Nested Diffusion Processes for Anytime Image Generation

Noam Elata[1] Bahjat Kawar[2] Tomer Michaeli[1] Michael Elad[2]
[1]Department of ECE [2]Department of CS
Technion – Israel Institute of Technology
{noamelata@campus, bahjat.kawar@cs, tomer.m@ee, elad@cs}.technion.ac.il

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. 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 generation result remains comparable. We illustrate the applicability of Nested Diffusion in several settings, including for solving inverse problems, and for rapid text-based content creation by allowing user intervention throughout the sampling process. [1]

1. Introduction

Diffusion models (DMs) have emerged as a promising class of generative models [17, 47, 51]. Having achieved state-of-the-art capabilities in image generation, DMs have also excelled at various tasks, such as inverse problem solving [6, 22, 24, 34, 49, 54], and image editing [4, 14, 25, 35]. DMs have also been successfully applied to a wide range of domains, ranging from speech and audio [20, 27] to protein structures [39, 45] and medical data [7, 23, 52].

The sampling process of modern DMs can be computationally expensive [33, 44, 48], due to the large networks used and the iterative nature of the reverse diffusion process. Despite the progress in acceleration of DMs [44, 48, 50], many of the leading diffusion model-based applications remain prohibitively slow.

During sampling, the diffusion process creates intermediate image predictions as a byproduct, denoted as $\hat{\mathbf{x}}_0$, at various time steps. In theory, this allows for monitoring the generation process and assessing the resulting images without waiting for its completion. However, these predictions do not align with the learned image manifold and often exhibit a smooth or blurry appearance [24].

To address this issue, we propose Nested Diffusion, a novel technique that leverages a pretrained DM to iteratively refine generated images, acting as an *anytime* generation algorithm. With Nested Diffusion, intermediate predictions of the final generated image are of better quality, which allows users to observe the generated image during the sampling process and to conclude the generation if the intermediate yielded image is already satisfactory. Through experiments, we observe that our Nested Diffusion shows superior intermediate generation quality compared to the original DM, while maintaining comparable final results.

Access to high quality intermediate predictions is advantageous in several DM-based applications beyond image generation. Nested Diffusion's anytime algorithm is also beneficial in solving inverse problems, given the prominence of DMs in this area. In scenarios where multiple output images are generated, users can control the generation process by selecting the most promising candidate and influencing the sampling process to prioritize their preferred images. This valuable capability also enables interactive online adjustments during the generation process.

We introduce Nested Diffusion, a novel anytime sampling algorithm for DMs. We showcase the versatility of our method through various applications, including conditional image generation, and inverse problem solving. Additionally, we highlight the ability of Nested Diffusion to enable interactive content creation during the sampling process.

2. Preliminaries

2.1. Anytime Algorithms

Anytime algorithms [3, 13, 19, 58] are a class of methods that attempt to address real-world problems under re-

[1]Code available at https://github.com/noamelata/NestedDiffusion.

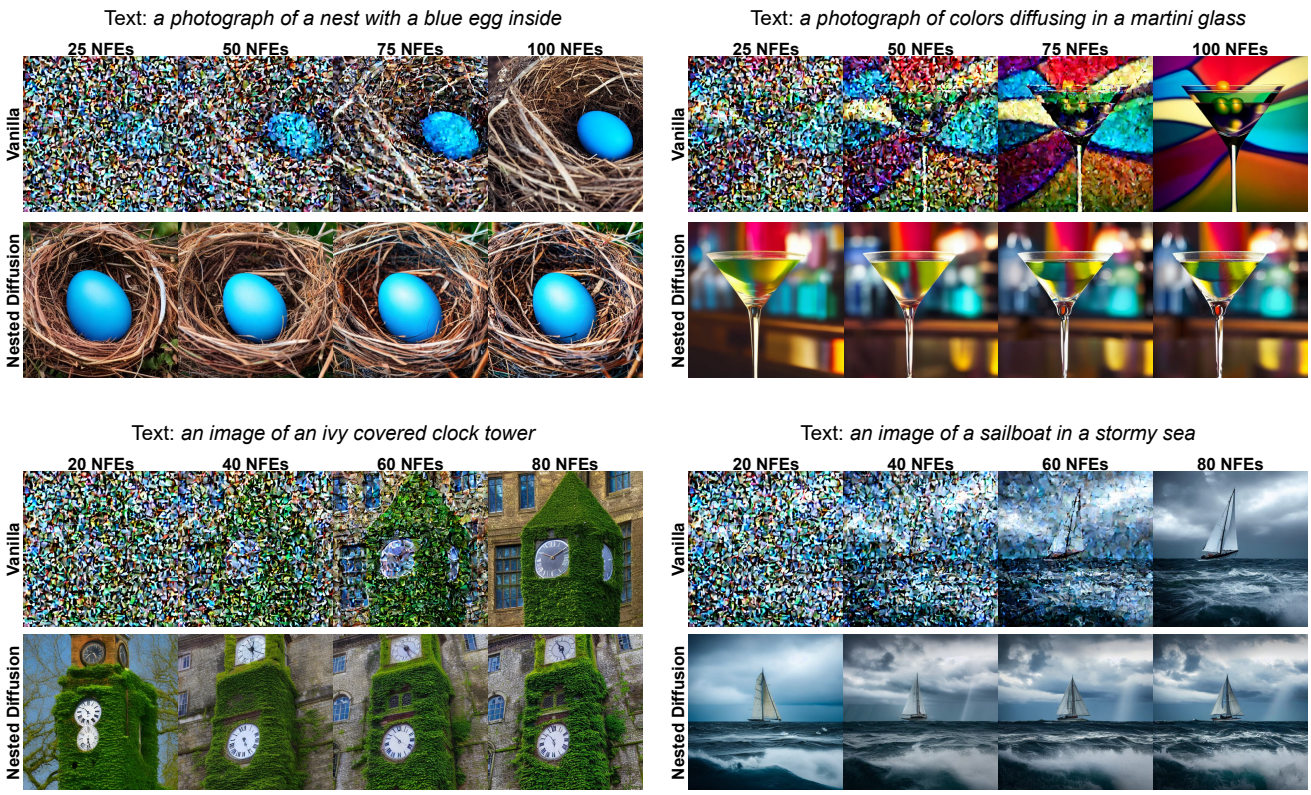


Figure 1. **Results of intermediate predictions of Stable Diffusion from a diffusion process of 100 NFEs (top) and 80 NFEs (bottom).**

source constraints, time limitations, or uncertain input information. Specifically, these algorithms generate progressively improved solutions, which enable early interruption. Unlike conventional algorithms that require completion for a final solution, anytime algorithms offer users the flexibility to obtain usable solutions at any stage of execution. This adaptability proves highly valuable in time-sensitive decision-making and iterative problem-solving scenarios.

2.2. Diffusion Models

Diffusion models (DMs) [17, 47, 51] are the state-of-the-art generative models [10], 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 accumulated Gaussian noise through the process $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$ defined over timesteps $t = 1, \ldots, T$, where β_t are the noise amplitudes. During training, the reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is learned by maximizing the evidence lower bound (ELBO) on the training dataset. The ELBO can be written in terms 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 expression when $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is modeled as a Gaussian distribution. At inference time, 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 from a distributions approximating the real image distribution, $\mathbf{x}_0 \sim q(\mathbf{x}_0)$.

3. Nested Diffusion

3.1. Formulation

In DDPM [17], $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 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 also interpret this sampling process by expressing the distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ as a convolution of two others [55] – the distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)$ which has a closed form, and the DNN-based approximation $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$,

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

Here, $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ is the distribution represented by the DNN in the context of predicting $\mathbf{x}_0$. For instance, DMs with a deterministic DNN output, such as DDPM, would correspond to a Dirac delta function distribution around the DNN-estimated $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$. The stochasticity in the reverse diffusion process would be obtained by setting $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)$ as a fixed Gaussian.

Algorithm 1 Sampling from a Regular Diffusion Process

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

Algorithm 2 Sampling from Nested Diffusion

Outer diffusion denoted in blue, with step size s^o
Inner diffusion denoted in purple, with step sizes $\{s^i_t\}$
$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
for t in $\{T, T-s^o, \dots, 1+s^o, 1\}$ **do**
  $\mathbf{x}'_t = \mathbf{x}_t$ ▷ Beginning of inner diffusion
  for τ in $\{t, t-s^i_t, \dots, 1+s^i_t, 1\}$ **do**
    $\hat{\mathbf{x}}'_0 \sim p_\theta(\mathbf{x}'_0|\mathbf{x}'_\tau)$
    $\mathbf{x}'_{\tau-s^i_t} \sim q'(\mathbf{x}'_{\tau-s^i_t}|\hat{\mathbf{x}}'_0, \mathbf{x}'_\tau)$
  end for
  $\hat{\mathbf{x}}_0 = \mathbf{x}'_0$ ▷ End of inner diffusion
  $\mathbf{x}_{t-s^o} \sim q(\mathbf{x}_{t-s^o}|\hat{\mathbf{x}}_0, \mathbf{x}_t)$
end for
return $\mathbf{x}_0$

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 $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\hat{\mathbf{x}}_0, \mathbf{x}_t)$, yielding $\mathbf{x}_{t-1}$ that follows Equation (1). The generalized reverse diffusion process, following this interpretation, is presented in Algorithm 1.

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 [48] 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 [12] by Xiao et al. (2022) [55]. Here, we we propose to use a complete diffusion process 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. For simplicity of notation, we denote the outer diffusion variables and distributions in blue and the inner diffusion variables and distributions in purple. As shown in Algorithm 2 and Figure 2, for each sampling step t in the outer diffusion, the inner diffusion uses an unaltered (vanilla) DM to generate a plausible image $\hat{\mathbf{x}}_0$, which would then be used to calculate $\mathbf{x}_{t-s^o}$ 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.

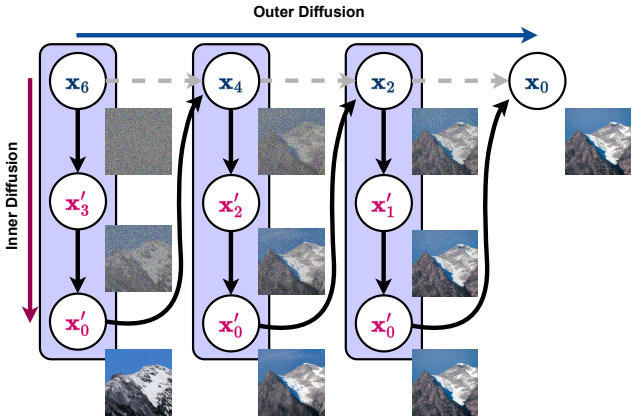
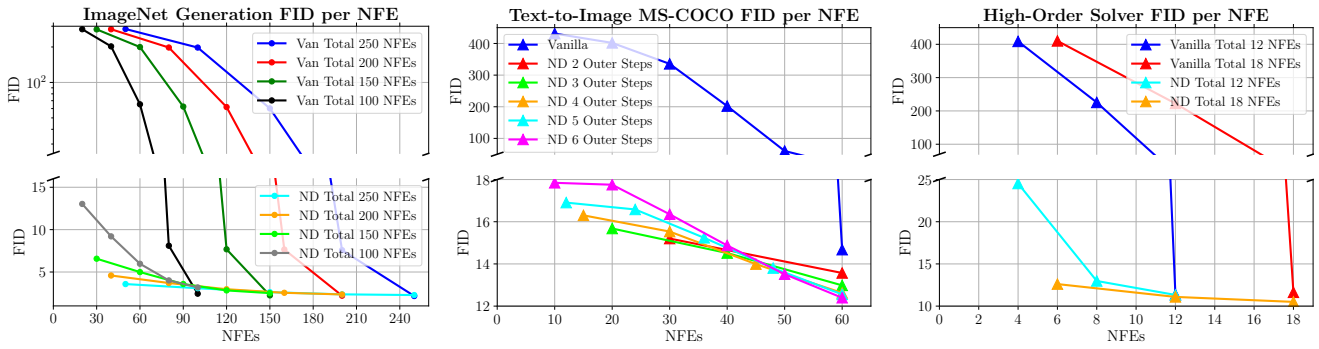


Figure 2. **Schematic description of Nested Diffusion.** The outer diffusion process is depicted using the dotted gray arrows

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 image 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 $|\text{outer steps}| \times |\text{inner steps}|$ 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 supplementary material). 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. In the supplementary material, we propose a metric for comparing between Nested Diffusion with different R_{ND}. However, we suggest tuning the R_{ND} parameter according to the specific application and hardware, aiming to provide users with a waiting time between image updates



(a) **50K FID for class-conditional ImageNet generation.** Evaluation of intermediate predictions from Nested (ND) and vanilla (Van) diffusion processes.

(b) **MS-COCO 30K FID of text-to-image generation.** Intermediate predictions are generated with a total of 60 NFEs, and different outer step configurations. Vanilla diffusion is equivalent to using Nested Diffusion with one outer step – in blue.

(c) **MS-COCO 30K FID of text-to-image generation using DPM-Solver++ for the inner diffusion.** Using a high-order solver enables high-quality generation with fewer NFEs.

Figure 3. **FID as a function of NFE for ImageNet, text-to-image, and high-order solver text-to-image generation.**

ranging from one to ten seconds – in Nested Diffusion, the waiting time is the duration of an inner diffusion process.

The computation devoted to each outer step is not required to be the same, i.e. we can have a different ratio per outer step. As the number of inner steps corresponds to the number of NFEs, changing the length of each outer step determines the computation devoted to this step. In our experiments, we use the same number of inner steps for each outer step for simplicity. We leave for future work to explore dynamic allocation of the number of inner steps per outer step.

4. Experiments

We evaluate Nested Diffusion as an anytime image generator using a DiT model [37] trained on 256×256-pixel ImageNet [8] images and on Stable Diffusion [41] V1.5. We also show that Nested Diffusion can incorporate an inverse problem solver, and present several examples on CelebA-HQ256 [21] using DDRM [22]. To ensure a fair comparison, we compare Nested Diffusion against the unaltered sampling algorithm (vanilla) using the same DNN models, hyperparameters, and total number of NFEs used. The sampling speed of Nested Diffusion is also equal to that of vanilla diffusion, as sampling time is directly proportional to the total number of NFEs used for generation. All experiments use DDIM [48] sampling for the outer diffusion. The inner diffusion hyperparameters are chosen according to the best practices of the model selected for the experiment.

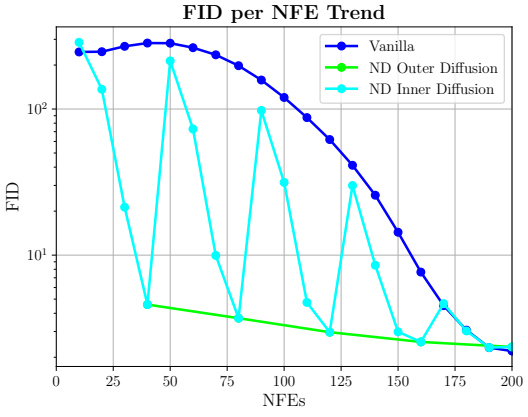


Figure 4. **50K FID evaluation of Nested Diffusion's inner and outer diffusion processes.** FID is measured on intermediate predictions of class-conditional ImageNet generation, compared to a vanilla diffusion process, every 10 NFEs. The Nested Diffusion outer process's FID scores correspond to every fourth inner diffusion measurement, *i.e.*, every 40 NFEs.

4.1. Class-Conditional ImageNet Generation

The denoising DNN employed in DiT [37] uses a VAE [26] based architecture to decode generated latent samples [41, 53], thus enabling the application of the DMs in a more efficient latent space. The DNN yields both the mean and variance of a Gaussian distribution $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$, enabling sampling using the reparameterization trick [26]. In addition, the DNN has been trained with class-labels, using Classifier-Free Guidance (CFG) [18] to generate class-conditional samples. Figure 5 shows samples generated using 250 vanilla diffusion steps compared against Nested diffusion with 5 outer steps and 50 inner steps each (totaling 250 NFEs). The latents from the intermediate steps are decoded using the VAE decoder.

In Figure 3a we compare the FID [16] of intermediate estimations of Nested Diffusion with the intermediate estimations of vanilla DMs, for the same number of NFEs[2]. We note that the intermediate FID scores for Nested Diffusion are much better than their vanilla counterparts, while

[2]FID for vanilla diffusion DiT reflect results reproduced by us, which are slightly better than reported in the original paper [37].

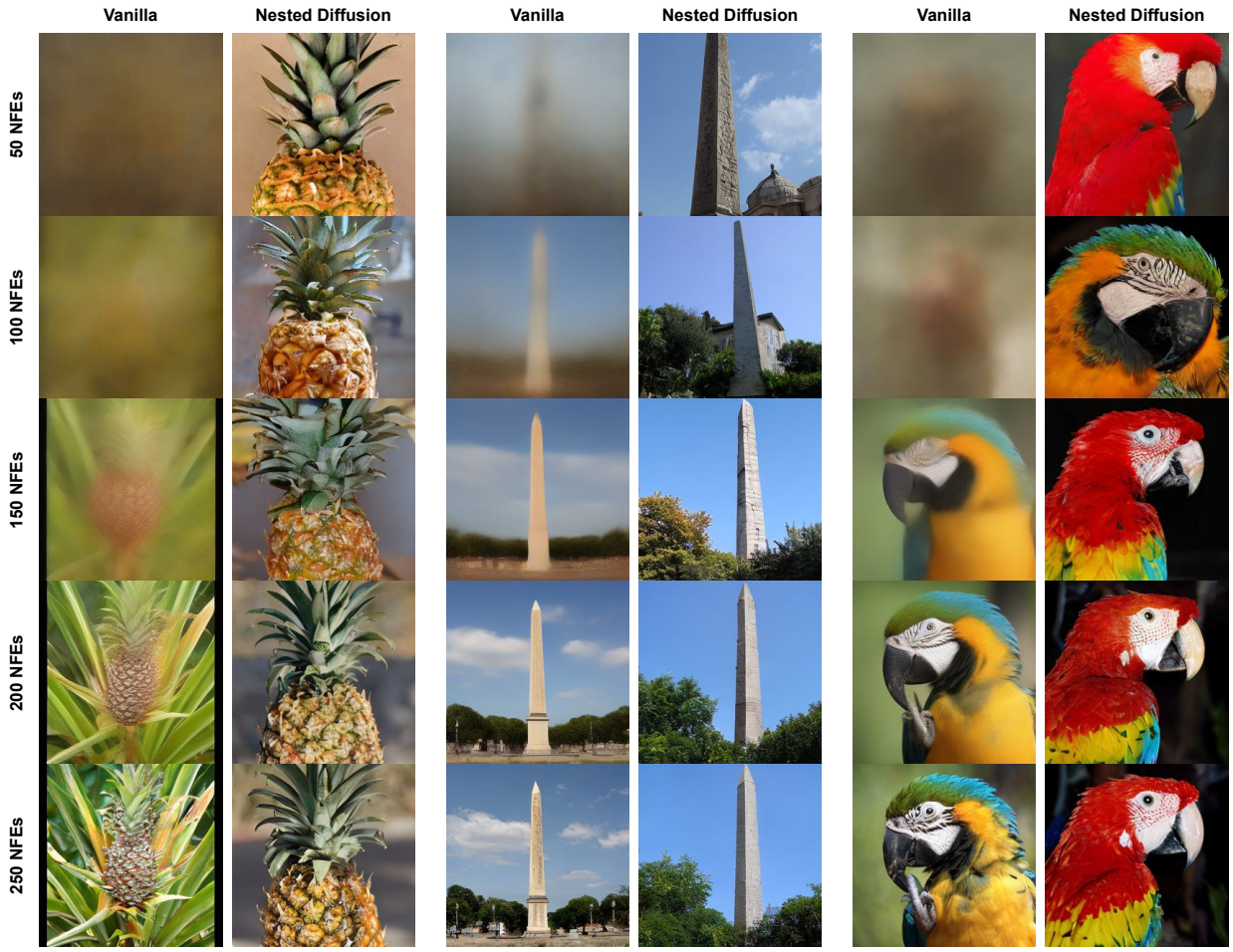


Figure 5. **Samples of class-conditional ImageNet generation, comparing vanilla DMs against Nested Diffusion.**

the final result's FID (without early termination) of Nested Diffusion is comparable to the vanilla diffusion. Exact FID values can be found in the supplementary material.

The sample quality trend for intermediate **inner samples** $\{\hat{\mathbf{x}}_0'\}$ is visualized in Figure 4 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 $\hat{\mathbf{x}}_0$. We observe that the graph bears similarity to simulated annealing with restarts. Nested Diffusion would return the last $\hat{\mathbf{x}}_0$ computed if terminated prematurely – corresponding to the local minima in the graph, shown in green.

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 [41]. We use Stable Diffusion to test Nested Diffusion for text-to-image generation. Similarly to Section 4.1, Stable Diffusion's process runs in a latent space, and uses CFG [18] for text-conditional sampling. In Figure 1, we present intermediate results from Nested Diffusion using 4 outer steps and compare them to their counterparts 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 DMs do not correspond to valid latents. As a result, when these latents are decoded, they do not produce natural-looking images.

Following previous work [2,40,41,43], we assess Nested Diffusion's performance in text-to-image generation using 30K FID on the MS-COCO [29] validation dataset. The results, presented in Figure 3b, surpass our previous findings, with Nested Diffusion demonstrating comparable intermediate results to vanilla diffusion and slightly improved final results. More examples for generated images and CLIP-Scores [15] can be found in the supplementary material.

To assess Nested Diffusion's potential and efficiency

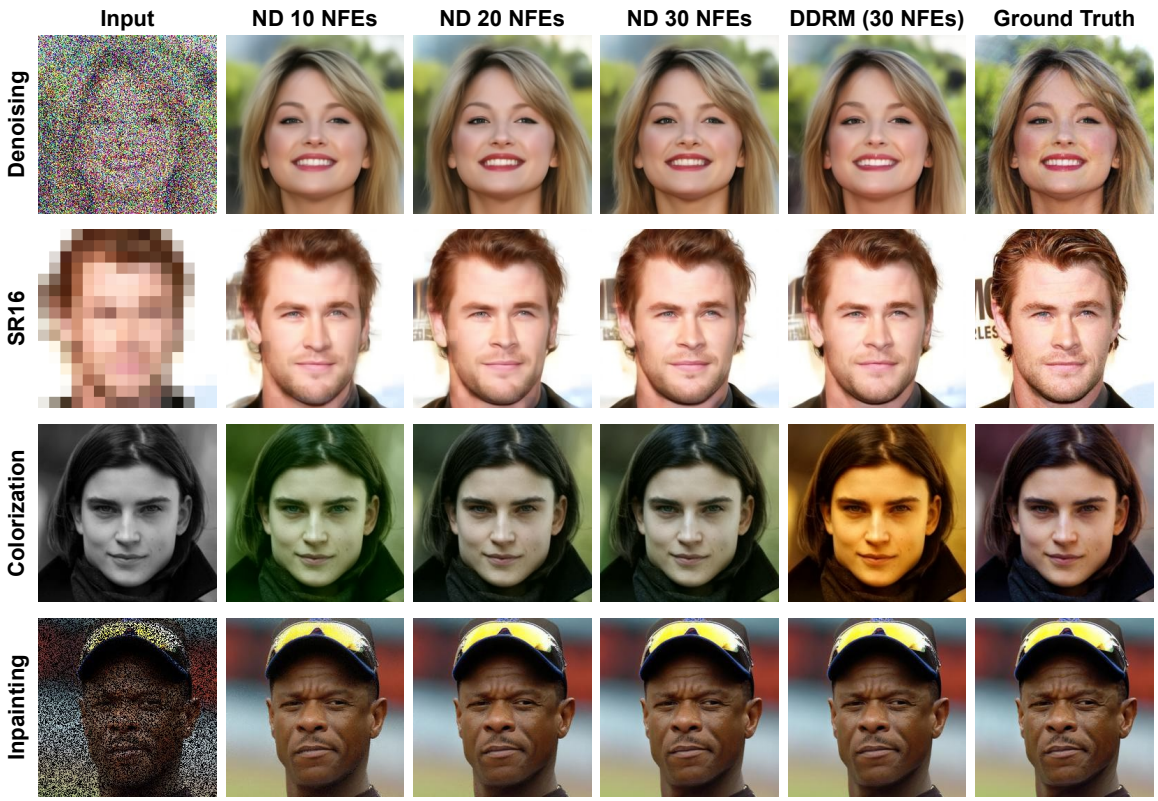


Figure 6. **Inverse problem solutions with Nested Diffusion on CelebA-HQ256 using DDRM.** Nested Diffusion is denoted as ND.

with advanced high-order schedulers, we replicated the text-to-image experiment while employing DPM-Solver++ [33] as the inner diffusion sampling schedule. This change enables using 10-20 NFEs for high quality samples, accelerating generation. As shown in Figure 3c, Nested Diffusion's final result is of comparable quality to vanilla DM and intermediate prediction quality is improved, demonstrating Nested Diffusion's potential use of high-order solvers.

4.3. Inverse Problem Solving

DMs have demonstrated their effectiveness in tackling inverse problems, whether by training conditional DNNs tailored for specific tasks [42] or by adapting unconditional DMs DNNs using modified sampling algorithms [6, 22, 24, 34, 49]. Following our notation, these inverse problem solvers sample using Algorithm 1, but exchange the DNN $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ for a conditional $p_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{y})$, where $\mathbf{y}$ represents the available measurements. To apply Nested Diffusion in inverse problem solving, a similar substitution is made in the Nested Diffusion sampling Algorithm 2, where the entire inner diffusion process is replaced with a diffusion-based inverse problem solver conditioned on $\mathbf{y}$. Analogous to image generation scenarios, Nested Diffusion transforms the inverse problem solver into an anytime algorithm, producing plausible results during the sampling process. An exact inverse problem solving algorithm using Nested diffusion is included in the supplementary material.

To evaluate the efficiency of Nested Diffusion for inverse problems, we conduct experiments on the CelebA-HQ256 dataset [21], employing DDRM [22] as the inverse problem solver. Following DDRM, we rely on a pre-trained DDPM [35], and use default hyperparameters except for number of sampling steps used. The results, depicted in Figure 6, demonstrate the generalization capabilities of Nested Diffusion in tackling inverse problems like inpainting, super-resolution, colorization and denoising. The algorithm produces valid intermediate predictions and achieves comparable final results, demonstrating its effectiveness in addressing various inverse problems.

5. Generation With Human Feedback

An emerging area of interest in guided image generation focuses on tuning the generated results to the user's preferences [28, 57]. This type of guidance typically requires user interaction with the model during training, attempting to fine-tune the DMs's generation process using direct feedback. The fine-tuned models show a greater capability to match the model's behaviour with the user's demands.

Nested Diffusion, by its inherent design, allows users to view the generated output throughout the sampling algo-

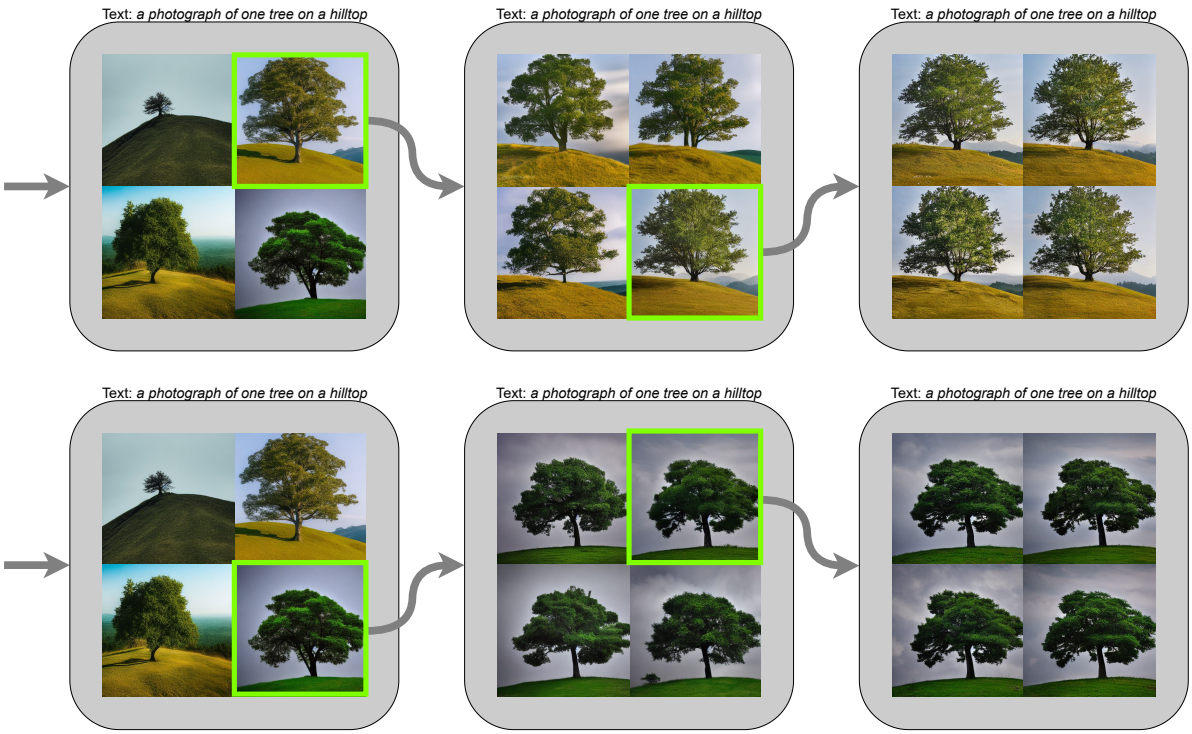


Figure 7. **An example of generation with human feedback.** The top and bottom graphs differ by the user's preference for the image, marked with a bright green frame.

rithm, enabling straightforward guidance of the process towards desired outcomes. For instance, in many cases multiple images are generated simultaneously using different random noise vectors, to provide the user with several alternative results. If Nested Diffusion is used for sampling, a user can see a likeness of the final possibilities. By pruning unwanted generation attempts, computational resources can be efficiently allocated to explore additional options based on the remaining intermediate predictions.

In contrast with model fine-tuning methods, Nested Diffusion can incorporate human feedback inherently, with no requirement for further training. Moreover, Nested Diffusion may be combined with fine-tuned DMs to further enhance their consistency with the user's preferences.

Figure 7 shows an example of a human feedback-based generation scheme implemented using Nested Diffusion. The samples were generated following the generation details provided in Section 4.2, using 3 outer steps with 20 inner steps each. At the conclusion of each inner diffusion process, the user is presented with four intermediate samples, allowing them to select their desired output. The chosen sample is then propagated to replace the other samples, and the sampling algorithm resumes its execution.

In addition to selecting from a pool of several options to guide the generation, further refinement of the sampling procedure can be achieved by integrating editing techniques into the sampling process. This editing can be accomplished using one of the many available diffusion-based image editing methods [1, 4, 14, 25, 35] in tandem with Nested Diffusion, by modifying the intermediate $\hat{\mathbf{x}}_0$, similar to SDEdit [35], or adjusting the subsequent inner diffusion process. However, we adopt a simpler approach: we add details to the textual prompt at the conclusion of each inner diffusion process during text-to-image generation. In Figure 8, we show some promising results for our approach.

6. Related Work

The noise scheduling in reverse diffusion sampling has garnered considerable attention in recent years [5]. DDPM [17] implements a linearly increasing schedule, while IDPM [36] demonstrates the potential of cosine scheduling in achieving improved sampling outcomes. In DDIM [48], the authors eliminate the forward diffusion's Markovian assumption, resulting in a deterministic reverse process that can accelerate sampling. Using ODE solving methods [30, 32, 33], the sampling process can attain superior results and faster generation. Nested Diffusion, while not strictly a noise schedule, intertwines two separate noise schedules (the inner and outer diffusion processes) into one sampling process.

Creative scheduling of noise can be employed in other domains besides image generation. In the field of image editing, SDEdit [35] degrades an edited clean image with

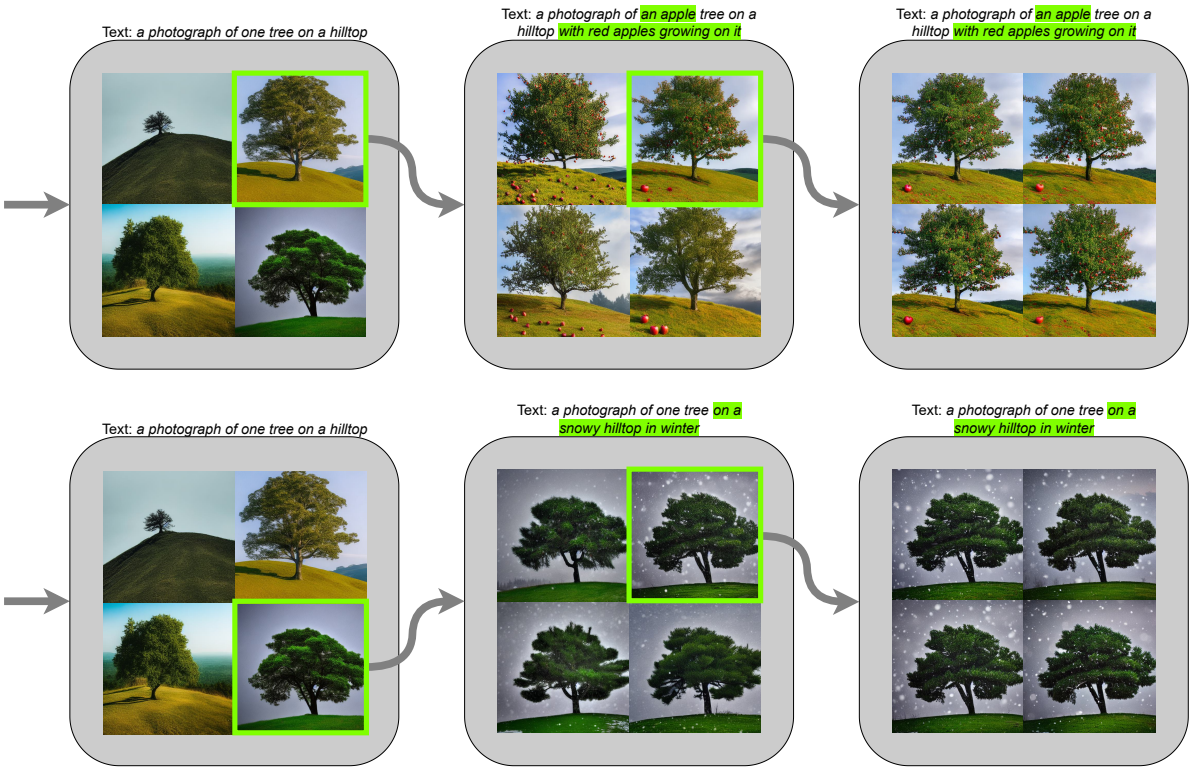


Figure 8. **An example of interactive content creation using human feedback.** The image selected by the user in each graph is marked with a bright green frame. The text prompt is changed after the first outer step.

noise and subsequently denoises it using a DM. This process enhances the realism of the edited image, facilitating photo-realistic editing using simple tools.

Noise has also been used in inverse problem solvers to "time-travel" in the diffusion process [34, 54]. These approaches revert the diffusion process to a previous step by adding random Gaussian noise, requiring additional NFEs and enhancing image fidelity. However, unlike Nested Diffusion, these methods add noise to revert a specific number of steps (a hyperparameter) and do not involve multiple diffusion processes. Consequently, they do not benefit from the anytime algorithm property and require more NFEs compared to alternative approaches.

Nested Diffusion is orthogonal to many diffusion acceleration methods, such as the fast sampling offered by DPM-Solver++ [33] shown in Section 4.2, and may work well with parallelized sampling [38, 46] or trajectory-based methods [31, 50]. In this work, we have not delved into some of these avenues for several reasons; Parallelized sampling typically requires more NFEs per image even when reducing overall sampling speed. Trajectory-based methods, while requiring a fraction of the resources, require additional training and may still fall short of achieving the performance standards set by multi-step diffusion sampling techniques [37, 41]. We have chosen to optimize Nested Diffusion for high quality at the expense of multiple steps and use NFEs to measure our computational resources. Nevertheless, combining these methods with anytime generation holds promise for future work.

7. Conclusion

We introduced Nested Diffusion, a probabilistic approach that harnesses a diffusion process as a building block in another diffusion process. Our approach allows anytime sampling from a pre-trained diffusion model. Through quantitative and qualitative evaluation, we demonstrated the effectiveness of Nested Diffusion in tandem with state-of-the-art DMs, including latent diffusion, CFG-based class-conditional generation, and text-to-image generation. Furthermore, we explored the potential of Nested Diffusion in enabling generation with human feedback and facilitating interactive content creation. Our findings highlight the versatility and practical applications of Nested Diffusion in various domains of generative modeling.

8. Acknowledgements

This work was supported by the Israel Science Foundation grant 2318/22, the Ollendorff Minerva Center, Technion, a gift from KLA, and the Council For Higher Education - Planning & Budgeting Committee, Israel.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, June 2022. 7

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 5, 13

[3] Mark Boddy and Thomas L Dean. *Solving time-dependent planning problems*. Brown University, Department of Computer Science, 1989. 1

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 1, 7

[5] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023. 7

[6] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 6

[7] Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated MRI. *Medical Image Analysis*, 80:102479, 2022. 1

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 4

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 11

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[11] Stephanie Fu*, Netanel Tamir*, Shobhita Sundaram*, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv:2306.09344*, 2023. 8, 11

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3

[13] Joshua Grass and Shlomo Zilberstein. Anytime algorithm development tools. *ACM SIGART Bulletin*, 7(2):20–27, 1996. 1

[14] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 1, 7

[15] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 5, 8, 11, 13

[16] 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*, volume 30, 2017. 4, 13

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 7

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4, 5, 13

[19] Michael C Horsch and David L Poole. An anytime algorithm for decision making under uncertainty. *arXiv preprint arXiv:1301.7384*, 2013. 1

[20] Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. Diff-tts: A denoising diffusion model for text-to-speech. *arXiv preprint arXiv:2104.01409*, 2021. 1

[21] 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. 4, 6, 14

[22] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022. 1, 4, 6, 13, 14

[23] Bahjat Kawar, Noam Elata, Tomer Michaeli, and Michael Elad. Gsure-based diffusion model training with corrupted data. *arXiv preprint arXiv:2305.13128*, 2023. 1

[24] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Stochastic image denoising by sampling from the posterior distribution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1866–1875, 2021. 1, 6

[25] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023. 1, 7

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[27] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 1

[28] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023. 6

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5, 13

[30] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022. 7

[31] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 8

[32] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022. 7

[33] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 1, 6, 7, 8, 13

[34] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 1, 6, 8

[35] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 6, 7

[36] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 7

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 4, 8, 11, 13

[38] Ashwini Pokle, Zhengyang Geng, and J Zico Kolter. Deep equilibrium approaches to diffusion models. *Advances in Neural Information Processing Systems*, 35:37975–37990, 2022. 8

[39] Zhuoran Qiao, Weili Nie, Arash Vahdat, Thomas F Miller III, and Anima Anandkumar. Dynamic-backbone protein-ligand structure prediction with multiscale generative diffusion models. *arXiv preprint arXiv:2209.15171*, 2022. 1

[40] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 5, 13

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 4, 5, 8, 13

[42] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 6

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 5, 13

[44] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 1

[45] Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022. 1

[46] Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. *arXiv preprint arXiv:2305.16317*, 2023. 8

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1, 2

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 1, 3, 4, 7, 13

[49] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations (ICLR)*, May 2023. 1, 6

[50] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 1, 8

[51] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2

[52] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2023. 1

[53] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021. 4

[54] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022. 1, 8

[55] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations (ICLR)*, 2022. 2, 3

[56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 8, 11

[57] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*, 2023. 6

[58] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *AI magazine*, 17(3):73–73, 1996. 1