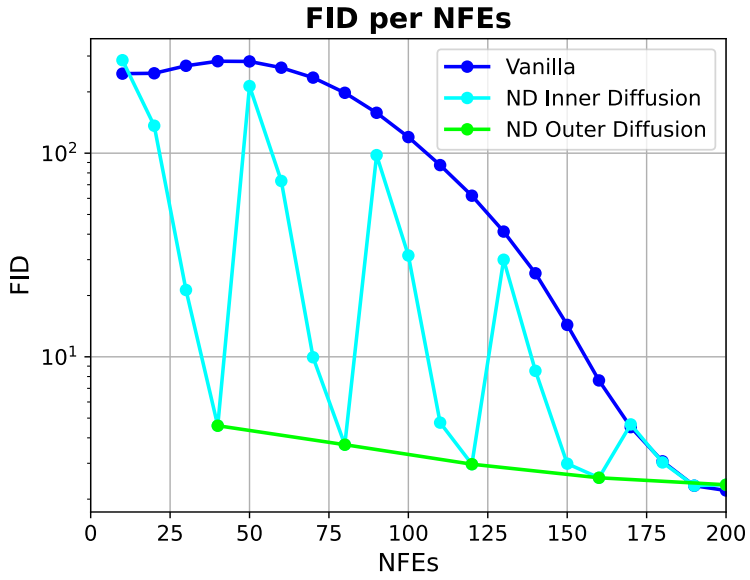

Figure 4. 50K FID evaluation of intermediate predictions of Nested Diffusion's inner and outer diffusion process, compared against a vanilla diffusion process. FID is measured every 10 NFEs for Nested Diffusion's inner diffusion process and for the vanilla one, whereas the Nested Diffusion outer process's FID scores correspond to every fourth inner diffusion measurement, *i.e.*, every 40 NFEs.

Table 1. 50K FID evaluation of Nested (ND) and vanilla (Van) diffusion processes when stopped at different percentages of the full algorithm runtime (100, 150, 250 NFEs).

	TOTAL 100 NFES			TOTAL 150 NFES			TOTAL 250 NFES		
%	NFES	VAN	ND	NFES	VAN	ND	NFES	VAN	ND
20%	20	282.89	13.03	30	282.05	6.57	50	284.13	3.57
40%	40	202.34	9.20	60	199.74	4.99	100	197.74	3.08
60%	60	65.22	5.97	90	62.37	3.58	150	60.19	2.61
80%	80	8.10	4.00	120	7.67	2.82	200	7.57	2.36
100%	100	2.44	3.18	150	2.24	2.50	250	2.16	2.28

A. Outer Steps – Inner Steps Trade-off

In Figure 4 we visualize the sample quality trend for intermediate **inner samples** $\hat{\mathbf{x}}'_{0'}$ using FID. The graph shows five distinct drops, corresponding to the five outer diffusion steps. Within each outer step, the inner diffusion's intermediate prediction's quality improves quickly until yielding its final $\mathbf{x}'_{0'}$, which (as shown in Algorithm 2) is also the outer diffusion's intermediate prediction $\hat{\mathbf{x}}_0$. Nested Diffusion would return the last $\hat{\mathbf{x}}_0$ computed if terminated prematurely – corresponding to the local minimas in the graph, shown in green.

The ratio $R_{ND} = \frac{|\text{outer steps}|}{|\text{inner steps}|}$ determines the NFEs required for each update of the Nested Diffusion intermediate prediction. Faster update rates come at a cost of lower intermediate prediction samples quality. Figure 5 shows this trade-off, showing Nested Diffusion sampling with different R_{ND} where all other hyperparameters as well as the random seed remain equal.

B. More Examples

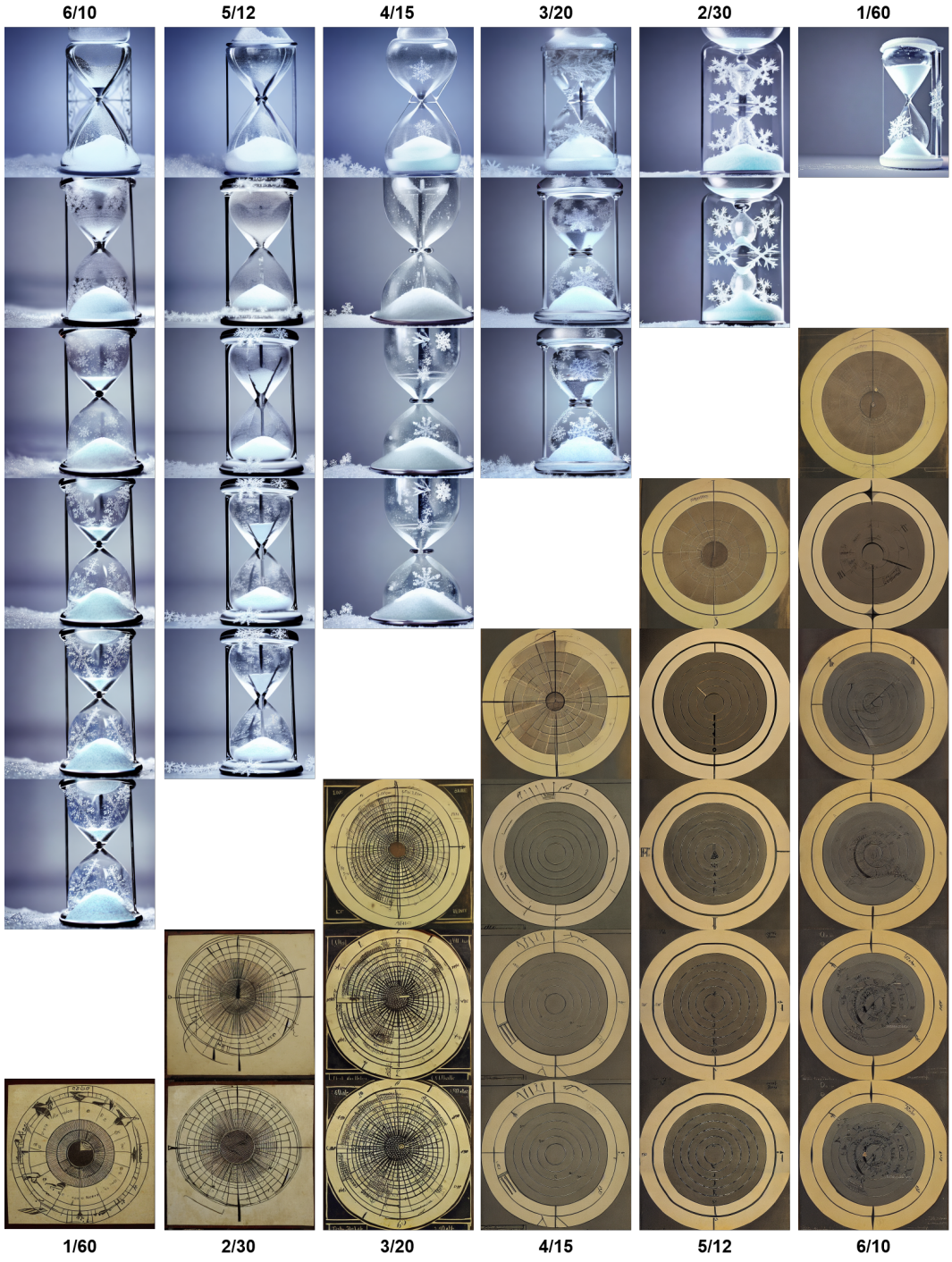

Figure 5. Qualitative examples of Nested Diffusion with different ratios R_{ND}, each column denoted with |outer steps|/|inner steps| at the top or bottom. Top text: *a photograph of an hourglass filled with snowflakes*. Bottom text: *a diagram of an ancient sundial*. Diffusion process progresses from top to bottom.

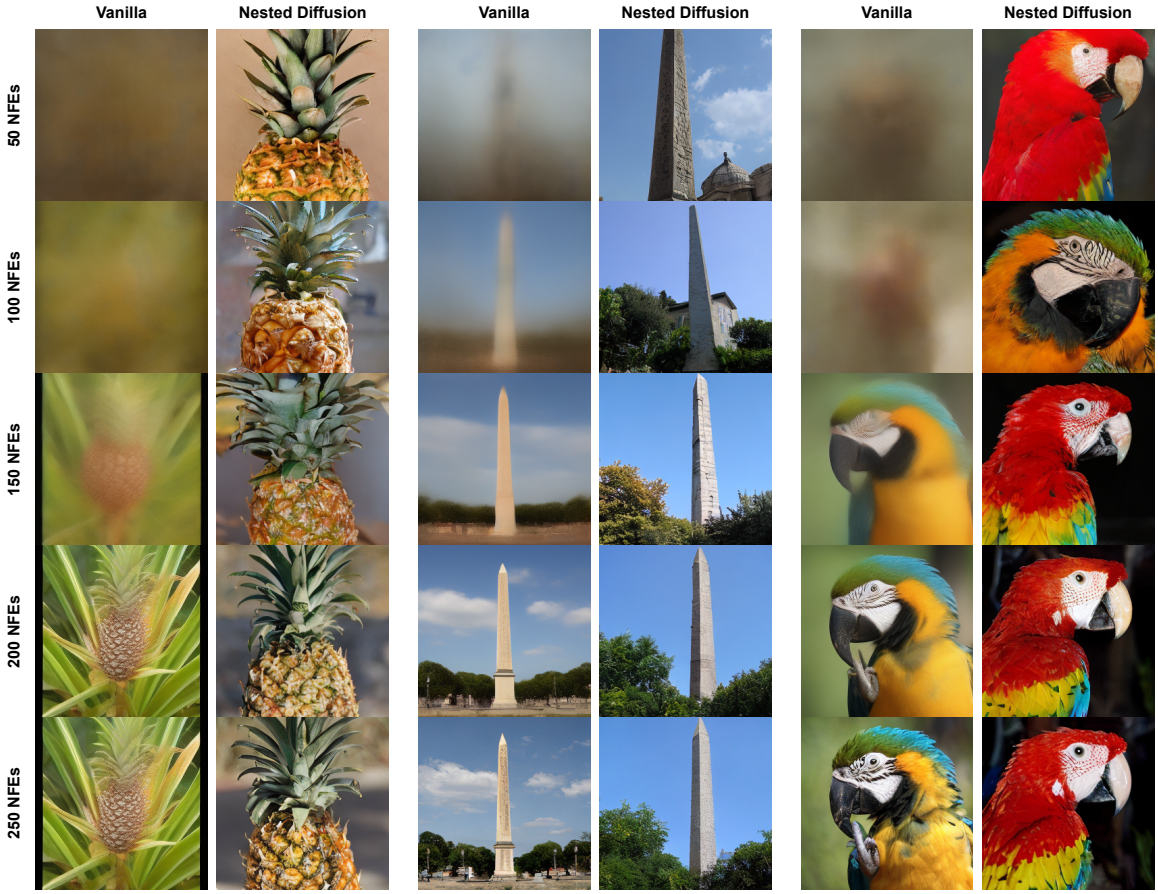

Figure 6. Additional samples of ImageNet generation, comparing vanilla diffusion model against Nested Diffusion.

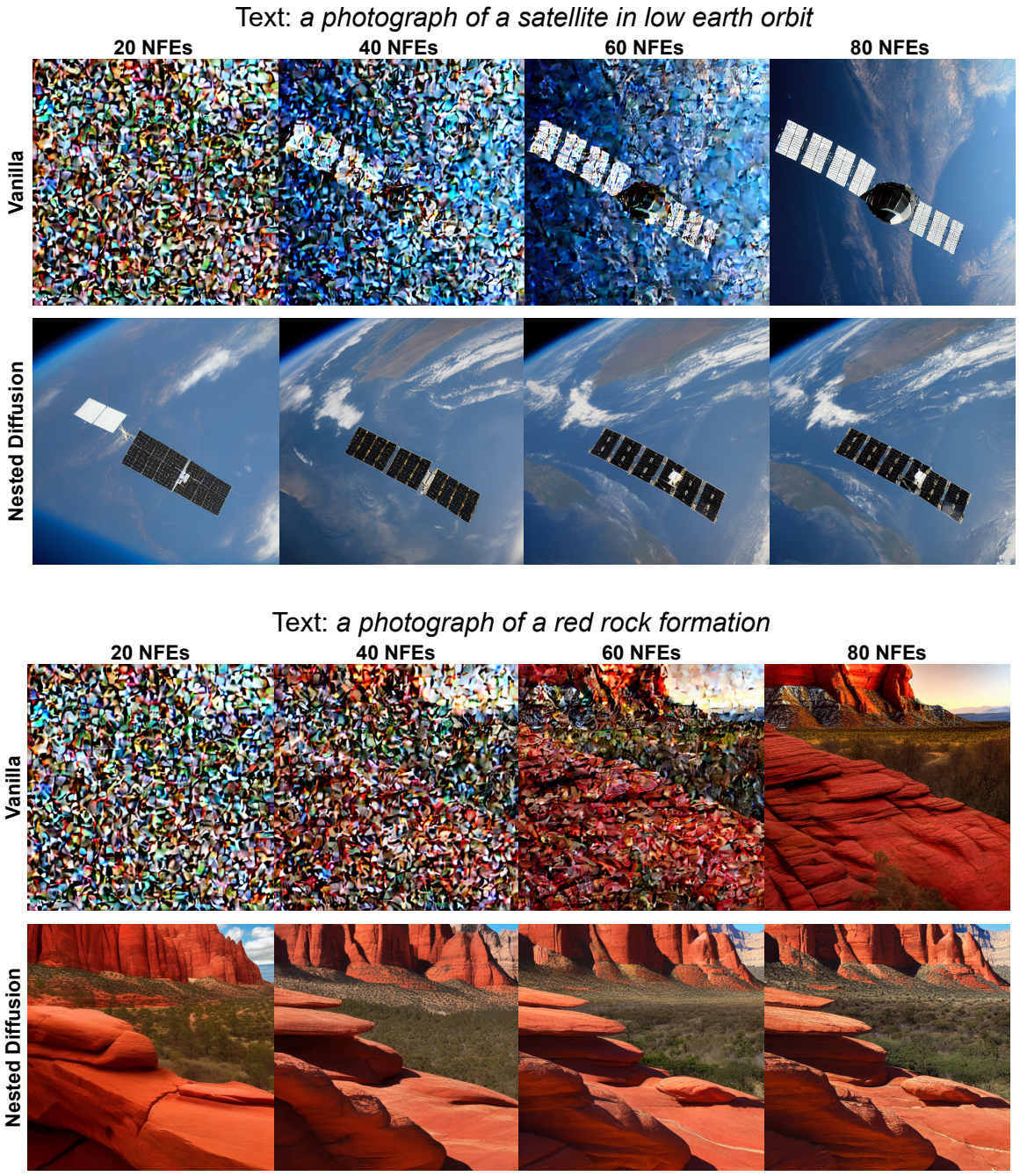

Figure 7. More Results of intermediate predictions of Stable Diffusion from a reverse diffusion process with 80 steps.