Scalable High-Resolution Pixel-Space Image Synthesis with Hourglass Diffusion Transformers

Katherine Crowson [*1] **Stefan Andreas Baumann** [*2] **Alex Birch** [*3] **Tanishq Mathew Abraham** [1]
Daniel Z. Kaplan [4] **Enrico Shippole** [5]

Abstract

We present the Hourglass Diffusion Transformer (HDiT), an image-generative model that exhibits linear scaling with pixel count, supporting training at high resolution (e.g. 1024×1024) directly in pixel-space. Building on the Transformer architecture, which is known to scale to billions of parameters, it bridges the gap between the efficiency of convolutional U-Nets and the scalability of Transformers. HDiT trains successfully without typical high-resolution training techniques such as multiscale architectures, latent autoencoders or self-conditioning. We demonstrate that HDiT performs competitively with existing models on ImageNet 256^2, and sets a new state-of-the-art for diffusion models on FFHQ-1024^2. Code is available at `github.com/crowsonkb/k-diffusion`.

Figure 1: Samples generated directly in RGB pixel space using our ⧗ HDiT models trained on FFHQ-1024^2 and ImageNet-256^2.

1. Introduction

Diffusion models have emerged as the pre-eminent method for image generation, as evidenced by state-of-the-art approaches like Stable Diffusion (Rombach et al., 2022), Imagen (Saharia et al., 2022), eDiff-I (Balaji et al., 2023), or Dall-E 2 (Ramesh et al., 2022). They are versatile, succeeding in modalities such as video and audio (Blattmann et al., 2023; Kong et al., 2021). They boast scalability, training stability, and output diversity.

Diffusion model architectures employ diverse backbones, spanning CNN-based (Ho et al., 2020), transformer-based (Peebles & Xie, 2023; Bao et al., 2023a), CNN-transformer-hybrid (Hoogeboom et al., 2023), or even state-space models (Yan et al., 2023). There is likewise variation in the approaches used to scale these models to support high-resolution image synthesis. Current approaches add complexity to training, necessitate additional models, or sacrifice quality.

Latent diffusion models (Rombach et al., 2022) (LDMs) reign as the dominant method for achieving high-resolution image synthesis. In practice, they fail to represent fine detail (Dai et al., 2023, see also Figure 2), impacting sample quality and limiting its utility in applications such as image editing. Other approaches to high-resolution synthesis include cascaded super-resolution (Saharia et al., 2022), multi-scale losses (Hoogeboom et al., 2023), the incorporation of inputs and outputs at multiple resolutions (Gu et al., 2023), or the utilization of self-conditioning and the adaptation of fundamentally new architecture schemes (Jabri et al., 2023).

[*]Equal contribution [1]Stability AI, United States [2]CompVis @ LMU Munich, Germany [3]Birchlabs, England, United Kingdom [4]realiz.ai, New York, United States [5]Independent Researcher, Florida, United States. Correspondence to: Katherine Crowson <crowsonkb@gmail.com>, Stefan Baumann <stefan.baumann@lmu.de>, Alex Birch <alex@birchlabs.co.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

We seek to advance the state of pixel-space diffusion, offering a path to synthesis at high resolutions without resorting to LDMs. Eliminating the latent VAE frees us from quality limitations endemic to such VAEs (illustrated in Figure 2), and bolsters downstream applications such as image editing (a process which LDMs encumber with poor reconstruction). We expound the merits of pixel-space DMs versus LDMs in Appendix B.

Our work tackles high-resolution synthesis via backbone improvements, which grant the efficiency needed to target pixel-space directly. We introduce a pure transformer architecture inspired by the hierarchical structure introduced in (Nawrot et al., 2022), which we call the Hourglass Diffusion Transformer (⧗ HDiT). Our backbone is capable of high-quality image generation at megapixel scale in standard diffusion setups. This architecture, even at low spatial resolutions such as 128×128 is substantially more efficient than common diffusion transformer backbones such as DiT (Peebles & Xie, 2023) (see Table 1 and Figure 8) while being competitive in generation quality. When scaling the model architecture to target resolutions according to our scheme, we obtain $\mathcal{O}(n)$ computational complexity scaling with the target number of image tokens n in place of the $\mathcal{O}(n^2)$ scaling of normal diffusion transformer architectures, making this the first transformer-based diffusion backbone competitive in computational complexity with convolutional U-Nets for pixel-space high-resolution image synthesis.

Our main contributions are as follows:

- We introduce the Hourglass Diffusion Transformer (⧗ HDiT), which achieves subquadratic scaling of compute with resolution. We show how our architecture choices help improve upon the quality of the baseline DiT (Peebles & Xie, 2023) in pixel-space image synthesis.
- We demonstrate high-quality pixel-space generation at 1024×1024 resolutions, setting a state-of-the-art FID for diffusion models on FFHQ-1024^2. We do so without training complications such as progressive growing or multiscale losses.
- We show HDiT's competence in a large-scaling training scenario through competitive evaluation on ImageNet-256^2. Quantitatively, it measures well against even latent transformer-based diffusion models despite undertaking the training at a higher effective resolution.

Figure 2: Motivation: Detail loss incurred through use of a standard VAE (Rombach et al., 2022) on one of Figure 1 samples. Notably, this VAE is employed by the baseline DiT (Peebles & Xie, 2023) architecture against which we compare.

2. Related Work

2.1. Transformers

The transformer architecture (Vaswani et al., 2017) reigns as state-of-the-art in various domains (OpenAI, 2023; Zong et al., 2022; Zhang et al., 2022b; Yu et al., 2022; Piergiovanni et al., 2023). It has been scaled to tens of billions of parameters in the vision domain, (Dehghani et al., 2023) and beyond that in natural language processing (Chowdhery et al., 2023; Fedus et al., 2022). Transformers consider interactions between all elements in the sequence via the attention mechanism. Long-range interactions can be learned, but computational complexity scales quadratically with the length of input sequence.

Transformer-based Diffusion Models Recent works have applied transformers to diffusion models. Diffusion priors (Ramesh et al., 2022) have provided low-dimensional embeddings on which to condition image synthesis, and latent diffusion (Rombach et al., 2022) has achieved state-of-the-art performance generating images from compressed image latents (Peebles & Xie, 2023; Bao et al., 2023a; Zheng et al., 2023; Gao et al., 2023; Bao et al., 2023b; Chen et al., 2023a;b). Transformer-based architectures (Hoogeboom et al., 2023; Jing et al., 2023) have been applied to U-Nets (Ronneberger et al., 2015), either at the lowest levels (Ho et al., 2020), or by altogether hybridizing the two architectures (Cao et al., 2022). The quadratic computational complexity of transformers' attention mechanism precludes high-resolution synthesis in pixel-space (Yang et al., 2022b); latent representations are typically used to reduce the operating resolution.

Diffusion Transformers (DiT) (Peebles & Xie, 2023), are amenable to masked training (Gao et al., 2023; Zheng et al., 2023), which incentivizes models to better learn feature relationships. It is orthogonal and complementary to the architecture improvements pursued in this work.

Hourglass Transformers The Hourglass architecture (Nawrot et al., 2022) is a hierarchical implementation of transformers, shown to be more efficient at language modeling than standard Transformer models, in training and in inference. Sequences are shortened as they descend

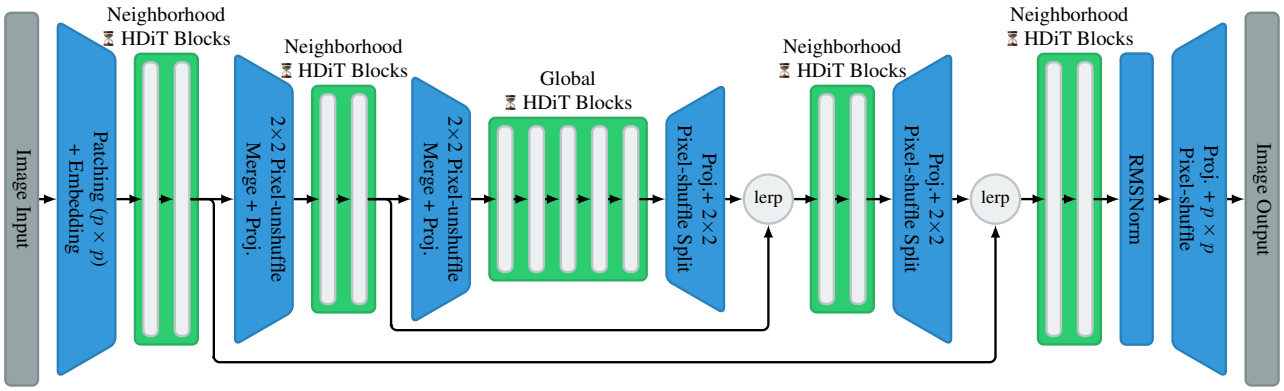


Figure 3: High-level overview of our ⧗ HDiT architecture, specifically the version for ImageNet at input resolutions of 256^2 at patch size $p = 4$, which has three levels. For any doubling in target resolution, another neighborhood attention block is added. "lerp" denotes a linear interpolation with learnable interpolation weight. All ⧗ HDiT blocks have the noise level and the conditioning (embedded jointly using a mapping network) as additional inputs.

the encoder levels of the transformer, culminating in the shortest representation in the middle, then re-expanded as they ascend the decoder levels. Skip connections reintroduce higher-resolution information near the expansion steps. Hourglasses resemble U-Nets (Ronneberger et al., 2015) without convolutional layers. Hierarchical structures (Wang et al., 2022) have excelled at image restoration, a task similar to the denoising objective pursued in diffusion.

2.2. High-Resolution Image Synthesis with Diffusion Models

High-resolution image synthesis in diffusion models has been extensively studied, yet it remains a challenge to current single-stage models. Popular approaches separate the generation process into multiple steps. Cascaded super-resolution (Ho et al., 2021) targets initially a low-resolution image, scaling it via a series of super-resolution models. Latent diffusion targets a spatially downsampled "latent" representation, which can be decoded into a higher-resolution pixel image via a convolutional model (Rombach et al., 2022) or another diffusion model (Betker et al., 2023). The latent representation can itself also be super-resoluted (Fischer et al., 2023). Latent diffusion is the strategy chosen by most transformer-based diffusion models (see Section 2.1). Recent works explore high-resolution image synthesis in pixel space, in an effort to simplify the overall architecture. Fundamentally new backbone architectures (Jabri et al., 2023) have been proposed. Spatial dimensions have been reduced via discrete wavelet transforms (Hoogeboom et al., 2023). The diffusion training process has not stood still, with proposals such as self-conditioning across sampling steps (Jabri et al., 2023), multiresolution training (Gu et al., 2023), and multiresolution losses (Hoogeboom et al., 2023) offering a path to higher resolutions. The necessity of such substantial modifications of the diffusion process is proving difficult to overcome, with simpler approaches (Song et al., 2021) – single-stage and lacking the aforementioned training adaptations – struggling to produce samples that fully utilize the available resolution and are globally coherent.

3. Preliminaries

3.1. Diffusion Models

Diffusion models generate data by learning to reverse a diffusion process. This diffusion process is most commonly defined to be a Gaussian noising process. Given a data distribution $p_{\text{data}}(\mathbf{x})$, we define a *forward* noising process with the family of distributions $p(\mathbf{x}_{\sigma_t}; \sigma_t)$ that is obtained by adding i.i.d. Gaussian noise of standard deviation σ_t which is provided by a predefined monotonically increasing noise level schedule. Therefore, $\mathbf{x}_{\sigma_t} = \mathbf{x}_0 + \sigma_t \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. A denoising neural network $D_\theta(\mathbf{x}_{\sigma_t}, \sigma_t)$ is trained to predict $\mathbf{x}_0$ given $\mathbf{x}_{\sigma_t}$. Sampling is done by starting at $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{max}}^2 \mathbf{I})$ and sequentially denoising at each of the noise levels before resulting in the sample $\mathbf{x}$. The denoiser neural network is trained with a mean-squared error loss:

$$\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\epsilon, \sigma_t \sim p(\epsilon, \sigma_t)} \left[\lambda_{\sigma_t} \| D_\theta(\mathbf{x}_{\sigma_t}, \sigma_t) - \mathbf{x} \|_2^2 \right], \quad (1)$$

where λ_{σ_t} is a weighting function.

Recent works proposed various improvements to this basic formulation. Two notable approaches, which are also adapted by our model, are preconditioning to obtain more suitable prediction targets for the model (Karras et al., 2022) and adapting the loss weighting to a clamped signal-to-noise ratio (SNR) $\lambda_{\sigma_t} = \min\{\frac{1}{\sigma_t}, \gamma\}$ to improve model convergence (Hang et al., 2023). Another improvement has been the adaption of noise schedules for high resolutions. It was previously observed (Hoogeboom et al., 2023) that the com-

monly used noise schedules that were originally designed for low resolutions (32x32 or 64x64) fail to add enough noise at high resolutions. Therefore, the noise schedules can be shifted and interpolated from a reference low-resolution noise schedule in order to add appropriate noise at higher resolutions.

4. Hourglass Diffusion Transformers

Diffusion Transformers (Peebles & Xie, 2023) and other similar works (see Section 2.1) have demonstrated impressive performance as denoising diffusion autoencoders in latent diffusion (Rombach et al., 2022) setups, surpassing prior works in terms of generative quality (Gao et al., 2023; Zheng et al., 2023). However, their scalability to high resolutions is limited by the fact that the computational complexity increases quadratically ($\mathcal{O}(n^2)$ for images of shape $h \times w \times$ channels, with $n = w \cdot h$). This makes them prohibitively expensive to train and run on high-resolution inputs, effectively limiting transformers to spatially compressed latents at sufficiently small dimensions, unless very large patch sizes are used (Cao et al., 2022), which have been found to be detrimental to the quality of generated samples (Peebles & Xie, 2023).

We propose a new, improved hierarchical architecture based on Diffusion Transformers (Peebles & Xie, 2023), and Hourglass Transformers (Nawrot et al., 2022) – Hourglass Diffusion Transformers (⌛ HDiT) – that enables high-quality pixel-space image generation and can be efficiently adapted to higher resolutions with a computational complexity scaling of $\mathcal{O}(n)$ instead of $\mathcal{O}(n^2)$. This means that even scaling up these models to direct pixel-space generation at megapixel resolutions becomes viable, which we demonstrate for models at resolutions of up to 1024×1024 in Section 5.

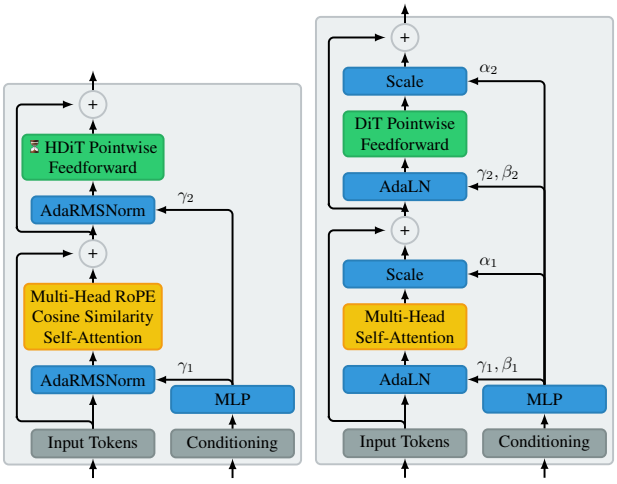
4.1. Leveraging the Hierarchical Nature of Images

Natural images exhibit hierarchies (Saremi & Sejnowski, 2013). This makes mapping the image generation process into a hierarchical model an intuitive choice, which has previously been successfully applied in the U-Net architecture (Ronneberger et al., 2015) commonly used in diffusion models but is not commonly used by diffusion transformers (Peebles & Xie, 2023; Bao et al., 2023a). To leverage this hierarchical nature of images for our transformer backbone, we apply the hourglass structure (Nawrot et al., 2022), which has been shown to be effective for a range of different modalities, including images, for the high-level structure of our transformer backbone. Based on the model's primary resolution, we choose the number of levels in the hierarchy, such that the innermost level has 16×16 tokens. We use a larger hidden dimension for lower-resolution levels, which have to process both low-resolution information and information relevant for following higher-resolution levels. For every level on the encoder side, we spatially merge 2×2 tokens into one using Pixel-UnShuffle (Shi et al., 2016) and do the inverse on the decoder side.

Skip Merging Mechanism One important consideration in hierarchical architectures is the merging mechanisms of skip connections, as it can influence the final performance significantly (Bao et al., 2023a). While the previous non-hierarchical U-ViT (Bao et al., 2023a) uses a concatenation-based skip implementation, similar to the standard U-Net (Ronneberger et al., 2015), and found this to be significantly better than other options, we find additive skips to perform better for this hierarchical architecture. As the usefulness of the information provided by the skips can differ significantly, especially in very deep hierarchies, we additionally enable the model to learn the relative importance of the skip and the upsampled branch by learning a linear interpolation (lerp) coefficient f between the two for each skip:

$$\mathbf{x}_{\text{merged}}^{(\text{l. lerp})} = f \cdot \mathbf{x}_{\text{skip}} + (1 - f) \cdot \mathbf{x}_{\text{upsampled}}. \quad (2)$$

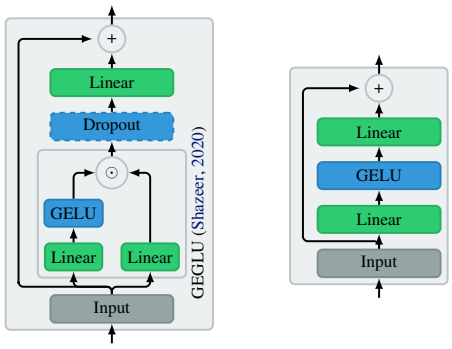
4.2. Hourglass Diffusion Transformer Block Design



(a) ⌛ HDiT Block Architecture. (b) DiT Block Architecture.

Figure 4: A comparison of our transformer block architecture and that used by DiT (Peebles & Xie, 2023).

Our basic transformer block design (shown in comparison with that of DiT in Figure 4) is generally inspired by the blocks used by LLaMA (Touvron et al., 2023), a transformer architecture that has recently been shown to be very capable of high-quality generation of language. To enable conditioning, we make the output scale used by RMSNorm operations adaptive, predicted by a mapping network conditioned on the class and diffusion time step. Unlike DiT, we do not employ an (adaptive) output gate, but initialize the output projections of both self-attention and FFN blocks to zeros. To make positional information accessible to the



(a) ⧗ HDiT FFN Block. (b) DiT FFN Block.

Figure 5: A comparison of our pointwise feedforward block architecture and that used by DiT (Peebles & Xie, 2023).

transformer model, common diffusion transformer architectures like DiT and U-ViT use a learnable additive positional encoding. (Peebles & Xie, 2023; Bao et al., 2023a) As it is known to improve models' generalization and their capability of extrapolating to new sequence lengths, we replace this with an adaptation of rotary positional embeddings (RoPE) (Su et al., 2022) for 2D image data: we follow an approach similar to (Ho et al., 2019) and split the encoding to operate on each axis separately, applying RoPE for each spatial axis to distinct parts of query and key respectively. We also found that applying this encoding scheme to only half of the query and key vectors and not modifying the rest to be beneficial for performance. Overall, we find empirically that replacing the normal additive positional embedding with our adapted RoPE improves convergence and helps remove patch artifacts. Additionally to applying RoPE, we use a cosine similarity-based attention mechanism that has previously been used in (Liu et al., 2022a) (see Appendix E.1 for details). We note that a similar approach has been proven at the multi-billion parameter scale for vision transformers (Dehghani et al., 2023).

For the feedforward block (see Figure 5 for a comparison with DiT), instead of having an output gate like DiT, we use GEGLU (Shazeer, 2020), where the modulation signal comes from the data itself instead of the conditioning and is applied on the first instead of the second layer of the FFN.

4.3. Efficient Scaling to High Resolutions

The hourglass structure enables us to process an image at a variety of resolutions. We use global self-attention at low resolutions to achieve coherence, and local self-attention (Liu et al., 2021; 2022a; Hassani et al., 2023) at all higher resolutions to enhance detail. This limits the need for quadratic-complexity global attention to a manageable amount, and enjoys linear-complexity scaling for any further increase in resolution. Asymptotically, the complexity is $\mathcal{O}(n)$ (see Appendix A) w.r.t pixel count n.

A typical choice for localized self-attention would be Shifted Window attention (Liu et al., 2021; 2022a) as used by previous diffusion models (Cao et al., 2022; Li et al., 2022). We find, however, that Neighborhood attention (Hassani et al., 2023) performs significantly better in practice.

The maximum resolution at which to apply global self-attention[1] is a choice determined by dataset (the size at which small features requiring long-distance coherence become large enough for attention to reason about) and by task (the smallest feature whose long-distance relationships need to be preserved in order to be acceptable). At particularly low resolutions (e.g. 256^2), some datasets permit coherent generation with fewer levels of global attention.

5. Experiments

We evaluate the proposed ⧗ HDiT architecture on conditional and unconditional image generation, ablating over architectural choices (Section 5.2), and evaluating both megapixel pixel-space image generation (Section 5.3) and large-scale pixel-space image generation (Section 5.4).

5.1. Experimental Setup

Training Unless mentioned otherwise, we train class-conditional models on ImageNet (Deng et al., 2009) at a resolution of 128×128 directly on RGB pixels without any kind of latent representation. We adapt our general training setup from (Karras et al., 2022), including their preconditioner, and use a continuous-time diffusion formulation. We train all models with AdamW (Loshchilov & Hutter, 2019) using a constant learning rate of 5×10^{-4} and a weight decay of $\lambda = 0.01$. We generally train at a batch size of 256 for 400k steps (following (Peebles & Xie, 2023)) with stratified diffusion timestep sampling and do not use Dropout unless noted otherwise. For small-scale ImageNet trainings at 128×128, we do not apply any augmentation. For runs on small datasets, we apply a non-leaking augmentation scheme akin to (Karras et al., 2020a). Following common diffusion model training practice and (Peebles & Xie, 2023), we also compute the exponential moving average (EMA) of the model weights with a decay of 0.9999. We use this EMA version of the model for all evaluations and generated samples, and perform our sampling using 50 steps of DPM++(3M) (Lu et al., 2023) SDE sampling. For further details, see Table 7.

[1]For our FFHQ-1024^2 experiment, we apply two levels of global attention – one at 16^2 and one at 32^2. Whereas for ImageNet-128^2 and 256^2, we found like prior works (Ho et al., 2020; Hoogeboom et al., 2023; Nichol & Dhariwal, 2021) that a single level of 16^2 global attention suffices.

Evaluation Following common practice for generative image models, we report the Fréchet Inception Distance (FID) (Heusel et al., 2017) computed on 50k samples. To compute FID, we use the commonly used implementation from (Dhariwal & Nichol, 2021). We also report both the absolute and asymptotic computational complexity for our main ablation study, also including FLOPs for higher-resolution versions of the architecture.

5.2. Effect of the Architecture

To evaluate the effect of our architectural choices, we perform an ablation study where we start with a basic implementation of the hourglass architecture for diffusion and iteratively add the changes that enable our final architecture to efficiently perform high-quality megapixel image synthesis. We denote the ablation steps as **A**, **B1**, ..., **E**, and show their feature composition and experimental results in Table 1. We also provide a set of baselines **R1**, ..., **R4**, where we trained DiT (Peebles & Xie, 2023) models in various settings to enable a fair comparison. Additional experimental steps are shown in Appendix D.1.

We generally use DiT-B-scale models for this comparison (approx. 130M parameters for DiT, approx 105M to 120M for ⧗ HDiT depending on the ablation step), due to their relatively low training cost, and train them on pixel-space ImageNet (Deng et al., 2009) at a resolution of 128^2 and patch size of 4. The computational cost for the same architecture at resolutions of 256×256 and 512×512 is also reported. In the case of our models, every doubling in resolution involves adding one local attention block (except for ablation step **A**, where it is global) as per Section 4.1.

Baselines We train multiple versions of DiT in different setups to provide fair comparisons with it as baselines in Table 1. **R1** directly uses the official DiT implementation (Peebles & Xie, 2023) but omits the VAE latent computation step and adjusts the scaling to fit the data. No other changes were made, as DiT can be directly applied to pixel space (Peebles & Xie, 2023). We also train a baseline **R3** that uses the DiT-B hyperparameters and structure but applies them to our block architecture and training setup as used in **A**. This matches the performance of the original DiT trained with the original codebase. On top of this setup, we also add soft-min-snr loss weighting to **R4** (as in ablation step **E**) to enable a fair comparison with our final model.

Base Hourglass Structure Configuration **A** is a simple hourglass structure with lower-resolution levels and our linear skip interpolations, and the basic implementation of our blocks with RMSNorm, but without GEGLU, and with full global self-attention at every level. A simple additive positional encoding is used here. Even this simple architecture, without any of our additional changes, is already substantially cheaper (30% of the FLOPs per forward pass, less for higher resolutions) than similarly-sized DiT (Peebles & Xie, 2023) models operating in pixel space due to the hourglass structure. For higher resolutions than 128^2, this makes it viable to train pixel-space transformer-based models at all. This comes at the cost of increased FID compared to the DiT baselines at this step in the ablation.

Local Attention Mechanism Next, we add local attention to all levels except for the lowest-resolution one. We evaluate two options – Shifted-Window (Swin) (Liu et al., 2021; 2022a) attention (**B1**, a common choice in vision transformers and previously also used in diffusion models (Cao et al., 2022; Li et al., 2022)) and Neighborhood (Hassani et al., 2023) attention (**B2**). Both result in a small reduction in FLOPs even at the low-resolution scale of 128×128 but, most importantly, reduce the computational complexity w.r.t. the base resolution from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, enabling practical scaling to significantly higher resolutions. Both variants suffer from increased FID due to this reduced expressiveness of local attention. Still, this change is significantly less pronounced for Neighborhood attention, making it a clearly superior choice in this case compared to the common choice of Swin attention.

Feedforward Activation As the third step, we ablate over using GEGLU (Shazeer, 2020), where the data itself affects the modulation of the outputs of the feedforward block, compared to the standard GeLU for the feedforward network. Similar to previous work (Touvron et al., 2023), to account for the effective change of the hidden size due to the GEGLU operation, we decrease the hidden dimension from $4 \cdot d_{\text{model}}$ to $3 \cdot d_{\text{model}}$. We find that this change significantly improves FID at the cost of a slight increase in computational cost, as the width of the linear projections in the feedforward block has to be increased to account for the halving in output width.

Positional Encoding Next, we replace the standard additive positional embedding with our 2D axial adaptation of RoPE (Su et al., 2022) in **D** (see Appendix E.2 for details), completing our Hourglass DiT backbone architecture. This further improves FID. As an additional benefit, RoPE should enable significantly better extrapolation to other resolutions than additive positional embeddings, but this ablation study does not test for that. Qualitatively, we find that this also helps reduce patching artifacts in the generated images.

Loss Weighting Finally, we also ablate over replacing the standard $\frac{1}{\sigma^2}$ loss weighting (Ho et al., 2020; Song et al., 2021) with our adapted Min-SNR (Hang et al., 2023) loss weighting method that we call Soft-Min-SNR (see Appendix C), which reduces the loss weight compared to SNR weighting for low noise levels. This substantially improves FID further, demonstrating the effectiveness of ⧗ HDiT when coupled with an appropriate training setup for pixel-space diffusion.

Table 1: Ablation of our architectural choices, starting from a stripped-down implementation of our hourglass diffusion transformer that is similar to DiT-B/4 (Peebles & Xie, 2023). We also ablate over our additional choice of using soft-min-snr loss weighting, which we use to train our full models but do not consider part of our architecture. We present results for various DiT-B/4-based models as baselines. We also report computational cost per forward pass at multiple resolutions, including standard resolution-dependent model adaptations (relative to **R1** in gray). See Table 5 for an additional results.

	Configuration	FID↓	GFLOP@128^2↓	Complexity↓	GFLOP@256^2	GFLOP@512^2
Baselines (**R1** uses 250 DDPM sampling steps with learned $\sigma(t)$ as in the original publication instead of 50-step DPM++ sampling)						
R1	DiT-B/4 (Peebles & Xie, 2023)	42.03	106	$\mathcal{O}(n^2)$	657	6,341
R3	**R1** + our basic blocks & mapping net & trainer	42.49	106	$\mathcal{O}(n^2)$	657	6,341
R4	**R3** + Soft-Min-SNR	<u>30.71</u>	106	$\mathcal{O}(n^2)$	657	6,341
Ablation Steps						
A	Global Attention Diffusion Hourglass (Section 4.1)	50.76	32	$\mathcal{O}(n^2)$	114 (−83%)	1,060 (−83%)
B1	**A** + Swin Attn. (Liu et al., 2021)	55.93	**29**	$\boldsymbol{\mathcal{O}(n)}$	60 (−91%)	185 (−97%)
B2	**A** + Neighborhood Attn. (Hassani et al., 2023)	51.07	**29**	$\boldsymbol{\mathcal{O}(n)}$	60 (−91%)	184 (−97%)
C	**B2** + GeGLU (Shazeer, 2020)	44.36	<u>31</u>	$\boldsymbol{\mathcal{O}(n)}$	65 (−90%)	198 (−96%)
D	**C** + Axial RoPE (Section 4.2)	41.41	<u>31</u>	$\boldsymbol{\mathcal{O}(n)}$	65 (−90%)	198 (−96%)
E	**D** + Soft-Min-SNR (Appendix C)	**27.74**	<u>31</u>	$\boldsymbol{\mathcal{O}(n)}$	65 (−90%)	198 (−96%)

Table 2: Skip Information Merging Mechanism Ablation

Skip Implementation	FID↓
Concatenation (U-Net (Ronneberger et al., 2015))	33.75
Addition (Original Hourglass (Nawrot et al., 2022))	<u>28.37</u>
Learnable Linear Interpolation (**Ours**)	**27.74**

Skip Implementation Additionally to the main ablation study, we also ablate over different skip implementations based on ablation step **E**. We compare our learnable linear interpolation (lerp), which we empirically found to be especially helpful when training deep hierarchies, with both a standard additive skip, where the upsampled and skip data are directly added, and a concatenation version, where the data is first concatenated and then projected to the original channel count using a pointwise convolution. The results of this ablation are shown in Table 2. We find that, even for shallow hierarchies as used for ImageNet-128^2 generation in our ablations, the learnable linear interpolation outperforms the addition slightly, with both the learnable lerp and addition substantially outperforming the commonly used concatenation.

5.3. High-Resolution Pixel-Space Image Synthesis

In this section, we train our model for high-resolution pixel-space image synthesis. Following previous works, we train on FFHQ-1024^2 (Karras et al., 2021b), the standard benchmark dataset for image generation at such high resolutions.

Previous works use self-conditioning (Jabri et al., 2023), multi-scale architectures (Gu et al., 2023), or multi-scale losses (Hoogeboom et al., 2023) to enable synthesis at high resolutions. Our model does not require such tricks (though we expect them to further increase quality), and we train without them, with the exception of adapting the SNR at each step according to the increase in the images' redundancy (Hoogeboom et al., 2023). Our model generates high-quality, globally coherent samples (see Figure 6) that utilize the high resolution to produce sharp pictures with fine details.

Figure 6: Samples from our 85M-parameter FFHQ-1024^2 model. Best viewed zoomed in.

We benchmark our models against state-of-the-at counterparts in Table 3 for a quantitative comparison. We find that our model substantially outperforms this baseline both quantitatively and qualitatively (see Figure 14 and Figure 15 for samples from our method and competing methods). Notably, our model excels in generating faces with symmetric features, while the only other diffusion model, NCSN++, exhibits noticeable asymmetry. Moreover, ⧗ HDiT effectively leverages the available resolution, producing sharp

Table 3: Comparison of our results on FFHQ 1024 × 1024 to other models in the literature. 50k samples are used for FID computation unless specified otherwise.[2] FD_{D2} and KD_{D2} denote Fréchet and Kernel DINOv2 distances respectively.

Method	Params	FID↓	FD_{D2}↓	KD_{D2}↓
Diffusion Models (5k samples)				
NCSN++ (Song et al., 2021)	106M	53.52	608	1.879
⧗ HDiT-85M (**Ours**)	85M	**8.48**	**177**	**0.348**
Diffusion Models				
⧗ HDiT-85M (**Ours**)	85M	5.23	**149**	**0.354**
Generative Adversarial Networks				
HiT-B (Zhao et al., 2021)	117M	6.37	-	-
StyleSwin (Zhang et al., 2022a)	41M	5.07	360	0.946
StyleGAN2 (Karras et al., 2020b)	30M	2.70	253	0.578
StyleGAN3-T (Karras et al., 2021a)	22M	2.79	249	0.575
StyleGAN3-R (Karras et al., 2021a)	16M	3.07	273	0.651
StyleGAN-XL (Sauer et al., 2022)	71M	**2.02**	270	0.644

and finely detailed images, a notable improvement over the NCSN++ model, which often yields blurry samples, and also other competing methods. We find that our model is competitive regarding FID with high-resolution transformer GANs such as HiT (Zhao et al., 2021) or StyleSwin (Zhang et al., 2022a), but does not reach the same FID as state-of-the-art GANs such as StyleGAN-XL (Sauer et al., 2022). We evaluate with DINOv2-based metrics also, as FID is known to be flawed for evaluating FFHQ generation (Kynkäänniemi et al., 2023) and to advantage GAN-generated samples (Stein et al., 2023). Our model sets a new state-of-the-art for DINOv2-based Fréchet and Kernel distances, metrics which correlate better with human preference than their Inception counterparts (Stein et al., 2023).

5.4. Large-Scale ImageNet Image Synthesis

Earlier experiments (see Section 5.3) show ⧗ HDiT's sample fidelity at high resolutions. To evaluate capabilities at scale, we train a class-conditional pixel-space ImageNet-256^2 model. This 557M parameter model is smaller than many state-of-the-art models, and has not been hyperparameter-tuned. As in our high-resolution experiments, we refrain from applying non-standard training tricks or diffusion modifications, and, consistent with (Hoogeboom et al., 2023), we compare results without the application of classifier-free guidance, emphasizing an out-of-the-box comparison.

We show samples in Figure 7 and compare quantitatively with state-of-the-art diffusion models in Table 4. We find that qualitatively our model can generate high-fidelity samples on this task. Compared to the baseline model DiT, our model achieves a substantially lower FID and higher IS despite operating on pixel-space instead of lower-resolution latents. Compared to other single-stage pixel-space diffusion models, our model outperforms simple U-Net-based models such as ADM but is outperformed by models that use self-conditioning during sampling (RIN) or are substantially larger (simple diffusion, VDM++).

[2] We compare to NCSN++ on FID@5k due to its sampling cost, which for FID@50k would be similar to training our model.

Figure 7: Samples from our class-conditional 557M-parameter ImageNet-256^2 model without CFG.

Table 4: Comparison of our results on ImageNet-256^2 to other models in the literature. Following (Hoogeboom et al., 2023), we report results without classifier-free guidance. Besides FID@50k and IS@50k, we also report trainable parameter count, samples seen (training iterations times batch size), and sampling steps.

Method	Params	It.×BS	Steps	FID↓	IS↑
Latent Diffusion Models					
LDM-4 (Rombach et al., 2022)	400M+VAE	214M	250	10.56	209.5
DiT-XL/2 (Peebles & Xie, 2023)	675M+VAE	1.8B	250	9.62	121.5
U-ViT-H/2 (Bao et al., 2023a)	501M+VAE	512M	50·2	6.58	-
MDT-XL/2 (Gao et al., 2023)	676M+VAE	1.7B	250	6.23	143.0
MaskDiT/2 (Zheng et al., 2023)	736M+VAE	2B	40·2	5.69	178.0
Single-Stage Pixel-Space Diffusion Models					
iDDPM (Nichol & Dhariwal, 2021)	-	-	250	32.50	-
ADM (Dhariwal & Nichol, 2021)	554M	507M	1000	10.94	101.0
RIN (Jabri et al., 2023)	410M	614M	1000	4.51	161.0
simple diffusion (Hoogeboom et al., 2023)	2B	1B	512	2.77	211.8
VDM++ (Kingma & Gao, 2023)	2B	-	256·2	2.40	225.3
⧗ HDiT (**Ours**)	557M	742M	50·2	6.92	135.2

6. Conclusion

This work presents ⧗ HDiT, a hierarchical pure transformer backbone for diffusion image synthesis which scales to high resolutions more efficiently than previous transformer-based backbones. It adapts to the target resolution, processing local phenomena at high resolutions and global phenomena at low resolutions. Its computational complexity at higher resolutions scales with $\mathcal{O}(n)$ instead of $\mathcal{O}(n^2)$, bridging the gap between the scalability of transformer models and the efficiency of U-Nets. We demonstrate megapixel-scale pixel-space synthesis without tricks such as self-conditioning or multiresolution architectures, whilst staying competitive with other transformer diffusion backbones even at small

resolutions, both in fairly matched pixel-space settings, and when compared to transformers in latent-space.

7. Future Work

⧗ HDiT provides a basis for further research into efficient high-resolution image synthesis. While we only focus on unconditional and class-conditional image synthesis, ⧗ HDiT is likely well-suited to enhancing efficiency and performance in other generative tasks like super-resolution, text-to-image generation and other modalities such as audio and video, especially with architecture scaling. This work studied ⧗ HDiT in the context of pixel-space diffusion models but future works could investigate applying ⧗ HDiT in a latent diffusion setup to increase efficiency further and achieve multi-megapixel image resolutions, or apply orthogonal tricks such as self-conditioning (Jabri et al., 2023) or progressive training (Sauer et al., 2022) to improve the quality of generated samples further.

Our large-scale ImageNet experiment (see Section 5.4) shows promise, competing with many state-of-the-art architectures. Future work could realize the potential of ⧗ HDiT with hyperparameter tuning, architecture scaling, and recent practices (Karras et al., 2023).

Our architecture with local attention blocks could enable efficient diffusion superresolution and diffusion VAE feature decoding models: if all levels are set to perform local attention only (global attention blocks should not be necessary as the global structure is already present in the samples for these applications), one can train efficient transformer-based models that can scale to arbitrary resolutions.

Impact Statement

This work aims to improve the capabilities of diffusion models by enabling the training of high-resolution pixel-space transformer-based diffusion models. While many other high-resolution diffusion models exist already, the majority do not operate in pixel-space. Operating in pixel-space potentially enables substantially higher-quality image editing and controllable generation capabilities as downstream tasks. Especially in the context of image editing, capable image synthesis models such as Stable Diffusion (Rombach et al., 2022) have been found to carry risks of generating harmful or deceptive content. In general, progress in high-resolution image synthesis contributes to the production of believable disinformation and could worsen society's ability to trust the authenticity of content. Whilst our method improves the efficiency of transformer-based diffusion models, it remains the case that training and inferencing of diffusion models is energy-intensive, potentially contributing to wider issues such as climate change.

Acknowledgements

We thank Tao Hu for his extensive input and guidance, and *uptightmoose* for their input during the paper writing process. We also thank the reviewers for their useful suggestions. We gratefully acknowledge Jenia Jitsev for his advice on scaling law experiments and support for final set of experiments in the revision, and LAION for access to compute budgets granted by Gauss Centre for Supercomputing e.V. and by the John von Neumann Institute for Computing (NIC) on the supercomputers JUWELS Booster and JURECA at Jülich Supercomputing Centre (JSC). ES gratefully acknowledges Stability AI for resources to conduct experiments.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., Karras, T., and Liu, M.-Y. eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers, 2023.

Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., and Zhu, J. All are Worth Words: A ViT Backbone for Diffusion Models. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023a.

Bao, F., Nie, S., Xue, K., Li, C., Pu, S., Wang, Y., Yue, G., Cao, Y., Su, H., and Zhu, J. One Transformer Fits All Distributions in Multi-Modal Diffusion at Scale. In International Conference on Machine Learning (ICML). JMLR.org, 2023b.

Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao, Y., and Ramesh, A. Improving Image Generation with Better Captions. Technical report, 2023.

Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

Cao, H., Wang, J., Ren, T., Qi, X., Chen, Y., Yao, Y., and Zhang, L. Exploring Vision Transformers as Diffusion Learners, 2022.

Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., and Li, Z. PixArt-α: Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis, 2023a.

Chen, S., Xu, M., Ren, J., Cong, Y., He, S., Xie, Y., Sinha, A., Luo, P., Xiang, T., and Perez-Rua, J.-M. GenTron: Delving Deep into Diffusion Transformers for Image and Video Generation, 2023b.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. PaLM: Scaling Language Modeling with Pathways. Journal of Machine Learning Research (JMLR), 2023.

Dai, X., Hou, J., Ma, C.-Y., Tsai, S., Wang, J., Wang, R., Zhang, P., Vandenhende, S., Wang, X., Dubey, A., Yu, M., Kadian, A., Radenovic, F., Mahajan, D., Li, K., Zhao, Y., Petrovic, V., Singh, M. K., Motwani, S., Wen, Y., Song, Y., Sumbaly, R., Ramanathan, V., He, Z., Vajda, P., and Parikh, D. Emu: Enhancing Image Generation Models Using Photogenic Needles in a Haystack, 2023.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A., Caron, M., Geirhos, R., Alabdulmohsin, I., Jenatton, R., Beyer, L., Tschannen, M., Arnab, A., Wang, X., Riquelme, C., Minderer, M., Puigcerver, J., Evci, U., Kumar, M., Van Steenkiste, S., Elsayed, G. F., Mahendran, A., Yu, F., Oliver, A., Huot, F., Bastings, J., Collier, M. P., Gritsenko, A. A., Birodkar, V., Vasconcelos, C., Tay, Y., Mensink, T., Kolesnikov, A., Pavetić, F., Tran, D., Kipf, T., Lučić, M., Zhai, X., Keysers, D., Harmsen, J., and Houlsby, N. Scaling Vision Transformers to 22 Billion Parameters. In International Conference on Machine Learning (ICML). JMLR.org, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

Dhariwal, P. and Nichol, A. Q. Diffusion Models Beat GANs on Image Synthesis. In Conference on Neural Information Processing Systems (NeurIPS), 2021.

Fedus, W., Zoph, B., and Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. Journal of Machine Learning Research (JMLR), 2022.

Fischer, J. S., Gui, M., Ma, P., Stracke, N., Baumann, S. A., and Ommer, B. Boosting Latent Diffusion with Flow Matching, 2023.

Gao, S., Zhou, P., Cheng, M.-M., and Yan, S. Masked Diffusion Transformer is a Strong Image Synthesizer. In IEEE/CVF International Conference on Computer Vision (ICCV), October 2023.

Gu, J., Zhai, S., Zhang, Y., Susskind, J., and Jaitly, N. Matryoshka Diffusion Models, 2023.

Hang, T., Gu, S., Li, C., Bao, J., Chen, D., Hu, H., Geng, X., and Guo, B. Efficient Diffusion Training via Min-SNR Weighting Strategy. In IEEE/CVF International Conference on Computer Vision (ICCV), October 2023.

Hassani, A., Walton, S., Li, J., Li, S., and Shi, H. Neighborhood Attention Transformer. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2023.

Henry, A., Dachapally, P. R., Pawar, S. S., and Chen, Y. Query-key normalization for transformers. In Cohn, T., He, Y., and Liu, Y. (eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, November 2020.

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 Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), Conference on Neural Information Processing Systems (NeurIPS), 2017.

Ho, J. and Salimans, T. Classifier-Free Diffusion Guidance. In NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, 2021.

Ho, J., Kalchbrenner, N., Weissenborn, D., and Salimans, T. Axial Attention in Multidimensional Transformers, 2019.

Ho, J., Jain, A., and Abbeel, P. Denoising Diffusion Probabilistic Models. In Conference on Neural Information Processing Systems (NeurIPS), 2020.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded Diffusion Models for High Fidelity Image Generation, 2021.

Hoogeboom, E., Heek, J., and Salimans, T. Simple Diffusion: End-to-End Diffusion for High Resolution Images. In International Conference on Machine Learning (ICML). JMLR.org, 2023.

Huang, X. and Belongie, S. Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization. In IEEE/CVF International Conference on Computer Vision (ICCV), Oct 2017.

Jabri, A., Fleet, D., and Chen, T. Scalable Adaptive Computation for Iterative Generation, 2023.

Jing, X., Chang, Y., Yang, Z., Xie, J., Triantafyllopoulos, A., and Schuller, B. W. U-DiT TTS: U-Diffusion Vision Transformer for Text-to-Speech, 2023.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. Training Generative Adversarial Networks with Limited Data. In Conference on Neural Information Processing Systems (NeurIPS), 2020a.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020b.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-Free Generative Adversarial Networks. In Conference on Neural Information Processing Systems (NeurIPS), 2021a.

Karras, T., Laine, S., and Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2021b.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the Design Space of Diffusion-Based Generative Models. In Conference on Neural Information Processing Systems (NeurIPS), 2022.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and Improving the Training Dynamics of Diffusion Models. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

Kingma, D. P. and Gao, R. Understanding Diffusion Objectives as the ELBO with Simple Data Augmentation, 2023.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. DiffWave: A Versatile Diffusion Model for Audio Synthesis. In International Conference on Learning Representations (ICLR), 2021.

Kynkäänniemi, T., Karras, T., Aittala, M., Aila, T., and Lehtinen, J. The Role of ImageNet Classes in Fréchet Inception Distance. In International Conference on Learning Representations (ICLR), 2023. URL https://openreview.net/forum?id=4oXTQ6m_ws8.

Li, R., Li, W., Yang, Y., Wei, H., Jiang, J., and Bai, Q. Swinv2-Imagen: Hierarchical Vision Transformer Diffusion Models for Text-to-Image Generation, 2022.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., and Guo, B. Swin Transformer V2: Scaling Up Capacity and Resolution. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022a.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., and Guo, B. SWin Transformer v2, 2022b. URL https://github.com/microsoft/Swin-Transformer/blob/2cb103f2de145ff43bb9f6fc2ae8800c24/models/swin_transformer_v2.py#L156.

Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. In International Conference on Learning Representations (ICLR), 2019.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. 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.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models, 2023.

Nawrot, P., Tworkowski, S., Tyrolski, M., Kaiser, L., Wu, Y., Szegedy, C., and Michalewski, H. Hierarchical Transformers Are More Efficient Language Models. In Findings of the Association for Computational Linguistics: NAACL 2022, July 2022.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In International Conference on Machine Learning (ICML). PMLR, 2021.

OpenAI. GPT-4 Technical Report. Technical report, 2023.

Peebles, W. and Xie, S. Scalable Diffusion Models with Transformers. In IEEE/CVF International Conference on Computer Vision (ICCV), October 2023.

Piergiovanni, A., Kuo, W., and Angelova, A. Rethinking Video ViTs: Sparse Video Tubes for Joint Image and Video Learning. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-Resolution Image Synthesis With Latent Diffusion Models. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Gontijo-Lopes, R., Ayan, B. K., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), Conference on Neural Information Processing Systems (NeurIPS), 2022.

Saremi, S. and Sejnowski, T. J. Hierarchical model of natural images and the origin of scale invariance. Proceedings of the National Academy of Sciences, 110 (8):3071–3076, February 2013. ISSN 1091-6490. doi: 10.1073/pnas.1222618110.

Sauer, A., Schwarz, K., and Geiger, A. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. In ACM SIGGRAPH 2022 Conference Proceedings. Association for Computing Machinery, 2022.

Shazeer, N. GLU Variants Improve Transformer, 2020.

Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), jun 2016.

Shonenkov, A., Konstantinov, M., Bakshandaeva, D., Schuhmann, C., Ivanova, K., and Klokova, N. Deepfloyd if, 2023.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In International Conference on Learning Representations (ICLR), 2021.

Stein, G., Cresswell, J. C., Hosseinzadeh, R., Sui, Y., Ross, B. L., Villecroze, V., Liu, Z., Caterini, A. L., Taylor, J. E. T., and Loaiza-Ganem, G. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models, 2023.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding, 2022.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. LLaMA: Open and Efficient Foundation Language Models. Technical report, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is All you Need. In Conference on Neural Information Processing Systems (NeurIPS), 2017.

Wang, P. Flash Cosine Similarity Attention, 2022. URL https://github.com/lucidrains/flash-cosine-sim-attention/tree/6f17f29a979a8bcab2479c65b7740523.

Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., and Li, H. Uformer: A General U-Shaped Transformer for Image Restoration. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

Yan, J. N., Gu, J., and Rush, A. M. Diffusion Models Without Attention, 2023.

Yang, G. and Hu, E. J. Feature learning in infinite-width neural networks, 2022.

Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer, 2022a.

Yang, X., Shih, S.-M., Fu, Y., Zhao, X., and Ji, S. Your ViT is Secretly a Hybrid Discriminative-Generative Diffusion Model, 2022b.

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. CoCa: Contrastive Captioners are Image-Text Foundation Models. Transactions on Machine Learning Research (TMLR), 2022.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., and Yang, M.-H. Restormer: Efficient Transformer for High-Resolution Image Restoration. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

Zhang, B. and Sennrich, R. Root Mean Square Layer Normalization. In Advances in Neural Information Processing Systems 32, Vancouver, Canada, 2019.

Zhang, B., Gu, S., Zhang, B., Bao, J., Chen, D., Wen, F., Wang, Y., and Guo, B. StyleSwin: Transformer-Based GAN for High-Resolution Image Generation. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11304–11314, June 2022a.

Zhang, Y., Qin, J., Park, D. S., Han, W., Chiu, C.-C., Pang, R., Le, Q. V., and Wu, Y. Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition, 2022b.

Zhao, L., Zhang, Z., Chen, T., Metaxas, D., and Zhang, H. Improved Transformer for High-Resolution GANs. In Conference on Neural Information Processing Systems (NeurIPS), 2021.

Zheng, H., Nie, W., Vahdat, A., and Anandkumar, A. Fast Training of Diffusion Models with Masked Transformers, 2023.

Zong, Z., Song, G., and Liu, Y. DETRs with Collaborative Hybrid Assignments Training. In IEEE/CVF International Conference on Computer Vision (ICCV), 2022.

A. Computational Complexity of HDiT

In a traditional vision transformer, including those for diffusion models (Peebles & Xie, 2023; Bao et al., 2023a), the asymptotic computational complexity with regard to image size is dominated by the self-attention mechanism, which scales as $\mathcal{O}(n^2 d)$ with token/pixel count n and embedding dimension d. The feedforward blocks and the attention projection heads, in turn, scale as $\mathcal{O}(nd^2)$.

For our Hourglass Diffusion Transformer architecture, we adjust the architecture for different target resolutions, similarly to previous approaches used with U-Nets (Ronneberger et al., 2015). Our architecture is divided into multiple hierarchical levels, where the outermost level operates at full patch resolution, and each additional level operates at half of the spatial resolution per axis. For simplicity, we will first cover the cost at square resolutions of powers of two.

When designing the architecture for a specific resolution, we start with a dataset-dependent *core* architecture, which, for natural images, typically includes one or two global-attention hierarchy levels that operate at 16^2 or 16^2 and 32^2, respectively. Around that are a number of local attention levels. As this core only operates on a fixed resolution, it does not influence the asymptotic computational complexity of the overall model.

Asymptotic Complexity Scaling When this architecture is adapted to a higher resolution, additional local attention levels with shared parameters are added to keep the innermost level operating at 16^2. This means that the number of levels in our hierarchy scales with the number of image tokens as $\mathcal{O}(\log(n))$. While this might intuitively lead one to the conclusion of the overall complexity being $\mathcal{O}(n \log(n) d)$, as local attention layers' complexity is $\mathcal{O}(nd)$, the reduction in resolution at each level in the hierarchy has to be considered: due to the spatial downsampling, the number of tokens decreases by a factor of four at every level in the hierarchy, making the cost of the self-attention – the only part of our model whose complexity does not scale linearly with token count – of the additional levels

$$\sum_{l=1}^{\log_4(n)-\log_4(\text{res}_{\text{core}})} \frac{nd}{4^{l-1}}.$$

Factoring out n and defining $m = l - 1$ yields

$$n \cdot \sum_{m=0}^{\log_4(n)-\log_4(\text{res}_{\text{core}})-1} d \cdot \left(\frac{1}{4}\right)^m,$$

a (cut-off) geometric series with a common ratio of less than one, which means that, as the geometric series converges, it does not affect the asymptotic complexity, making the cumulative complexity of the local self-attention of the additional levels $\mathcal{O}(nd)$. Thus, as no other parts scale worse than $\mathcal{O}(nd)$ either, the overall complexity of the Hourglass Diffusion Transformer architecture, as the target resolution is increased, is $\mathcal{O}(nd)$.

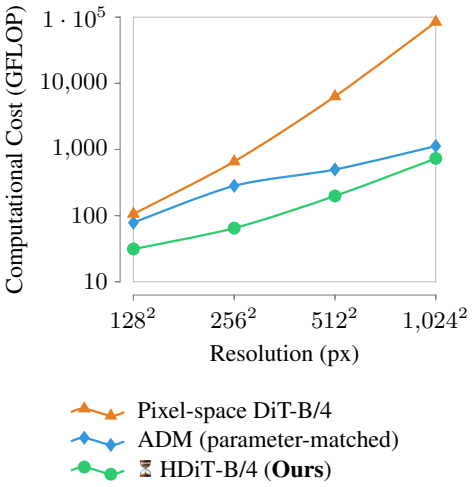


Figure 8: Scaling of computational cost w.r.t. target resolution of our ⧗ HDiT-B/4 model vs. DiT-B/4 (Peebles & Xie, 2023) and ADM (Dhariwal & Nichol, 2021).

Local Complexity Scaling at Arbitrary Resolutions When the target resolution is increased by a factor smaller than a power of two per axis, the architecture is not adapted. This means that, for these intermediate resolutions, a different scaling behavior prevails. Here, the cost of the local attention levels, whose number does not change in this case, scales with $\mathcal{O}(nd)$ as before, but the global attention levels incur a quadratic increase in cost with the resolution. As the resolution is increased further, however, new levels are added, which reduce the resolution the global attention blocks operate at to their original values, and retaining the overall asymptotic scaling behavior of $\mathcal{O}(nd)$.

FLOP Comparison with DiT and Diffusion U-Nets While the asymptotic computational cost is important, it only describes how computational cost scales with resolution (in the theoretical limit). For practical purposes, it is important that the theoretical improvement from $\mathcal{O}(n^2 d)$ to $\mathcal{O}(nd)$ from DiT to ⧗ HDiT also results in lower FLOPs. To investigate this, we calculate the practical FLOPs for a parameter-matched pixel-space DiT and HDiT at various resolutions, which we show in Figure 8. We find that the theoretical improvements translate to real-world improvements, with HDiT already being more than 10 times more efficient at 256^2 resolution, which further increases to a more than 100 times improvement for 1024^2. We also investigate a representative standard CNN-based diffusion U-Net (Dhariwal & Nichol, 2021). Here, we also find substantial performance gains of about 10 times at low resolutions, although the gap narrows at higher resolutions.

B. Pixel-space vs. Latent Diffusion Models

Extending upon the brief motivation presented in the introduction, we compare the advantages and disadvantages of pixel-space and latent (Rombach et al., 2022) diffusion models.

B.1. Advantages of Pixel-space over Latent Diffusion

Several factors motivate the exploration of pixel-based alternatives:

1. Architectural Simplicity and Latent Space Limitations: Pixel-based models circumvent the need for complex latent space engineering, simplifying model architecture. Relying on a learned latent space introduces limitations tied to the VAE's representational capacity.

2. Quality Constraints and High-Frequency Information Loss: VAE-based diffusion models are inherently bounded by the reconstruction quality of the underlying VAE. Critically, VAEs are prone to losing high-frequency image details, hindering the generation of sharp and realistic images. We see evidence of this when roundtripping one of the generated images from our 557M ImageNet-256^2 model from Figure 1 through the VAE used by DiT (Peebles & Xie, 2023) in Figure 2

3. Fidelity Limitations for Image Manipulation: Faithful image reconstruction is crucial for downstream tasks like editing and transformation. VAEs often struggle with faithful reconstruction, limiting their applicability in these domains.

4. Challenges with Dynamic Thresholding and Intermediate Step Visualization: Integrating advanced sampling techniques like dynamic thresholding, as proposed in the DPM Solver (Lu et al., 2022) literature, remains challenging within the latent space framework. Similarly, visualizing intermediate generation steps requires computationally expensive decoding, hindering iterative design processes.

5. Limited Compatibility with Classifier Guidance (Dhariwal & Nichol, 2021): Leveraging classifier guidance, a powerful technique for controlling image generation, proves difficult with latent space models. This difficulty arises from the mismatch between the pixel-space nature of most classifiers and the latent space representation of the diffusion model.

6. Empirical Evidence in Text-to-3D Synthesis: Recent work in text-to-3D generation has demonstrated superior performance with pixel-based diffusion models, highlighting their potential for high-fidelity synthesis (Shonenkov et al., 2023).

7. Information Loss and Inpainting Challenges: The inherent information compression within the VAE latent space can negatively impact inpainting tasks. Specifically, it can lead to undesirable leakage of information from the surrounding regions into the inpainted area.

B.2. Advantages of Latent Diffusion Models

Latent diffusion models (Rombach et al., 2022) operate on the in the latent space of a variational auto-encoder. This allows for a substantial reduction in the spatial resolution, leading to a signficant computational reduction. The aforemenetioned reduction allows for usage of what would otherwise be computational infeasible choices, such as transformer models (Vaswani et al., 2017). The VAE inherently constrains the diffusion process to a manifold of plausible images. This effectively raises the lower bound on the average quality of generated images, leading to more consistent quality of images.

C. Soft-Min-SNR Loss Weighting

Min-SNR loss weighting (Hang et al., 2023) is a recently introduced training loss weighting scheme that improves diffusion model training. It adapts the SNR weighting scheme (for image data scaled to $\mathbf{x} \in [-1, 1]^{h \times w \times c}$)

$$w_{\text{SNR}}(\sigma) = \frac{1}{\sigma^2} \tag{3}$$

by clipping it at an SNR of $\gamma = 5$:

$$w_{\text{Min-SNR}}(\sigma) = \min\left\{\frac{1}{\sigma^2}, \gamma\right\}. \tag{4}$$

We utilize a slightly modified version that smoothes out the transition between the normal SNR weighting and the clipped section:

$$w_{\text{Soft-Min-SNR}}(\sigma) = \frac{1}{\sigma^2 + \gamma^{-1}}. \tag{5}$$

For $\sigma \ll \gamma$ and $\sigma \gg \gamma$, this matches Min-SNR, while providing a smooth transition between both sections.

In practice, we also change the hyperparameter γ from $\gamma = 5$ to $\gamma = 4$.

Plotting the resulting loss weight for both min-snr and our soft-min-snr as shown in Figure 9 shows that our loss weighting is identical to min-snr, except for the transition, where it is significantly smoother. An ablation of our soft-min-snr compared to min-snr also shows that our loss weighting scheme leads to an improved FID score for our model, as shown in Table 5, steps **D** (SNR), **E2** (Min-SNR, $\gamma = 5$), **E3** (Min-SNR, $\gamma = 4$), **E** (Soft-Min-SNR, $\gamma = 4$).

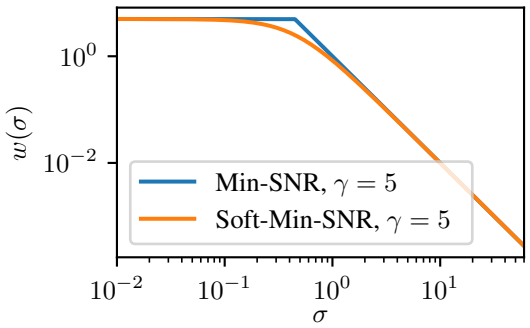


Figure 9: The resulting loss weighting over σ for our soft-min-snr weighting (orange) and min-snr weighting (blue) with $\gamma = 5$.

D. Additional Experimental Results

This section presents results for auxiliary experiments that provide additional context for the experiments presented in the main body of the paper.

D.1. Additional Ablation Results

In Table 5, we present additional results for the main ablation study initially presented in Section 5.2.

Loss Weighting In **E2** and **E3**, we apply Min-SNR (Hang et al., 2023) loss weighting with the original hyperparameter $\gamma = 5$ and the value $\gamma = 4$ used for our Soft-Min-SNR. This shows that, in our setting, both the change of γ and the smoother loss weighting help improve FID but that the smoothing plays a substantially larger role.

Subtractive Ablations vs. DiT Extending our ablation in Section 5.2, we also perform two subtractive ablations investigating the norm and activation choice in combination, whose results are shown in Table 5. Ablation step **G** takes the full model but replaces the adaptive RMSNorm (Zhang & Sennrich, 2019) with an adaptive layer norm (Ba et al., 2016) as used by DiT (Peebles & Xie, 2023). Despite offering twice as many degrees of freedom due to predicting a shift in addition to the scale, we see no significant change in FID. Completely reverting to DiT-style blocks by changing GeGLU to GELU and adding an output gate controlled via the mapping network in step **H** results in a worse FID, corroborating the results from the original ablation step **C**, even in combination with the different norm.

Additional Baselines In Section 5.2, we only present **R1**, **R3**, and **R4** for simplicity. To evaluate the influence of our trainer and our loss weighting scheme, we also add an intermediate step, **R2**. This step wraps the official implementation of DiT-B/4 and adapts it to our codebase and trainer.[3] This leads to a substantial reduction in FID compared to the original trainer, showing that it is important that the training setting matches the architecture. **R3** replaces the wrapped DiT model with a hyperparameter-matched single-level version of ablation step **A**, matching the performance of the original DiT trained with the original codebase. On top of this setup, we also add soft-min-snr loss weighting to **R4** as in ablation step **E** to enable a fair comparison with our final model.

Table 5: Additional ablation results on RGB ImageNet-128^2. Results already presented in Table 1 are presented in gray font as a reference.

Configuration		FID↓
Baselines		
R1	DiT-B/4 (Peebles & Xie, 2023)	42.03
R2	**R1** + Our Trainer	69.86
R3	**R2** + Our Basic Blocks & Mapping Network	42.49
R4	**R3** + Soft-Min-SNR	30.71
Ablation Steps		
A	Global Attention Diffusion Hourglass (Section 4.1)	50.76
B1	**A** + Swin Attn. (Liu et al., 2021)	55.93
B2	**A** + Neighborhood Attn. (Hassani et al., 2023)	51.07
C	**B2** + GeGLU (Shazeer, 2020)	44.36
D	**C** + Axial RoPE (Section 4.2)	41.41
E	**D** + Soft-Min-SNR (Appendix **C**)	27.74
E2	**D** + Min-SNR (Hang et al., 2023) ($\gamma = 5$)	36.65
E3	**D** + Min-SNR (Hang et al., 2023) ($\gamma = 4$)	35.62
F1	**E** + Concatenation Skip	33.75
F2	**E** + Additive Skip	28.37
G	**E** + AdaRMSNorm → AdaLN	27.69
H	**G** + GeGLU → GeLU, DiT-style Output Gate	30.66

D.2. Effect of CFG for our 557M ImageNet-256^2 Model

In addition to the analyses in Section 5.4, which do not use classifier-free guidance (CFG) (Ho & Salimans, 2021), we also analyze the FID-IS-tradeoff for difference guidance scales w_{cfg} (we follow the guidance scale formulation used in (Saharia et al., 2022), where $w_{cfg} = 1$ corresponds to no classifier-free guidance being applied). The resulting curve is shown in Figure 10, with the lowest FID of 3.21 being achieved around $w_{cfg} = 1.3$, with a corresponding IS of 220.6.

[3]The pixel-space DiT **R2** was trained with an identical setup to the rest of our ablations except for the optimizer parameters: we initially tried training this model with our optimizer parameters but found it to both be unstable and worse than with the original parameters, so we used the original parameters from (Peebles & Xie, 2023) for the comparison.

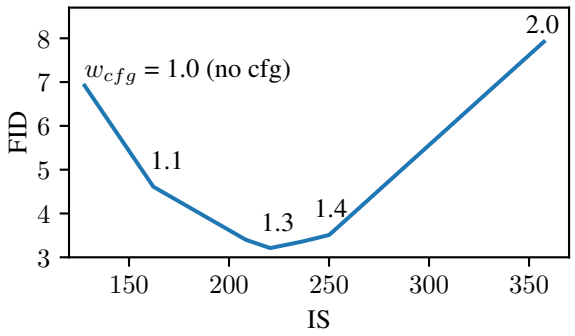


Figure 10: Inception Score vs. Fréchet Inception Distance at different classifier-free guidance weight scales (1 = no guidance) for our 557M ImageNet-256^2 model.

D.3. Scaling Behavior

To analyze the model's scaling behavior, we train a set of 9 models with varying model & patch sizes. The hyperparameters are taken from our main 557M run and all models are trained in exactly the same setting for 1M steps. The shared hyperparameters are shown in Table 8a, and the individual run-specific details are shown in Table 8b. We show qualitative results in Table 6.

We show quantitative FID evaluations of the models in Table 6. Curiously, the most compute-intensive model (557M, patch size 4^2) not only underperforms its smaller peers, but *significantly* underperforms the identically-configured model from our large-scale ImageNet experiment (which, following a longer 2.2M steps of training, achieved an FID of 6.92 [3.21 with CFG]). We attribute this discrepancy to a suboptimal choice of hyperparameters, imposing a fixed learning rate (5e-4) and batch size (256) across all experiments. Our large-scale ImageNet experiment (Section 5.4) mitigates this high learning rate by employing larger batch sizes later in training (see Table 7). The notion of larger models' preferring larger batch sizes / lower learning rates, is corroborated by the line of work investigating μP-Parametrization (Yang & Hu, 2022; Yang et al., 2022a), which found that, using standard parameterizations (as we did for ⧗ HDiT), a model's optimal learning rate decreases as size increases. Our learning rate of 5e-4 seems to work well for small models but seems to be too high for the larger models. Future work could change the parametrization to μP to enable using the same learning rate for all scales and revisit this experiment.

Qualitatively, we find that patch sizes as large as 16^2 and 8^2 are too ambitious for the transformer sizes ($\sim$100–500M) over which we ablated. Only the largest transformer (557M) achieved consistent coherence, and even then, only at the smallest patch size 4^2. Studying the examplar sample grids in Figure 13: we see that generation of round tennis balls or pumpkins succeeds at 4^2 patch size for all transformer sizes, with some success also at 8^2 patch size for the largest transformer. Balloons are coherent at 4^2 patch size only, from the largest transformer and tenuously from the smallest. French loaves are coherent for the largest transformer only, at patch size 4^2 (and tenuously 8^2, notwithstanding questionable background forms), with texture best at 4^2 (and arguably gummy at 8^2). Ultimately, the 8^2 patch size had too many coherence failures to recommend it, with even the largest transformer suffering discontinuous balloons, ill-defined cat eyelids, hyperbolic fox ears, vases with apertures, amorphous bullfrogs, wolf eye asymmetry, unbalanced poodles, and lemons eaten by their own leaves. Likewise, the medium transformer struggles at the lowest patch size 4^2, exhibiting octopoid loaves, indistinct fox bodies, asymmetric cats and wolves, and fissured tennis balls. Coherence worsened further as model size decreased or as patch size increased.

Table 6: Quantitative evaluation of our ImageNet-256^2 Transformer Size vs Patch Size sweep, illustrated in Figure 13.

Parameter	Small	Medium	Large
Patch Size 16^2			
Parameters	116M	267M	507M
FID↓	90.6	51.8	37.6
Patch Size 8^2			
Parameters	134M	294M	547M
FID↓	50.3	30.9	33.6
Patch Size 4^4			
Parameters	139M	302M	557M
FID↓	21.6	24.0	29.3

E. Implementation Details

This section aims to answer potential questions about implementation details of ⧗ HDiT for convenience. For further details, we refer to the reference implementation.

E.1. Scaled Cosine Similarity Attention

For the attention mechanism, we use a slight variation of the cosine similarity-based attention introduced in (Liu et al., 2022a) they dub *Scaled Cosine Attention* (a similar approach has also recently been used in (Karras et al., 2023)): instead of computing the self-attention as

$$\mathrm{SA}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_{\text{head}}}}\right)V, \tag{6}$$

they compute it as

$$\mathrm{SCA}(Q, K, V) = \mathrm{softmax}\left(\frac{\mathrm{sim}_{\cos}(Q, K)}{\tau} + B_{ij}\right)V, \quad (7)$$

with τ being a per-head per-layer learnable scalar, and B_{ij} being the relative positional bias between pixel i and j (which we do not use in our models). In practice, they parametrize τ based on a learnable parameter θ in the following way (Liu et al., 2022b):

$$\frac{1}{\tau} = \exp\left(\min\left\{\theta, \log\frac{1}{0.01}\right\}\right), \quad (8)$$

with θ being initialized to $\theta = \log 10$.

Improving Scale Learning Stability We find that their parametrization of τ causes the learned scales to vary significantly during training, necessitating the clamping to a maximum value of 100 before exponentiation to prevent destabilization of the training. In this setting, we find that a significant number of scale factors τ reach this maximum value and values below 1 during our trainings. We speculate that this instability might be the cause of the behaviour observed in (Wang, 2022), where using scaled cosine similarity attention was detrimental to the performance of generative models. To alleviate this problem, we find simply learning τ directly, as done for normal attention in (Henry et al., 2020), prevents this large variance of its values in our models, with our converged models' scale typically reaching a range between 5 and 50.

E.2. Axial RoPE

We extend rotary positional embeddings (Su et al., 2022) to 2D image data. We split the encoding to operate independently along each axis, applying RoPE for each spatial axis to half of the query and key each. Empirically, we find that applying this embedding scheme to only half of key & query and leaving the other half unmodified (see Figure 11 for an illustration) results in better performance than applying it for the full key & query.

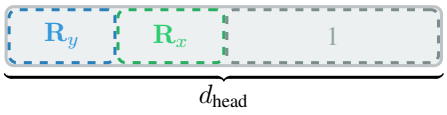


Figure 11: Illustration of our 2D axial RoPE embedding scheme. The rotation for the vertical position $\mathbf{R}_y$ and horizontal position $\mathbf{R}_x$ are applied to one quarter of the key/query each, while the rest is left unaffected.

E.3. Conditioning

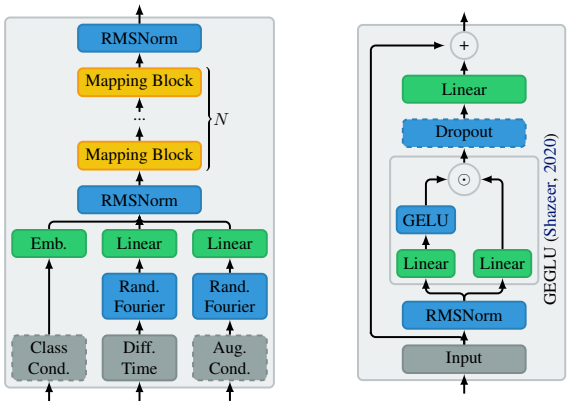
Adaptive RMSNorm Following common practice, we implement conditioning using adaptive norms (Huang & Belongie, 2017), where we apply a standard RMSNorm (Zhang & Sennrich, 2019)

$$x_{i,\text{scaled}} = \frac{x_i}{\mathrm{RMS}(\mathbf{x})} \cdot g_i, \text{with } \mathrm{RMS}(\mathbf{x}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}, \quad (9)$$

with $\mathbf{g}$ being predicted from the mapping network based on the conditioning $\mathbf{c}$ instead of being a learned vector as $\mathbf{g} = 1 + \mathrm{mapping}(\mathbf{c})$. At initialization, the final linear projection is initialized to zero, making

Mapping Network The prediction of the RMSNorm scales is implemented via a mapping network that takes the diffusion timestep, the class conditioning, and, optionally, augmentation information to prevent augmentation leakage (Karras et al., 2020a).

The mapping network consists of N blocks that process the conditioning information. The blocks' architecture is almost identical to our pointwise FFN block (see Figure 5). For the initial embedding, we use a standard learnable embedding for the class conditioning, and random fourier features (following (Karras et al., 2022)) followed by linear projections for the diffusion timestep and augmentation conditioning. An overview of the network and block structure is given in Figure 12.



(a) ⧗ HDiT mapping network. (b) ⧗ HDiT mapping block.

Figure 12: An overview of our mapping network architecture.

This general conditioning embedding is then passed to each block in the main network, where it is projected locally to obtain the relevant information for that block and obtain the feature scales.

E.4. Token Merging & Splitting

For token merging and splitting inside our architecture, we follow a standard Pixel-Shuffle (Shi et al., 2016)-based approach. Token merging is implemented as a reshaping of the tensor from $B \times H \times W \times C$ to $B \times \frac{H}{M} \times \frac{W}{M} \times CM^2$

(*Pixel-UnShuffle*), with $M = 2$, followed by a pointwise linear projection to adjust the channel count to the appropriate model width at that level. Similarly, token splitting is implemented as a pointwise linear projection, bringing the channel count from the model width to CM^2, followed by a reshaping $B \times H \times W \times CM^2$ to $B \times HM \times WM \times C$ (*Pixel-Shuffle*). This follows various previous implementations such as (Zamir et al., 2022).

F. Experiment Details

We provide an overview of all relevant hyperparameters, training hardware, and time for the experiments presented in this paper in Table 7 and Table 8.

Table 7: Details of our training and inference setup.

Parameter	**ImageNet**-128^2	**FFHQ**-1024^2	**ImageNet**-256^2
Experiment	Ablation **E**[4] (Section 5.2)	High-Res Synthesis (Section 5.3)	Large-Scale (Section 5.4)
Parameters	117M	85M	557M
GFLOP/forward	31	206	198
Training Steps	400k	1M	2.2M
Batch Size	256	256	256+[5]
Precision	bfloat16	bfloat16	bfloat16
Training Hardware	4 A100 80GiB	64 A100 80GiB	8 H100 80GiB
Training Time	15 hours[6]	5 days[6]	7.6 days
Patch Size	4	4	4
Levels (Local + Global Attention)	1 + 1	3 + 2	2 + 1
Depth	[2, 11]	[2, 2, 2, 2, 2]	[2, 2, 16]
Widths	[384, 768]	[128, 256, 384, 768, 1024]	[384, 768, 1536]
Attention Heads (Width / Head Dim)	[6, 12]	[2, 4, 6, 12, 16]	[6, 12, 24]
Attention Head Dim	64	64	64
Neighborhood Kernel Size	7	7	7
Mapping Depth	1	2	2
Mapping Width	768	768	768
Data Sigma	0.5	0.5	0.5
Sigma Range	[1e-3, 1e3]	[1e-3, 1e3]	[1e-3, 1e3]
Sigma Sampling Density	interpolated cosine	interpolated cosine	interpolated cosine
Augmentation Probability	0	0.12	0
Dropout Rate	0	[0, 0, 0, 0, 0.1]	0
Conditioning Dropout Rate	0.1	0.1	0.1
Optimizer	AdamW	AdamW	AdamW
Learning Rate	5e-4	5e-4	5e-4
Betas	[0.9, 0.95]	[0.9, 0.95]	[0.9, 0.95]
Eps	1e-8	1e-8	1e-8
Weight Decay	1e-2	1e-2	1e-2
EMA Decay	0.9999	0.9999	0.9999
Sampler	DPM++(3M) SDE	DPM++(3M) SDE	DPM++(3M) SDE
Sampling Steps	50	50	50

[4]The other ablation steps generally use the same parameters, except for the architectural changes indicated in the experiment description.

[5]We initially trained for 2M steps. We then experimented with progressively increasing the batch size (waiting until the loss plateaued to a new, lower level each time), training at batch size 512 for an additional 50k steps, at batch size 1024 for 100k, and at batch size 2048 for 50k steps.

[6]Wall clock time, including startup, validation, checkpoint saving, etc.

Table 8: Details of our ImageNet-256^2 Transformer Size vs. Patch Size training and inference setup.

(a) Details common to all configs in our Transformer Size vs. Patch Size experiments.

Parameter	
Training Steps	1M
Batch Size	256
Precision	bfloat16
Attention Head Dim	64
Neighborhood Kernel Size[7]	7
Mapping Depth	2
Mapping Width	768
Data Sigma	0.5
Sigma Range	[1e-3, 1e3]
Sigma Sampling Density	interpolated cosine
Augmentation Probability	0
Dropout Rate	0
Conditioning Dropout Rate	0.1
Optimizer	AdamW
Learning Rate	5e-4
Betas	[0.9, 0.95]
Eps	1e-8
Weight Decay	1e-2
EMA Decay	0.9999
Sampler	DPM++(2M) SDE
Sampling Steps	50

(b) Config-specific details of our Transformer Size vs. Patch Size experiments.

Parameter	Small	Medium	Large
Patch Size 16^2			
Parameters	116M	267M	507M
GFLOP/forward	29	68	129
Training Hardware	4 A100 80GiB	4 A100 40GiB	4 A100 40GiB
Training Time[6]	1.1 days	2.5 days	4.4 days
Levels (Local + Global Attention)	0 + 1	0 + 1	0 + 1
Depth	8	12	16
Widths	1024	1280	1536
Attention Heads (Width / Head Dim)	16	20	24
Patch Size 8^2			
Parameters	134M	294M	547M
GFLOP/forward	44	91	163
Training Hardware	4 A100 80GiB	2×4 A100 40GiB	2×4 A100 40GiB
Training Time[6]	2.6 days	2.2 days	3.6 days
Levels (Local + Global Attention)	1 + 1	1 + 1	1 + 1
Depth	[2, 8]	[2, 12]	[2, 16]
Widths	[512, 1024]	[640, 1280]	[768, 1536]
Attention Heads (Width / Head Dim)	[8, 16]	[10, 20]	[12, 24]
Patch Size 4^4			
Parameters	139M	302M	557M
GFLOP/forward	60	115	198
Training Hardware	4xA100 40GiB	2x4xA100 40GiB	2x4xA100 40GiB
Training Time[6]	3.7 days	3.3 days	6.1 days
Levels (Local + Global Attention)	2 + 1	2 + 1	2 + 1
Depth	[2, 2, 8]	[2, 2, 12]	[2, 2, 16]
Widths	[256, 512, 1024]	[320, 640, 1280]	[384, 768, 1536]
Attention Heads (Width / Head Dim)	[4, 8, 16]	[5, 10, 20]	[6, 12, 24]

[7] Transformers with patch size 16^2 did not possess any neighborhood attention levels

G. Scaling Samples

We provide an equivalent of Fig. 7 from DiT (Peebles & Xie, 2023), where samples are generated with fixed random seed across multiple patch sizes and transformer scales, in Figure 13. The quality of generated samples increases with smaller patch sizes and larger transformers, matching the findings for DiT and demonstrating the scalability of ⧗ HDiT.

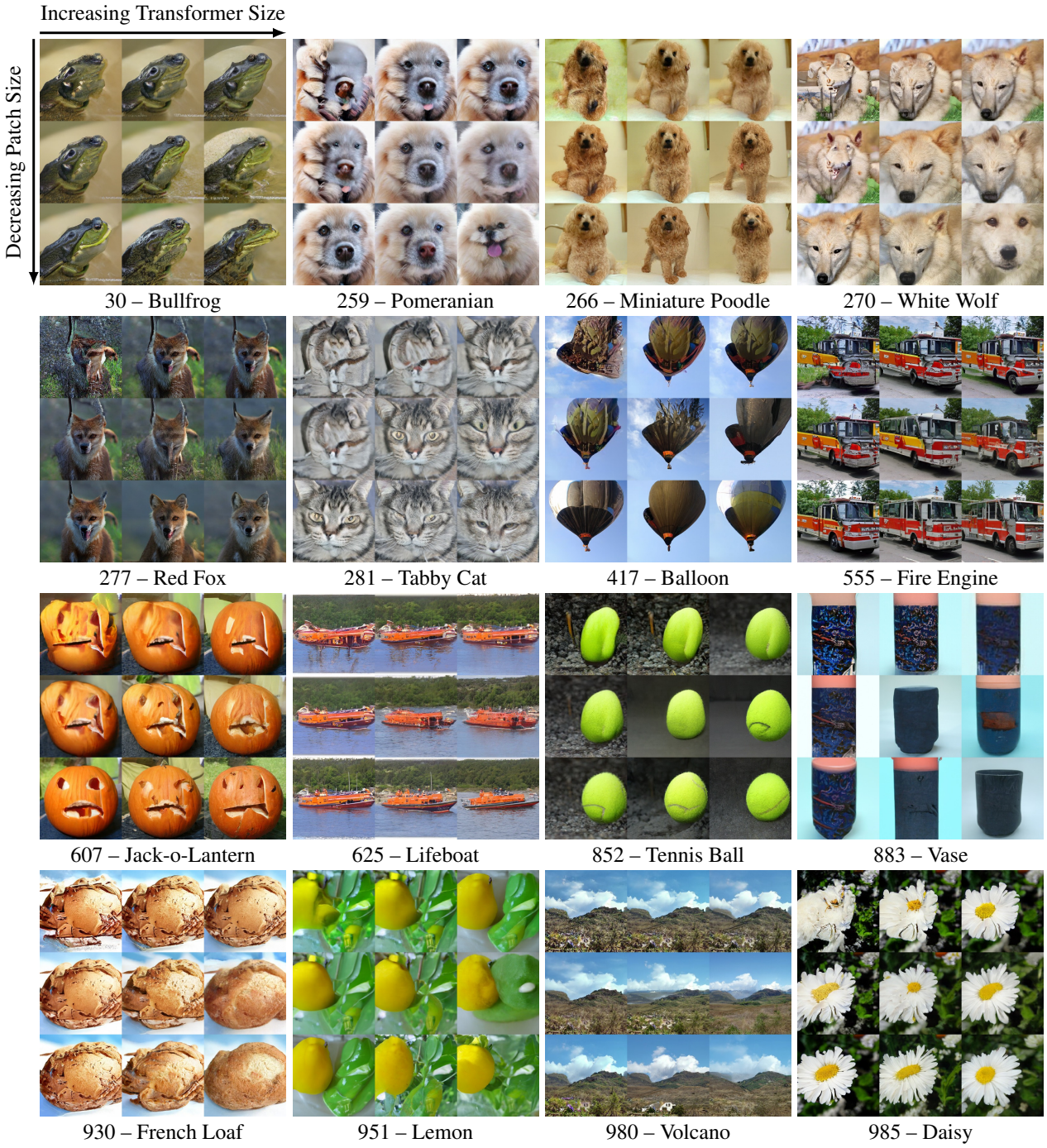


Figure 13: Scaling behaviour of our ⧗ HDiT across different model and patch sizes on pixel-space ImageNet-256^2. All models used to generate samples for this figure have been trained for 1M steps, and samples have been generated without classifier-free guidance. Patch sizes shown are $\{16, 8, 4\}$, transformer sizes approximately double at each step, up to our 557M ImageNet-256^2 model. See Table 8 for detailed hyperparameters.

H. Our FFHQ-1024^2 Samples

Figure 14: **Uncurated** samples from our 85M ⧗ HDiT FFHQ-1024^2 model.

I. FFHQ-1024^2 Reference Samples

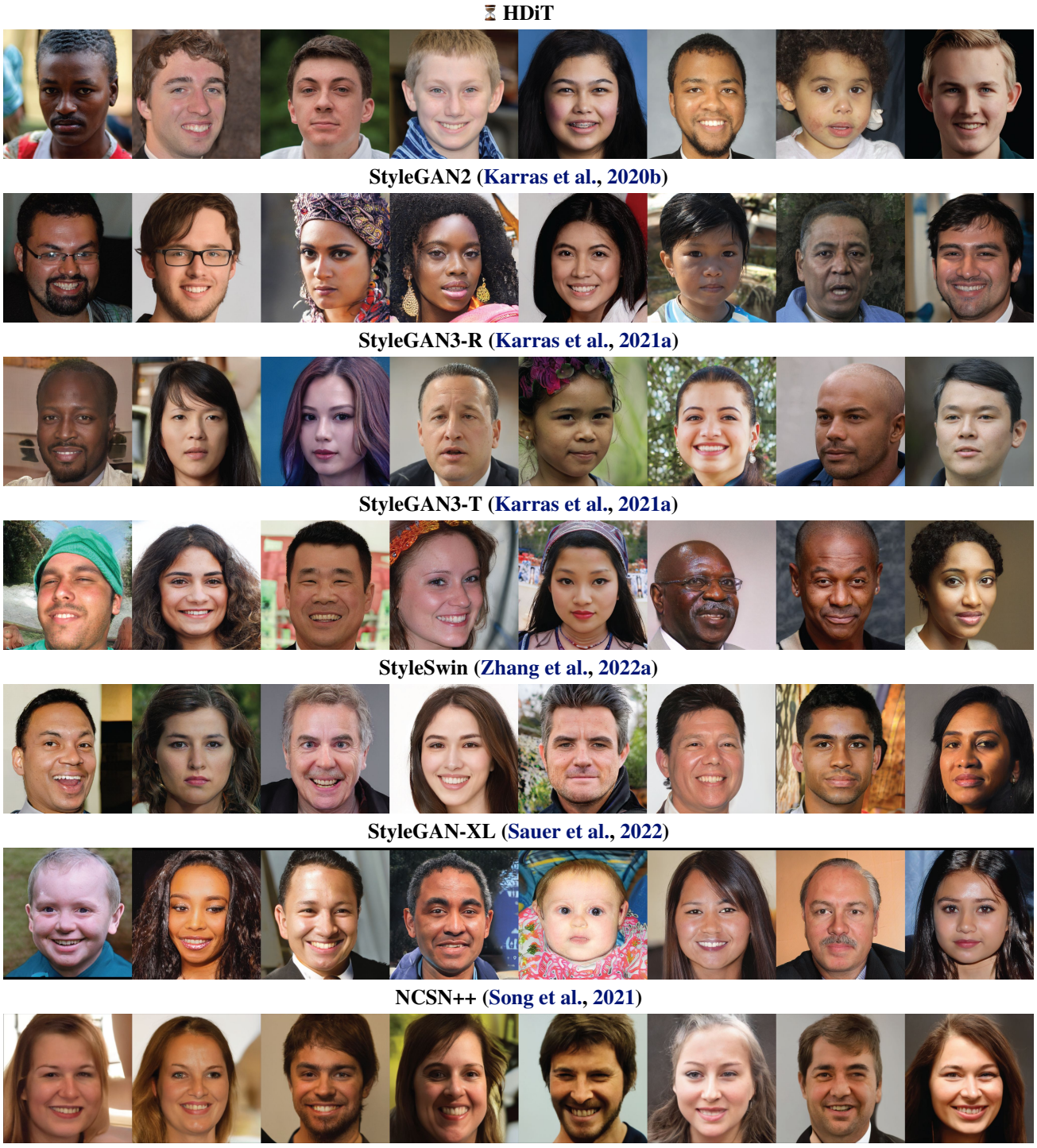


Figure 15: **Curated** FFHQ-1024^2 reference samples from Hourglass, StyleGAN2 (Karras et al., 2020b), StyleGAN3-R (Karras et al., 2021a), StyleGAN3-T (Karras et al., 2021a), StyleSwin (Zhang et al., 2022a), StyleGAN-XL (Sauer et al., 2022), and NCSN++ (Song et al., 2021) models.

J. Our ImageNet-256^2 Samples

Figure 16: **Uncurated** random class-conditional samples from our 557M ⧗ HDiT ImageNet-256^2 model.

Figure 17: More **uncurated** random class-conditional samples from our ⧗ HDiT-557M ImageNet-256^2 model.