

A Proof of Proposition 1

Given a shortcut model s_θ and its corresponding classifier-free guided output g_θ^w with guidance scale w . To analyze the generation process, we define a sequence of intermediate points $\{\mathbf{x}'_{\frac{i}{N}}\}_{i=0}^N$. This sequence is constructed by recursively applying N consecutive shortcut steps, each with the smallest step size $d = 1/N$. Note that N is chosen to be a power of 2. The process begins from the initial noise \mathbf{x}_0 at $t = 0$, conditioned on c . The starting point of the sequence is thus defined as $\mathbf{x}'_0 = \mathbf{x}_0$, and subsequent intermediate points are generated as follow:

$$\mathbf{x}'_{\frac{i+1}{N}} = \mathbf{x}'_{\frac{i}{N}} + s_\theta\left(\mathbf{x}'_{\frac{i}{N}}, \frac{i}{N}, c, d\right) d, \quad \text{for } i = 0, \dots, N-1. \quad (13)$$

Ideally, we assume that the model is perfectly trained, or equivalently, that the loss in Eq. (4) is minimized to zero. Under this assumption, we have:

$$s_\theta(\mathbf{x}'_t, t, c, 2d) = \frac{1}{2}[g_\theta^w(\mathbf{x}'_t, t, c, d) + g_\theta^w(\mathbf{x}'_{t+d}, t+d, c, d)], \quad (14)$$

$$s_\theta(\mathbf{x}'_t, t, \emptyset, 2d) = \frac{1}{2}[s_\theta(\mathbf{x}'_t, t, \emptyset, d) + s_\theta(\mathbf{x}'_{t+d}, t+d, \emptyset, d)]. \quad (15)$$

In the following derivation, we omit the time t in the network notation for simplicity, unless otherwise specified. First, we will prove that:

$$g_\theta^w(\mathbf{x}'_t, c, 2^j d) = \frac{1}{2^j} \sum_{i=0}^{2^j-1} g_\theta^{w^{j+1}}(\mathbf{x}'_{t+id}, c, d). \quad (16)$$

For the base case $j = 0$, we have $g_\theta^w(\mathbf{x}'_t, c, d) = g_\theta^w(\mathbf{x}'_t, c, d)$, which satisfies Eq. (16). Assuming that Eq. (16) holds for $j = k$, we now show that it also holds for $j = k+1$. Using Eq. (14) and Eq. (15), we obtain:

$$\begin{aligned} s_\theta(\mathbf{x}'_t, c, 2^{k+1}d) &= \frac{1}{2}[g_\theta^w(\mathbf{x}'_t, c, 2^k d) + g_\theta^w(\mathbf{x}'_{t+2^k d}, c, 2^k d)] \\ &= \frac{1}{2^{k+1}} \left[\sum_{i=0}^{2^k-1} g_\theta^{w^{k+1}}(\mathbf{x}'_{t+id}, c, d) + \sum_{i=0}^{2^k-1} g_\theta^{w^{k+1}}(\mathbf{x}'_{t+2^k d+id}, c, d) \right] \\ &= \frac{1}{2^{k+1}} \sum_{i=0}^{2^{k+1}-1} g_\theta^{w^{k+1}}(\mathbf{x}'_{t+id}, c, d), \\ s_\theta(\mathbf{x}'_t, \emptyset, 2^{k+1}d) &= \frac{1}{2}[s_\theta(\mathbf{x}'_t, \emptyset, 2^k d) + s_\theta(\mathbf{x}'_{t+2^k d}, \emptyset, 2^k d)] \\ &= \frac{1}{2^{k+1}} \sum_{i=0}^{2^{k+1}-1} s_\theta(\mathbf{x}'_{t+id}, \emptyset, d). \end{aligned}$$

Therefore, we have:

$$\begin{aligned} g_\theta^w(\mathbf{x}'_t, c, 2^{k+1}d) &= w s_\theta(\mathbf{x}'_t, c, 2^{k+1}d) + (1-w) s_\theta(\mathbf{x}'_t, \emptyset, 2^{k+1}d) \\ &= \frac{1}{2^{k+1}} \sum_{i=0}^{2^{k+1}-1} [w g_\theta^{w^{k+1}}(\mathbf{x}'_{t+id}, c, d) + (1-w) s_\theta(\mathbf{x}'_{t+id}, \emptyset, d)] \\ &= \frac{1}{2^{k+1}} \sum_{i=0}^{2^{k+1}-1} [w(w^{k+1} s_\theta(\mathbf{x}'_{t+id}, c, d) + (1-w^{k+1}) s_\theta(\mathbf{x}'_{t+id}, \emptyset, d)) + (1-w) s_\theta(\mathbf{x}'_{t+id}, \emptyset, d)] \\ &= \frac{1}{2^{k+1}} \sum_{i=0}^{2^{k+1}-1} [w^{k+2} s_\theta(\mathbf{x}'_{t+id}, c, d) + (1-w^{k+2}) s_\theta(\mathbf{x}'_{t+id}, \emptyset, d)] \\ &= \frac{1}{2^{k+1}} \sum_{i=0}^{2^{k+1}-1} g_\theta^{w^{k+2}}(\mathbf{x}'_{t+id}, c, d). \end{aligned}$$

404 This satisfies Eq. (16) for $j = k + 1$. By induction, we have Eq. (16) holds for all j . Based on this
 405 result, we can rewrite the output of the shortcut model for a single large step of size $Nd = 1$, denoted
 406 by $s_\theta(x_0, c, Nd)$, as follows:

$$\begin{aligned} s_\theta(x_0, c, Nd) &= \frac{1}{2} \left[g_\theta^w \left(x_0, c, \frac{N}{2}d \right) + g_\theta^w \left(x'_{\frac{1}{2}}, c, \frac{N}{2}d \right) \right] \\ &= \frac{1}{N} \left[\sum_{i=0}^{N/2-1} g_\theta^{w^{\log_2(N/2)+1}} \left(x'_{\frac{i}{N}}, c, d \right) + \sum_{i=0}^{N/2-1} g_\theta^{w^{\log_2(N/2)+1}} \left(x'_{\frac{1}{2} + \frac{i}{N}}, c, d \right) \right] \\ &= \frac{1}{N} \sum_{i=0}^{N-1} g_\theta^{w^{\log_2(N)}} \left(x'_{\frac{i}{N}}, c, d \right), \end{aligned}$$

407 which completes the proof.

408 B Multi-Level Wavelet Function

409 **Multi-Level Wavelet Function.** Algorithm 1 provides pseudo-code for our proposed multi-level
 410 wavelet objective described in Section 4.

Algorithm 1 Multi-level Wavelet Function

```

class MultiLevelWaveletLoss:
    def __init__(self):
        self.dwt = DWT_2D("haar") # Discrete Wavelet Transform (DWT)
        self.diff_func = MSELoss() # Distance function

    def concatenated_dwt(self, x):
        xll, xlh, xhl, xhh = self.dwt(x) # Decompose into 4 wavelet sub-bands using DWT
        details = torch.cat([xll, xlh, xhl, xhh], dim=1) # Concatenate the sub-bands
        return details

    def __call__(self, pred, target, num_levels):
        total_loss = diff_func(pred, target).mean() # Calculate low-level loss in original output space

        # Recursively calculate loss on wavelet sub-bands for each level
        pred_curr, target_curr = pred, target
        for current_level in range(num_levels):
            # Derive predicted and target sub-bands using outputs from previous level
            pred_bands = self.concatenated_dwt(pred_curr)
            target_bands = self.concatenated_dwt(target_curr)

            # Calculate loss on current level
            total_loss += diff_func(pred_bands, target_bands).mean()
            pred_curr, target_curr = pred_bands, target_bands

        # Taking average from all levels
        loss = total_loss / (num_levels + 1)
        return loss

```

411 **Results.** Fig. 3 presents qualitative comparisons illustrating the impact of different levels in our
 412 multi-level wavelet function versus traditional low-level loss. Incorporating more wavelet levels
 413 yields finer details and fewer artifacts, especially in one- and few-step generation.

414 C Injecting Conditional Inputs into Network

415 We explore two primary strategies for incorporating conditional information including CFG scale w ,
 416 current sample x_t , time t , condition c , and the target step size d into our network. The first strategy,
 417 similar to U-ViT [2], encodes each condition as an individual token appended to the input sequence
 418 of noisy image patch tokens. The second strategy employs AdaLN-Zero blocks [44] to modulate
 419 the network with each condition; these modulations are then aggregated through addition. We find
 420 that the AdaLN-Zero approach yields comparable performance to the U-ViT method but without the
 421 drawback of increased input sequence length. Given this efficiency benefit, we adopt AdaLN-Zero
 422 for injecting CFG information into our network.

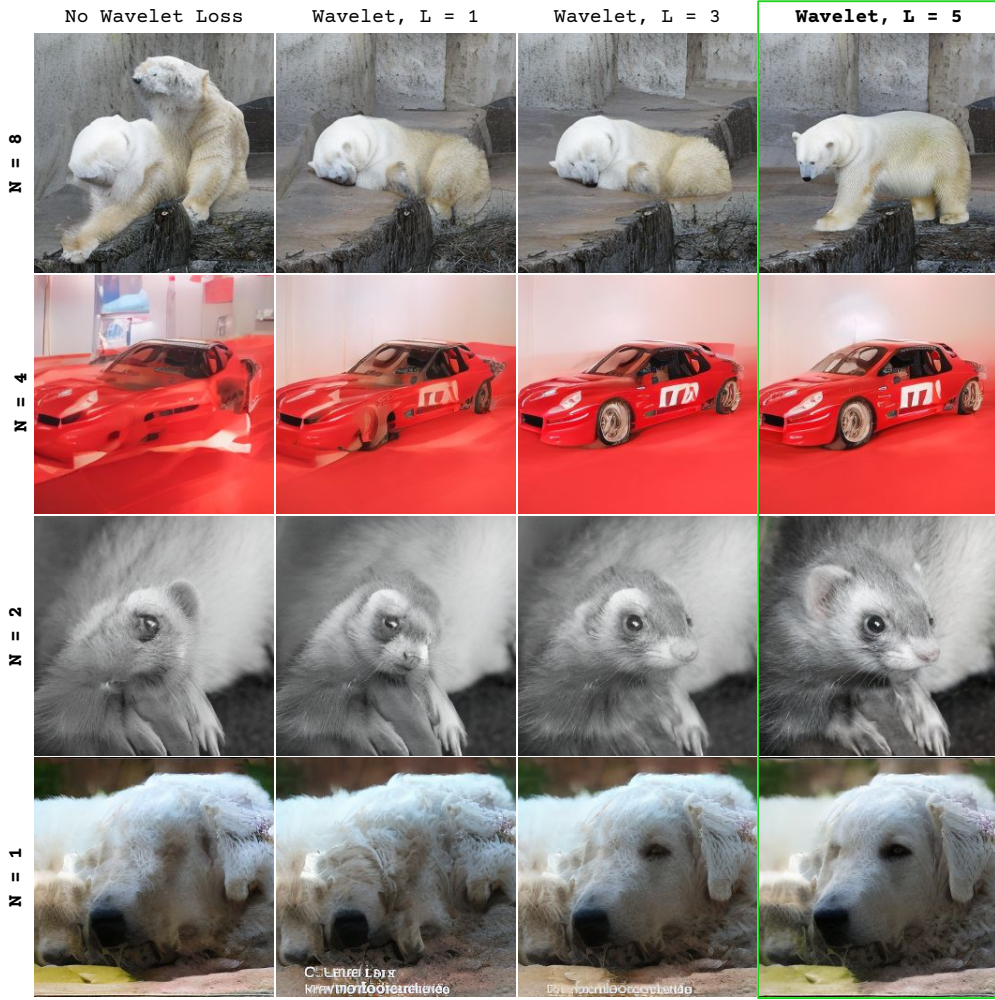


Figure 3: Effect of applying different numbers of wavelet loss layers (L) on image quality under one and few-step inference. The traditional low-level objective (leftmost column) results in noticeable degradation and artifacts. In contrast, using multi-level wavelet loss, especially with $L = 5$ (green box, right), produces high-quality, consistent images across all inference settings.

D Accelerating training speed

To mitigate the high computational cost of shortcut models, we explore resuming their training from existing a flow matching weights [66]. Table 3 shows that while resuming offers limited improvement for the original shortcut model, our improved shortcut models (iSM) demonstrate considerable gains. Specifically, when incorporating a series of proposed methods, iSM achieves significantly better FID scores. Models are trained for 250k steps for all evaluations.

Table 3: Performance of Improved Shortcut Models

Method	FID _{N=1} ↓	FID _{N=4} ↓
Shortcut Models [18]	21.38	13.46
<i>Improved Shortcut Models (iSM)</i>		
+ Integrated Guidance	9.62	3.17
+ Interval Guidance in Training	8.49	2.81
+ Multi-level Wavelet Function	8.12	2.64
+ Scaling OT Matching	7.97	2.23
+ Twin EMA	6.56	2.16

E Experiment Settings

E.1 Training & Parameterization Setting

For experiments on ImageNet, images are preprocessed to 256x256 resolution, following the protocol of ADM [12]. Our models operate in the latent space derived from the sd-vae-ft-mse autoencoder [1]. These latents are subsequently normalized channel-wise using means of [0.86488, -0.27787343, 0.21616915, 0.3738409] and standard deviations of [4.85503674, 5.31922414, 3.93725398, 3.9870003]. Finally, the normalized latents are scaled by a factor of 0.5, targeting an approximate standard deviation of 0.5 for the network input. Detailed configurations for all experiments are provided in Table 4.

Table 4: Experimental settings on ImageNet 256×256 .

Parameterization Setting	
Architecture	SiT-XL
GFlops	118.64
Params (M)	675
Flow Trajectory	OT-FM
Input dim.	$32 \times 32 \times 4$
Num. layers	24
Hidden dim.	1024
Num. heads	16
α_t	$1 - t$
σ_t	t
w_t	σ_t
Training objective	v-prediction
Training Setting	
Training iteration	800K
Dropout	0
Optimizer	AdamW
AdamW β_1	0.9
AdamW β_2	0.999
AdamW ϵ	10^{-8}
Learning Rate	0.0001
Weight Decay	0
Batch Size	256
Label Dropout	0.1
Methods	
CFG Scale w_{\max}	3.5
Interval t_{interval}	0.3
Wavelet Levels L	5
OT Scale K	32
Ratio of Empirical to Self-consistency Targets	0.25
EMA Parameters Used For Self-consistency Targets?	True
EMA Target Rate θ_{target}^-	0.95
EMA Parameters Used For Evaluation?	True
EMA Inference Rate θ_{infer}^-	0.9999

443 E.2 Evaluation Details

444 We follow the ADM [12] evaluation setup, using the same reference batches from their official
 445 implementation,¹ and compute FID [20] over 50K generated images.

446 E.3 Baselines

447 Below, we summarize the baseline methods used in our evaluation.

- 448 • **ADM** [12] improves U-Net-based diffusion architectures and introduces classifier-guided
 449 sampling to balance sample quality and diversity.
- 450 • **CDM** [23] proposes cascaded diffusion models, which generate images in a coarse-to-fine
 451 manner, similar to ProgressiveGAN [27].
- 452 • **Simple diffusion** [25] leverages a diffusion model for high-resolution images by simplifying
 453 the noise schedule and model architectures.

¹<https://github.com/openai/guided-diffusion/tree/main/evaluations>

- 454 • **LDM** [48] introduces the concept of operating diffusion models in a compressed latent
455 space, enhancing efficiency.
- 456 • **U-DiT** [57] proposes a series of U-shaped DiTs based on self-attention with downsampled
457 tokens.
- 458 • **U-ViT** [2] adapts Vision Transformers for latent diffusion by incorporating U-Net-like long
459 skip connections.
- 460 • **DiT** [44] pioneers the use of a pure transformer architecture as the backbone for diffusion
461 models, featuring AdaIN-zero modules.
- 462 • **SiT** [41] investigates how to improve DiT training by transitioning from discrete diffusion
463 to continuous flow-based modeling.
- 464 • **REPA** [66] accelerates the training of DiT/SiT models by regularizing network representa-
465 tions to align with features from pretrained visual encoders.
- 466 • **FlowDCN** [59] offers a fully convolutional architecture for generative modeling with linear
467 time and memory complexity, enabling efficient high-resolution image synthesis.
- 468 • **iCT** [53] introduces several techniques to enhance the training of CMs [54], including a log-
469 normal noise schedule, Pseudo-Huber loss functions, and a scheduler for total discretization
470 steps during training.
- 471 • **SM** [18] establishes a framework for one-step generation by combining flow matching with
472 a self-consistency objective.
- 473 • **IMM** [68] few-step generative models by inductively matching all moments of bootstrapped
474 samples derived from stochastic interpolants using Maximum Mean Discrepancy (MMD),
475 aiming for stable convergence.

476 **F Related Works**

477 **F.1 Diffusion, Consistency & Flow Matching models**

478 Generative modeling has seen significant advancements with methods like diffusion models, con-
479 sistency models, and flow matching. Diffusion models [52, 55, 22, 30] are a generative framework
480 that synthesizes data by gradually transforming random noise into outputs via a stochastic denoising
481 process. To introduce efficient generation, consistency models [54, 53, 39] aim for efficient gen-
482 eration by learning a mapping from any point on a solution trajectory directly to the data origin.
483 Meanwhile, flow matching [37, 38] proposes learning generative models by defining a target vector
484 field between noise and data and training a neural network to approximate this field. Both these
485 methods have demonstrated strong performance when scaled to text-to-image [48, 49, 45, 9, 15] and
486 text-to-video [21, 4, 43] applications.

487 **F.2 End-to-end training for one-to-many step generative models**

488 The quest for efficient, high-quality few-step generation has explored several avenues. While
489 early one-step models leveraged GANs [19, 28, 5] and MMD [36, 34], the inherent instability and
490 complexity of adversarial training limit their scalability. Consistency Models (CMs) [54, 53, 39]
491 address the instability of adversarial training without needing synthetic datasets. Unlike distillation,
492 CMs can be trained from scratch via consistency training (CT), independent of pre-trained diffusion
493 models. However, few-step generation with CMs usually involves discrete-time variants. These
494 variants require meticulous timestep scheduling and are susceptible to irreducible bias accumulation
495 due to the inherent ambiguity of discretization. Inductive Moment Matching (IMM) [68] further
496 advanced stable few-step training using moment matching. Shortcut Models (SMs) [18] introduced
497 step-size conditioning with self-bootstrapping. However, original SMs (Section 3) suffer from
498 inflexible guidance and guidance accumulation artifacts, which our work directly addresses.

499 **G More Qualitative Results**

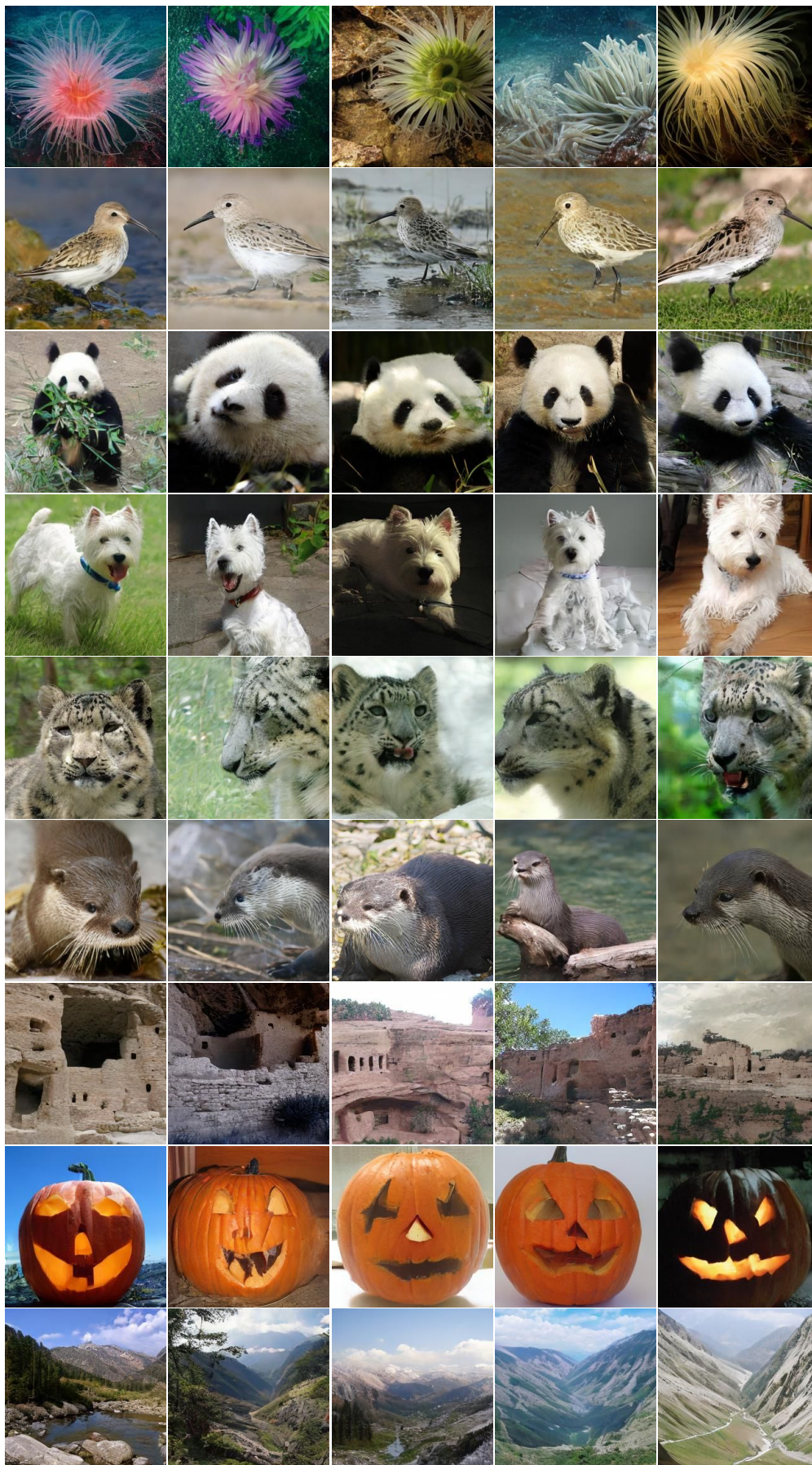


Figure 4: Uncurated samples on ImageNet-256 \times 256 with 1-step sampling.



Figure 5: Uncurated samples on ImageNet-256 \times 256 with 2-step sampling.



Figure 6: Uncurated samples on ImageNet-256 \times 256 with 4-step sampling.

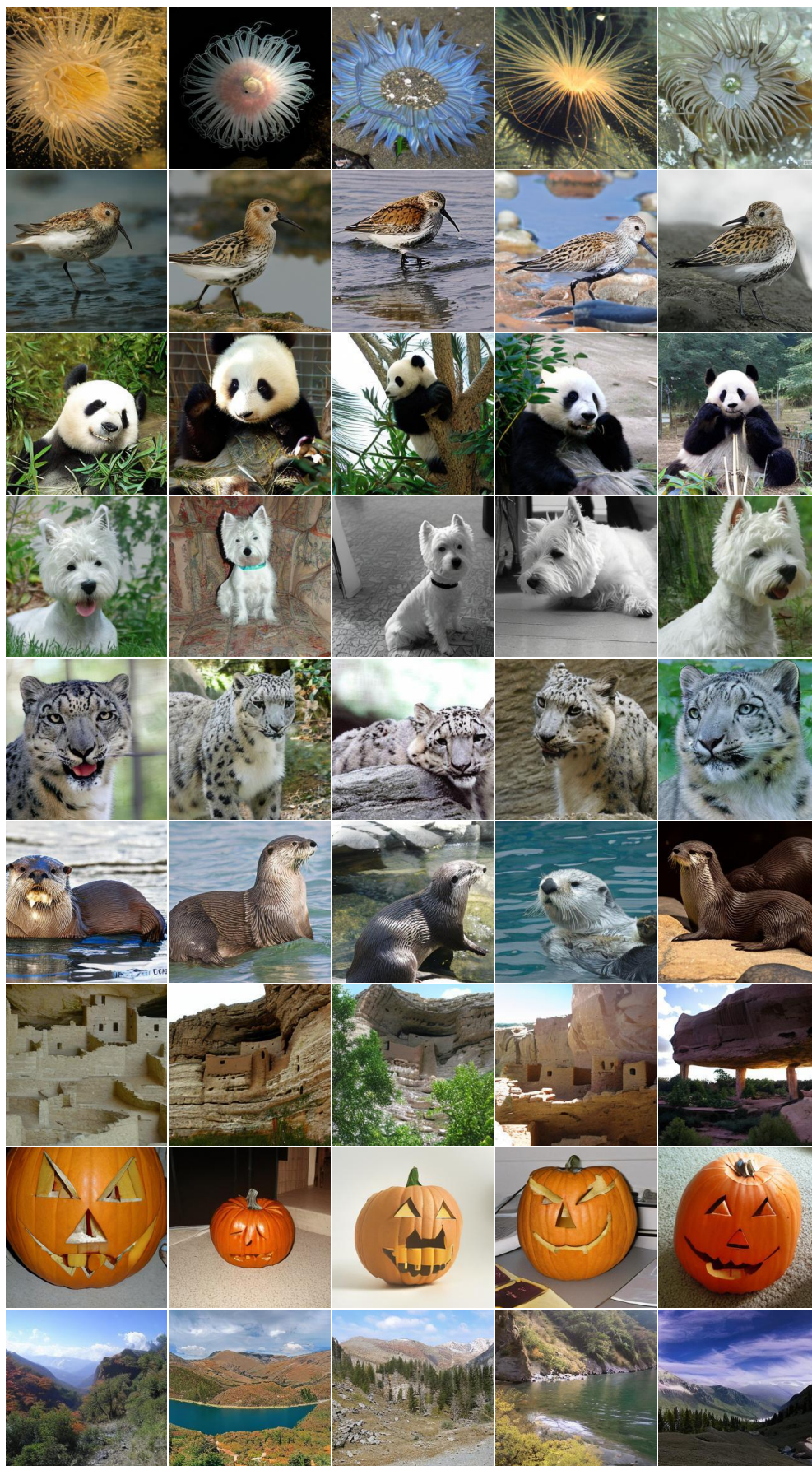


Figure 7: Uncurated samples on ImageNet-256 \times 256 with 8-step sampling.

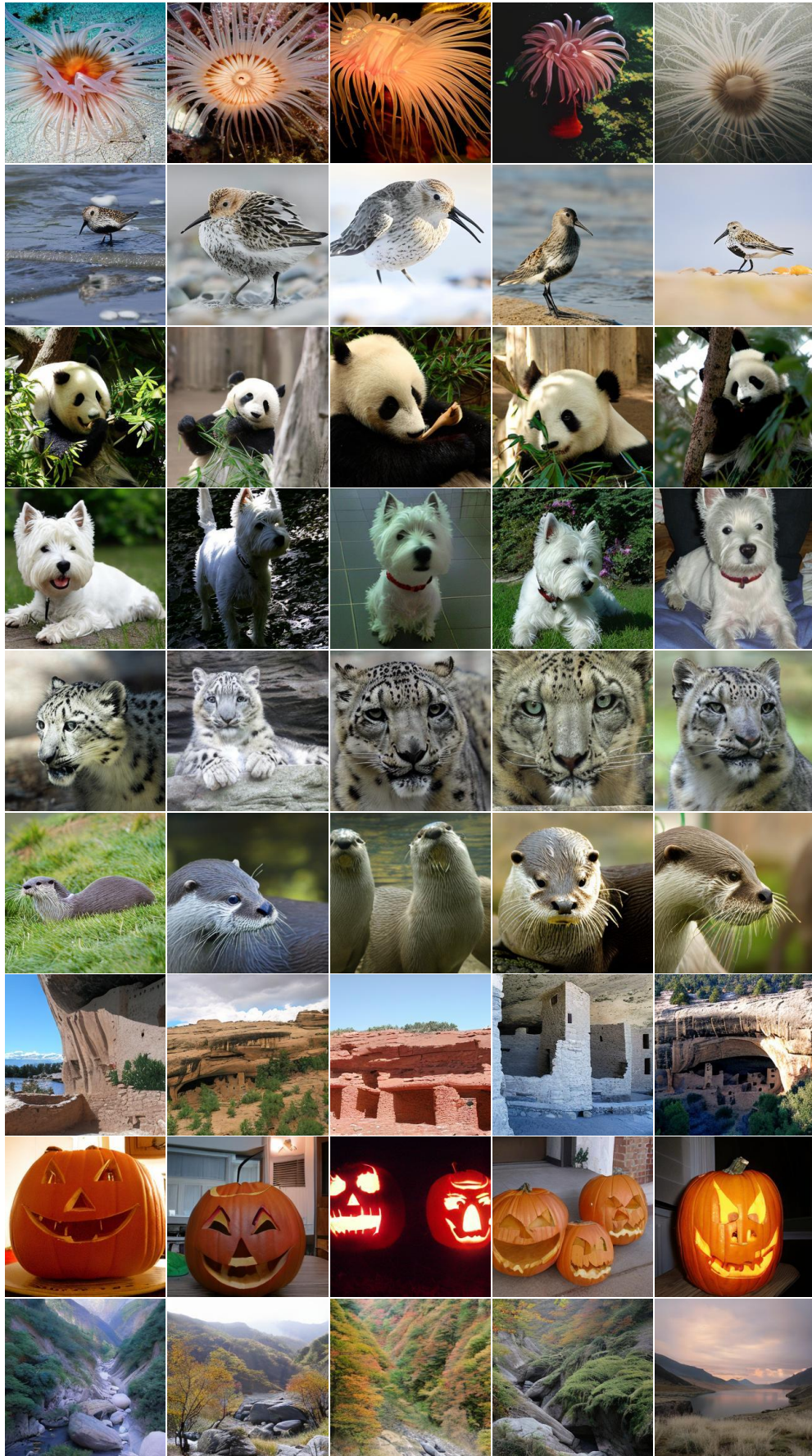


Figure 8: Uncurated samples on ImageNet-256 \times 256 with 128-step sampling.

References

- [1] S. AI. Fine-tuned vae for stable diffusion (sd-vae-ft-mse). <https://huggingface.co/stabilityai/sd-vae-ft-mse>, 2023. Accessed: 2025-04-28.
- [2] F. Bao, S. Nie, K. Xue, Y. Cao, C. Li, H. Su, and J. Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023.
- [3] R. Basri, M. Galun, A. Geifman, D. Jacobs, Y. Kasten, and S. Kritchman. Frequency bias in neural networks for input of non-uniform density. In *International conference on machine learning*, pages 685–694. PMLR, 2020.
- [4] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
- [5] A. Brock. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.
- [6] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.
- [7] C. Chen, R. Qian, W. Hu, T.-J. Fu, J. Tong, X. Wang, L. Li, B. Zhang, A. Schwing, W. Liu, et al. Dit-air: Revisiting the efficiency of diffusion model architecture design in text to image generation. *arXiv preprint arXiv:2503.10618*, 2025.
- [8] J. Chen, S. Xue, Y. Zhao, J. Yu, S. Paul, J. Chen, H. Cai, E. Xie, and S. Han. Sana-sprint: One-step diffusion with continuous-time consistency distillation, 2025.
- [9] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, et al. Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.
- [10] Y. Chen, G. Li, C. Jin, S. Liu, and T. Li. Ssd-gan: Measuring the realness in the spatial and spectral domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1105–1112, 2021.
- [11] T. Dao, T. H. Nguyen, T. Le, D. Vu, K. Nguyen, C. Pham, and A. Tran. Swiftbrush v2: Make your one-step diffusion model better than its teacher. In *European Conference on Computer Vision*, pages 176–192. Springer, 2024.
- [12] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [13] P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn’s algorithm. In *International conference on machine learning*, pages 1367–1376. PMLR, 2018.
- [14] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.
- [15] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.
- [16] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.
- [17] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pages 3247–3258. PMLR, 2020.

- [18] K. Frans, D. Hafner, S. Levine, and P. Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.
- [19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [20] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017.
- [21] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [22] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [23] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.
- [24] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [25] E. Hoogeboom, J. Heek, and T. Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.
- [26] M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023.
- [27] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.
- [28] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. 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.
- [29] M. Khayatkhoei and A. Elgammal. Spatial frequency bias in convolutional generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7152–7159, 2022.
- [30] D. Kingma, T. Salimans, B. Poole, and J. Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.
- [31] T. Kynkäänniemi, M. Aittala, T. Karras, S. Laine, T. Aila, and J. Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024.
- [32] B. F. Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024.
- [33] S. Lee, B. Kim, and J. C. Ye. Minimizing trajectory curvature of ode-based generative models. In *International Conference on Machine Learning*, pages 18957–18973. PMLR, 2023.
- [34] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30, 2017.
- [35] T. Li, Y. Tian, H. Li, M. Deng, and K. He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

- [36] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *International conference on machine learning*, pages 1718–1727. PMLR, 2015.
- [37] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- [38] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- [39] C. Lu and Y. Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.
- [40] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.
- [41] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden, and S. Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024.
- [42] T. H. Nguyen and A. Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7807–7816, 2024.
- [43] OpenAI. Video generation models as world simulators. <https://openai.com/sora/>, 2024.
- [44] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.
- [45] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- [46] A.-A. Pooladian, H. Ben-Hamu, C. Domingo-Enrich, B. Amos, Y. Lipman, and R. T. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *International Conference on Machine Learning*, pages 28100–28127. PMLR, 2023.
- [47] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.
- [48] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [49] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- [50] K. Schwarz, Y. Liao, and A. Geiger. On the frequency bias of generative models. *Advances in Neural Information Processing Systems*, 34:18126–18136, 2021.
- [51] K. Schwarz, Y. Liao, and A. Geiger. On the frequency bias of generative models. *Advances in Neural Information Processing Systems*, 34:18126–18136, 2021.
- [52] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- [53] Y. Song and P. Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.
- [54] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

- [55] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [56] K. Tian, Y. Jiang, Z. Yuan, B. Peng, and L. Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.
- [57] Y. Tian, Z. Tu, H. Chen, J. Hu, C. Xu, and Y. Wang. U-dits: Downsample tokens in u-shaped diffusion transformers. *arXiv preprint arXiv:2405.02730*, 2024.
- [58] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. Kou, Y. Lv, Y. Li, Y. Liu, Y. Wang, Y. Zhang, Y. Huang, Y. Li, Y. Wu, Y. Liu, Y. Pan, Y. Zheng, Y. Hong, Y. Shi, Y. Feng, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv: 2503.20314*, 2025.
- [59] S. Wang, Z. Li, T. Song, X. Li, T. Ge, B. Zheng, and L. Wang. Flowdcn: Exploring dcn-like architectures for fast image generation with arbitrary resolution. *arXiv preprint arXiv:2410.22655*, 2024.
- [60] E. Xie, J. Chen, J. Chen, H. Cai, H. Tang, Y. Lin, Z. Zhang, M. Li, L. Zhu, Y. Lu, and S. Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformer, 2024.
- [61] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.
- [62] Z.-Q. J. Xu, Y. Zhang, and Y. Xiao. Training behavior of deep neural network in frequency domain. In *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I* 26, pages 264–274. Springer, 2019.
- [63] J. Yao, C. Wang, W. Liu, and X. Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. *Advances in Neural Information Processing Systems*, 37:56166–56189, 2024.
- [64] T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and B. Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.
- [65] T. Yin, M. Gharbi, R. Zhang, E. Shechtman, F. Durand, W. T. Freeman, and T. Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.
- [66] S. Yu, S. Kwak, H. Jang, J. Jeong, J. Huang, J. Shin, and S. Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.
- [67] Y. Zhang and B. Hooi. Hipa: enabling one-step text-to-image diffusion models via high-frequency-promoting adaptation. *arXiv preprint arXiv:2311.18158*, 2023.
- [68] L. Zhou, S. Ermon, and J. Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025.