

HierarchicalPrune: Position-Aware Compression for Large-Scale Diffusion Models

Young D. Kwon^{1*}, Rui Li^{1*}

Sijia Li², Da Li¹, Sourav Bhattacharya¹, Stylianos I. Venieris¹

¹ Samsung AI Center-Cambridge, UK

²Independent Researcher

{yd.kwon, rui.li, s.venieris}@samsung.com

Abstract

State-of-the-art text-to-image diffusion models (DMs) achieve remarkable quality, yet their massive parameter scale (8-11B) poses significant challenges for inferences on resource-constrained devices. In this paper, we present *HierarchicalPrune*, a novel compression framework grounded in a key observation: DM blocks exhibit distinct functional hierarchies, where early blocks establish semantic structures while later blocks handle texture refinements. *HierarchicalPrune* synergistically combines three techniques: (1) Hierarchical Position Pruning, which identifies and removes less essential later blocks based on position hierarchy; (2) Positional Weight Preservation, which systematically protects early model portions that are essential for semantic structural integrity; and (3) Sensitivity-Guided Distillation, which adjusts knowledge-transfer intensity based on our discovery of block-wise sensitivity variations. As a result, our framework brings billion-scale diffusion models into a range more suitable for on-device inference, while preserving the quality of the output images. Specifically, combined with INT4 weight quantisation, *HierarchicalPrune* achieves 77.5-80.4% memory footprint reduction (e.g., from 15.8 GB to 3.2 GB) and 27.9-38.0% latency reduction, measured on server and consumer grade GPUs, with the minimum drop of 2.6% in GenEval score and 7% in HPSv2 score compared to the original model. Finally, our comprehensive user study with 85 participants demonstrates that *HierarchicalPrune* maintains perceptual quality comparable to the original model while significantly outperforming prior works.

1 Introduction

Diffusion-based text-to-image (T2I) synthesis (Song et al. 2021; Stability AI 2023; Lipman et al. 2023; Karras et al. 2022; Peebles and Xie 2022; Lu et al. 2024) has emerged as a powerful tool for a wide variety of applications, such as the generation of educational content, creative artwork, and UX/UI design prototyping, intensifying demand for deployable billion-parameter diffusion models (DMs). While recent advances, such as Stable Diffusion 3.5 (SD3.5) (Esser et al. 2024) and FLUX (Black Forest Labs 2024), significantly outperform previous generations (SDXL (Podell et al. 2023), SD1.5 (Rombach et al. 2022), and DALLE-2) in image quality and text alignment, they also come with excessive model

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

*Co-first authors

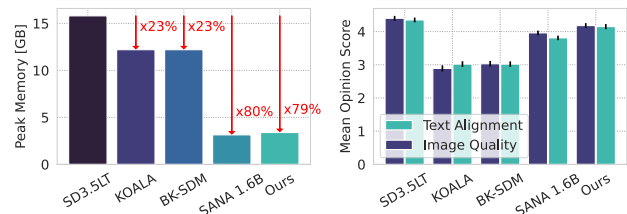


Figure 1: *HierarchicalPrune* achieves 79.5% memory reduction (left) while maintaining image quality. User study with 85 participants (right) demonstrates minimal quality drop (4.8-5.3%) with 95% confidence intervals, contrary to the excessive degradation (11.1-52.2%) of prior methods.

sizes (8-11B parameters) and compute demands, limiting the accessibility of such advanced models.

At the same time, despite quantitative metrics, e.g., GenEval (Ghosh, Hajishirzi, and Schmidt 2023) and DPG-Bench (Hu et al. 2024b), suggesting smaller models with approx. 2B parameters like SANA-Sprint (Chen et al. 2025b) perform better than their large-scale counterparts, user-based evaluations on the Artificial Analysis Leaderboard¹ reveal a significant gap in the *perceived quality* between compact models and their larger counterparts, which quantitative metrics fail to capture. As such, there is a longstanding need for deploying large models even in resource-constrained settings in order to provide users with high-quality T2I capabilities.

Nonetheless, this endeavour constitutes a very challenging task due to the high memory and compute intensity (Li et al. 2025) of large DMs, thereby often requiring cloud solutions equipped with high-end GPUs and a minimum of 80GB VRAM. Concurrently, most of the renowned recent releases of foundational DMs, including SD3.5 (Esser et al. 2024), FLUX (Black Forest Labs 2024), and Seedream 3.0 (ByteDance 2025), are built upon multi-modal diffusion Transformer (MMDiT) backbones (Esser et al. 2024) instead of U-Net, revealing new compression opportunities (Section 2.1). At the moment, existing efforts to improve efficiency face important limitations. **Firstly**, *sampling step reduction* (Li et al. 2023; Stability AI 2023; Sauer et al. 2023)

¹<https://artificialanalysis.ai/image/leaderboard/text-to-image>



Figure 2: High-resolution image samples generated by compressed model using *HierarchicalPrune*, showcasing its superior visual quality across various visual styles, precisely following text prompts, and preserving the ability to draw typography.

and *efficient operator design* (Xie et al. 2025; Dao et al. 2022) are tailored to improving the speed of DMs rather than reducing memory requirements, leaving the important task of DM deployment in memory-constrained devices unresolved. **Secondly**, existing depth-pruning methods (Lee et al. 2024; Kim et al. 2024a; Fang et al. 2024) show promising results in both memory and computation reduction (Kim et al. 2024; Fang et al. 2024), outperforming width pruning (Fang, Ma, and Wang 2023; Castells et al. 2024a), but face critical scalability challenges. While they achieve reasonable compression on U-Net-based small DMs (2.6B or less), they fail to compress large-scale, state-of-the-art (SOTA) DMs such as SD3.5 Large (8B) and FLUX (11B) without experiencing significant degradation at 20-30% memory reduction as demonstrated in Table 1. The full-block-removal methods employed therein cannot capture the fine-grained impact of the subcomponents within each block, and more importantly, the different roles of blocks in different positions across the network’s hierarchy, leading to excessive performance drop at high pruning ratios. **Lastly**, orthogonal to pruning, *reduced-precision computation* (He et al. 2023; Li et al. 2025; Wang et al. 2024) has been proven effective, but it has not been combined with prior block-removal DM compression methods.

By observing the limitations of prior work, this paper identifies a novel insight for DMs that consist of MMDiT blocks: Such DMs form a two-fold hierarchy, spanning the inter- and intra-block levels. Inter-block hierarchy reflects the contribution of different blocks to disparate aspects of the output image (e.g. semantic structure, finer visual details), which is determined by their position in the overall architecture. Intra-block hierarchy highlights the subcomponents that compose

each MMDiT block and their diverse patterns of importance to the overall quality.

Building on this new viewpoint, we propose *HierarchicalPrune*, a principled compression methodology that overcomes the limitations of existing methods using three hierarchy-informed techniques (Fig. 3). First, we introduce **Hierarchical Position Pruning (HPP)**, a method that leverages our empirical insight that later MMDiT blocks contribute less to fundamental image structure, by strategically maintaining early blocks that form core image structures while pruning later blocks that primarily handle refinements. Second, we incorporate **Positional Weight Preservation (PWP)**, which freezes the non-pruned and earlier portions of the model during the distillation process. This approach maintains the integrity of early blocks, which are essential for image formation, while allowing later, less critical blocks to be updated. Finally, we propose **Sensitivity-Guided Distillation (SGDistill)**, which operates with a counterintuitive yet effective principle: blocks with higher importance are also more sensitive to change. Our analysis reveals that *in aggressive pruning settings, attempting to update these highly important blocks often proves detrimental to model performance*. As such, we enforce inverse distillation weights—assigning minimal or zero update weights to the most important blocks, while concentrating updates on less sensitive components.

We extensively benchmark *HierarchicalPrune* on both server and desktop-grade GPUs, demonstrating superior results over SOTA methods. Our contributions include:

- We identify a dual hierarchical structure in MMDiT DMs: an inter-block hierarchy (earlier blocks establish seman-

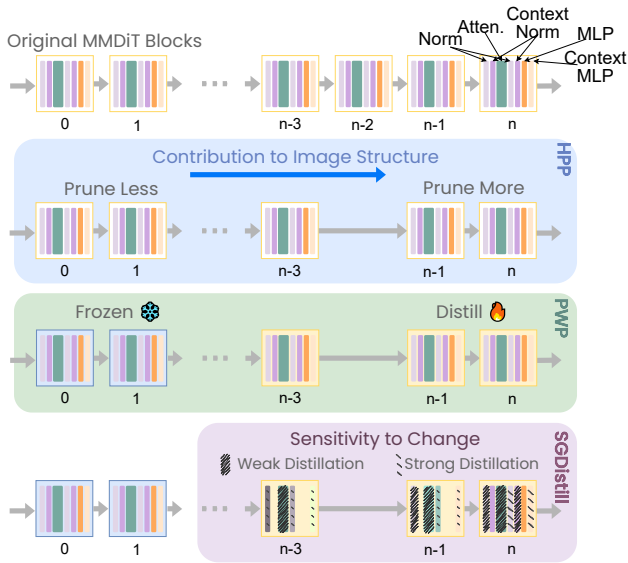


Figure 3: *HierarchicalPrune*'s compression framework leverages MMDiT's two-fold hierarchy (*inter-block*: early blocks establish semantics, later blocks refine; *intra-block*: varying subcomponent importance). It comprises (1) Hierarchical Position Pruning (HPP), maintaining early blocks while pruning later ones, (2) Positional Weight Preservation (PWP), freezing critical early blocks during distillation, and (3) Sensitivity-Guided Distillation (SGDistill), applying inverse weights—minimal updates to sensitive blocks and subcomponents. The resulting framework enables effective compression while preserving model capabilities.

tics, later blocks handle refinements) and an intra-block hierarchy (varying importance patterns of subcomponents within each MMDiT block).

- *HierarchicalPrune* establishes a comprehensive, position-aware pruning and distillation framework for large-scale DMs, for the first time, by combining HPP, PWP, and SGDistill with INT4 quantisation that achieves 77.5-80.4% memory reduction with minimal quality loss (3.2-4.8% ours vs. 15.3-41.2% degradation for prior works).
- An extensive user study with 85 participants demonstrates that *HierarchicalPrune* significantly outperforms all the baselines (Fig. 1): KOALA and BK-SDM suffer from a substantial 44.0-52.2% user-perceived quality degradation, and the SOTA small-scale DM, namely SANA-Sprint-1.6B, shows a 11.1-14.2% drop. In contrast, *HierarchicalPrune* observes a mere 4.8-5.3% degradation.

2 Methodology

2.1 Motivation

Examining MMDiT-based DMs, we identify a two-fold hierarchy: *i) inter-block* and *ii) intra-block hierarchy*. Inter-block hierarchy refers to the organisation of blocks along the overall architecture and implies a functional hierarchy where blocks have different responsibilities based on their position. Intra-block hierarchy focuses on the fact that individual blocks are internally composed of subcomponents, which in the case of

MMDiT consist of Norm, Context Norm, Attention, MLP, and Context MLP modules (Esser et al. 2024), with each subcomponent having a varying impact on performance based on its type and position.

Given this dual hierarchy, we conjectured that each block is responsible for different aspects of the generated images and that its position in the DM architecture largely determines these aspects. This also holds for subcomponents and is affected by their type. Concretely, we hypothesised that the contribution of different blocks to the output image is not uniform, regarding importance to performance and influence on specific image traits (*e.g.*, structure, texture or finer details).

To investigate this, we conducted a contribution analysis on SD3.5 Large Turbo over the HPSv2 dataset (Wu et al. 2023b), by removing both individual MMDiT blocks (Fig. 4a) and subcomponents (Fig. 4b, 4c and Fig. 8b–8e in Appendix B), and comparing the performance before and after. We observe that each of the subcomponents, as well as whole-block removal, demonstrates *different patterns of impact at different locations*. This was further highlighted when analysing the joint removal of multiple subcomponent types, where different subcomponent combinations exhibited significantly different effects on the final performance (see Appendix B).

We further observe that the performance drop is typically more severe in earlier stages of the network. Given the inter-block hierarchy hypothesis, we attribute this to earlier blocks contributing more to core elements of the output image that affect its quality, such as semantic structure, whereas later blocks are more important for finer visual details. To verify this, we examined the generated images when removing a number of MMDiT blocks at different locations throughout the network. Specifically, we randomly selected a few examples from the HPSv2 dataset and performed T2I generation with SD3.5 Large Turbo, removing three non-consecutive blocks each time. Fig. 5 shows that the overall image structure changes dramatically when layers are removed before and up to layer 10, while after layer 30, removal of the same number of blocks has minimal impact on the structure, while finer details, such as style, are still affected, providing evidence for the inter-block hierarchy of MMDiT DMs.

Despite this variability, existing full-block removal methods (Lee et al. 2024; Kim et al. 2024a) treat MMDiT blocks homogeneously, compromising performance when targeting high compression ratios. Our observations reveal a critical shortcoming: by ignoring the inter-block differences, these methods inadvertently prune blocks that are disproportionately important to visual quality while retaining less impactful ones. Moreover, by coarsely removing whole blocks, existing methods not only discard redundant layers but also eliminate subcomponents that might be essential for capturing fine-grain features. As shown in Section 3.2, this lack of differentiation leads to a steep drop in model performance under aggressive pruning rates. These insights motivate us to design hierarchy-informed techniques for the effective compression of large MMDiT-based DMs.

2.2 HierarchicalPrune

Motivated by our findings in Section 2.1, we propose *HierarchicalPrune* (Fig. 3), a cohesive, multi-stage pruning and

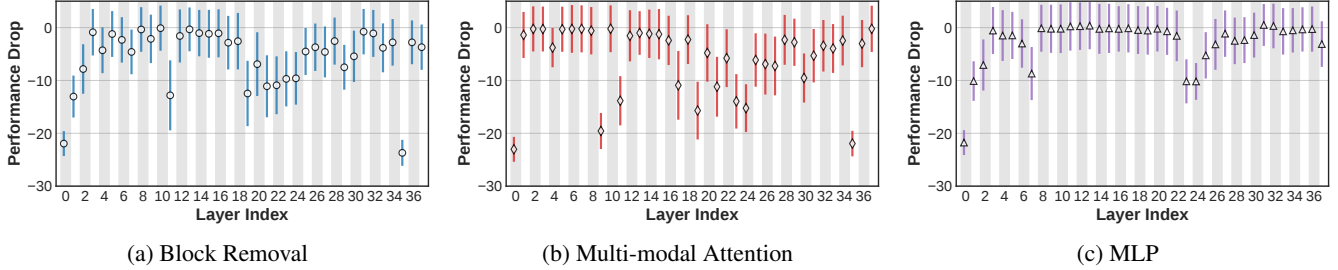


Figure 4: Fine-grained contribution analysis of SD3.5 Large Turbo on the HPSv2 dataset by removing either an entire MMDiT block (a), following prior depth pruning approaches (Lee et al. 2024; Kim et al. 2024a; Fang et al. 2024), or an intra-block subcomponent (b, c and see Fig. 8 in Appendix B for full set of analysis). We report the performance drop compared to the original model. The discrepancy in performance drop patterns reveals the different patterns of importance of each subcomponent.

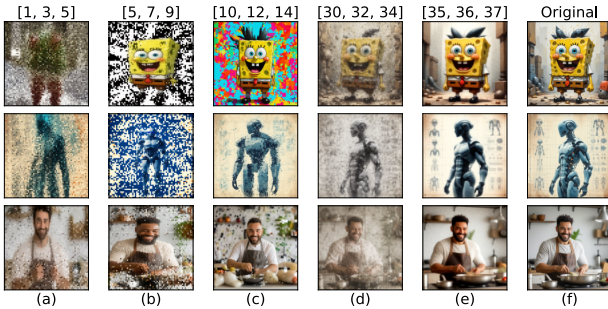


Figure 5: Impact of removing MMDiT blocks at different positions. Compared to original outputs (f), removing earlier layers leads to high impact on image structure (a-c), whereas removing later blocks affects mainly fine details (d, e).

distillation approach that respects the hierarchical nature of DMs. *HierarchicalPrune* introduces three key techniques: *i*) Hierarchical Position Pruning (HPP), *ii*) Positional Weight Preservation (PWP), and *iii*) Sensitivity-Guided Distillation (SGDistill). These techniques operate complementarily with the objective to maximise the model compression rate while maintaining output quality throughout the T2I process. Algorithm 1 presents the complete methodology. Given a pre-trained model m , the first stage (line 1) is responsible for producing a pruned model m_{pruned} with a compression ratio r using HPP.

In the second stage (lines 2-5), *HierarchicalPrune* first prepares the pruned model for distillation, by preserving its most sensitive blocks by means of PWP (line 2), yielding the selectively frozen model $m_{\text{pruned,frz}}$. Then, distillation is applied in order to improve the attainable image quality. Specifically, if the target compression ratio r indicates aggressive compression by surpassing a threshold r_{thres} , *HierarchicalPrune* applies SGDistill (line 3), our proposed sensitivity-guided distillation method. For moderate compression, a simpler distillation process is followed (line 5). As a final step, the resulting model is optionally quantised with 4-bit weights and 16-bit activations (line 6) to obtain additional compression.

Hierarchical Position Pruning: At its foundation, *Hierarchical Position Pruning (HPP)* leverages our insight that deeper MMDiT blocks contribute less to core visual structure.

Algorithm 1: HierarchicalPrune

Input: DM m with $|\mathcal{B}|$ MMDiT blocks and parameters θ
Subcomponent set \mathcal{C} of each MMDiT block
Quality threshold q , Compression ratio r
Dataset \mathcal{D} and calibration set $\mathcal{D}_{\text{calib}} \subset \mathcal{D}$

Output: Pruned and distilled DM m' with parameters θ'

** --- Stage 1 - HPP / PWP --- **

- 1 $\Delta P(i, c) \leftarrow \text{ContributionAnalysis}(m, \mathcal{D}_{\text{calib}}, i, c)$,
 $\forall i \in [0, |\mathcal{B}| - 1], \forall c \in \mathcal{C}$
▷ Estimate importance based on performance drop
- $\text{Score}(i, c) \leftarrow \text{PrunabilityScore}(\Delta P, i, c)$
 $\forall i \in [0, |\mathcal{B}| - 1], \forall c \in \mathcal{C}$
▷ Calculate prunability as per Eq. (1) and (2)
- $m_{\text{pruned}}, \theta_{\text{pruned}} \leftarrow \text{ApplyHPP}(m, \theta, \text{Score}, r)$
▷ Given the prunability per block, remove blocks until compression ratio

** --- Stage 2 - Distillation --- **

- 2 $m_{\text{pruned,frz}}, \theta_{\text{pruned,frz}} \leftarrow \text{ApplyPWP}(m_{\text{pruned}}, \theta_{\text{pruned}})$
▷ Freeze the non-pruned and earlier parts of the model
- if** $r \geq r_{\text{thres}}$ **then**
3 $m', \theta' \leftarrow \text{SGDistill}(m'_{\text{pruned,frz}}, \theta'_{\text{pruned,frz}}, \mathcal{D}, q)$
▷ If aggressive pruning, use sensitivity-guided distillation
- else**
- 5 $m', \theta' \leftarrow \text{Distill}(m'_{\text{pruned,frz}}, \theta'_{\text{pruned,frz}}, \mathcal{D}, q)$
▷ If moderate pruning, use normal distillation

** --- Stage 3 - Post-Training Quantisation --- **

- 6 $m', \theta' \leftarrow \text{PTQ}(m', \theta', \text{precision}=\text{W4A16})$
▷ Apply PTQ for further compression

7 **return** m', θ'

As such, HPP strategically targets blocks for removal based on their hierarchical position within the model, maintaining early blocks that form image structures while pruning deeper blocks that mainly handle visual details.

Formally, for a DM with $|\mathcal{B}|$ MMDiT blocks, we first introduce a position weight function $W_{\text{pos}}(i)$ (Eq. (2)) that captures the inter-block hierarchy by mapping the more critical earlier blocks to lower values and guiding the pruning process towards the later layers. Next, we calculate a prunability score Score (Eq. (1)) for each block in \mathcal{B} and subcomponent(s), consisting of the performance drop from the original

model scaled by the position weight function. Concretely:

$$\text{Score}(i, c) = -|\Delta P(i, c)| \times W_{\text{pos}}(i) \quad \forall i \in [0, |\mathcal{B}|-1] \quad (1)$$

$$W_{\text{pos}}(i) = e^{(i-|\mathcal{B}|)/|\mathcal{B}|} \quad (2)$$

where i is the block index and \mathcal{B} is the number of blocks, $c \in \mathcal{C}$ is the subcomponent type, $\Delta P(i, c)$ is the importance score quantified as performance drop from the original model when block i and/or subcomponents c are removed, $W_{\text{pos}}(i)$ is the position weight function that favours later layers.

Positional Weight Preservation: Building upon this position-aware foundation, we incorporate *Positional Weight Preservation (PWP)*, which freezes the non-pruned and earlier portions of the model during the distillation process. This straightforward yet effective approach maintains the integrity of early blocks which are essential for image formation, while allowing later, less critical blocks to be updated. By systematically keeping weights static based on their position in the network, PWP significantly outperforms basic position-based pruning, HPP, for moderate pruning scenarios (e.g., 25% parameter reduction), as it ensures that the most structurally important parts of the model remain intact. Note that even at moderate compression levels, prior methods suffer substantial quality degradation (15.3-41.2% in Table 1), while our approach maintains near-original performance.

Sensitivity-Guided Distillation: For aggressive parameter reduction (e.g., $\geq 30\%$), however, we discovered that even with careful pruning and preservation, excessive block-level pruning leads to unacceptable quality degradation. To address this challenge, we introduce *Sensitivity-Guided Distillation (SGDistill)*, which operates with a counterintuitive yet effective principle: blocks with higher importance are also more sensitive to change.

Our analysis reveals that *in aggressive pruning settings, attempting to update these highly important blocks often proves detrimental to model performance* (showing 31.9% average quality reduction even with PWP, see Table 3). Consequently, SGDistill applies inverse distillation weights—assigning minimal or zero update weights to the most important blocks, while concentrating updates on less sensitive components. This approach, combined with subcomponent (e.g., Normalisation and MLP within an MMDiT block) pruning for the remaining portion of the DM after pruning up to r_{thres} (line 2) with HPP and PWP, extends our hierarchical approach from the block-level to the intra-block subcomponents, preserving the carefully tuned parameters of critical blocks while allowing adaptation in less sensitive regions. Our experimental results confirm this hypothesis, showing that protecting sensitive blocks from significant updates during distillation is key to maintaining quality at high compression ratios, reducing average quality degradation from 31.9% to 10.1% (Table 3).

Concretely, the objective of the distillation process of *HierarchicalPrune* leverages a combination of the feature loss, $\mathcal{L}_{\text{feat}}$, i.e., a feature distillation loss, and the standard knowledge distillation (KD) loss to minimise the difference between the final output of the compressed (i.e., student) model θ' and the original (i.e., teacher) model θ , leading to the overall loss $\mathcal{L} = \mathcal{L}_{\text{feat}} + \mathcal{L}_{\text{KD}}$, with \mathcal{L}_{KD} expressed as:

$$\mathcal{L}_{\text{KD}} = \mathbb{E} [\|v_{\theta'}(x_t, t) - v_{\theta}(x_t, t)\|^2] \quad (3)$$

where t is the time step, and x_t is the noisy diffusion sample, started from the clean latent x_0 from the autoencoder (VAE). For the selected set of blocks to update $\mathcal{B}^* \in \mathcal{B}$ each associated with an importance score $\Delta P(i, c)$ (Eq. (1)), by denoting feature output of block i in teacher and student models as f_{θ}^i and $f_{\theta'}^i$, respectively, we have:

$$\mathcal{L}_{\text{feat}} = \mathbb{E} \left[\sum_{i \in [0, |\mathcal{B}^*|-1]} \|f_{\theta'}^i(x_t, t) - f_{\theta}^i(x_t, t)\|^2 \right] \quad (4)$$

SGDistill specifies that, for each block, we scale the final parameter update by its sensitivity, i.e., $\frac{1}{\Delta P(i, c)}$, regulating in this way the rate of change of each block during distillation.

3 Evaluation

To evaluate *HierarchicalPrune* against existing methods, we conducted both quantitative and qualitative comparisons (Section 3.2). We also perform an ablation study to assess each proposed component, the impact of quantisation, and the robustness of text-drawing capability (Section 3.3).

3.1 Experimental Setup

Architectures, Datasets, and Implementation: We target SD3.5 Large Turbo (8B) and FLUX.1-Schnell (12B), two SOTA models designed to perform diffusion tasks within a small number of steps (e.g., 4). Following prior work (Lee et al. 2024), we use the YE-POP dataset (HuggingFace 2024) consisting of 500K images. We implemented the *HierarchicalPrune* pipeline in PyTorch and Diffusers (von Platen et al. 2022), utilising SD3.5 Large Turbo and FLUX.1-Schnell model checkpoints. For 4-bit weight quantisation, we adopt `bitsandbytes` (Dettmers et al. 2022).

Baselines: We compare *HierarchicalPrune* with two prior works related to depth pruning and distillation: *i*) KOALA (Lee et al. 2024), and *ii*) BK-SDM (Kim et al. 2024a). BK-SDM proposed block pruning of U-Net-based models using the CLIP score (Hessel et al. 2021), followed by distilling the pruned model using knowledge from the original model. KOALA follows a similar method, replacing CLIP-score-based importance ranking with scores from each block’s input-output cosine similarity. Also, we compare against SANA-Sprint-1.6B (Chen et al. 2025b), a SOTA compact model optimised for efficient on-device deployment.

Metrics: To quantitatively evaluate all methods, we employ the two most recent and representative image quality metrics, GenEval (Ghosh, Hajishirzi, and Schmidt 2023) (used in (Xie et al. 2025; Esser et al. 2024)) and HPSv2 (Wu et al. 2023a) (used in KOALA (Lee et al. 2024)). Furthermore, we conduct a user study to assess human preferences for the generated images that are difficult to capture with the quantitative quality metrics. Similar to (Sheynin et al. 2023) and (Dai et al. 2023), we evaluate two aspects: (i) Text Alignment: How well a generated image follows the description of the text prompt, (ii) Image Quality: The overall visual quality considering clarity, colour, composition, and other factors. On the system resource side, we report the measured peak memory usage and latency of running the target models on diverse GPUs, including A6000 (Table 2), GTX 3090 and A100 GPUs (see Appendix Table 5).

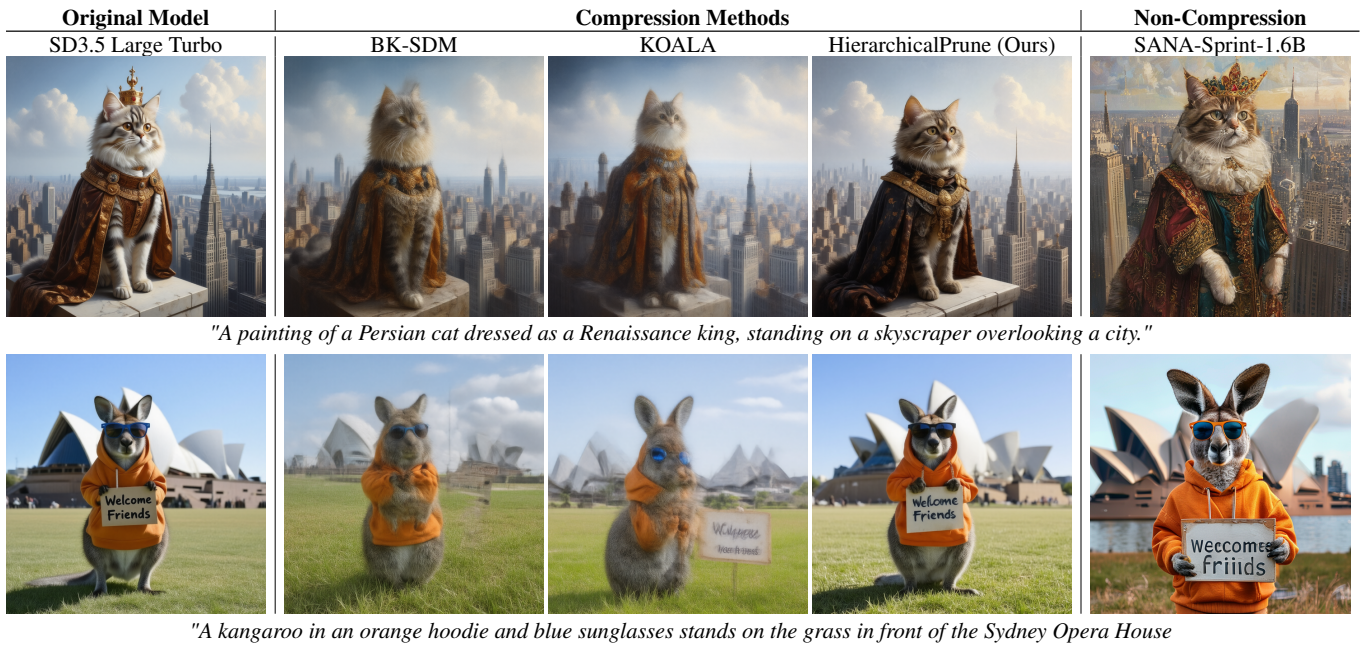


Figure 6: Visual comparison demonstrating the quality difference between the original model (column 1), depth pruning based on BK-SDM (column 2), KOALA (column 3), and our proposed *HierarchicalPrune* (column 4), as well as the SOTA small-scale diffusion model, SANA-Sprint-1.6B (column 5). Our approach successfully maintains visual quality while delivering 79.5% memory reduction over the original model. Notably, our method preserves the text drawing capability of the original SD3.5 Large Turbo model, where SANA-Sprint-1.6B is limited.

3.2 Main Results

Image Quality under Memory Compression: Fig. 6 presents a qualitative comparison of example outputs. *HierarchicalPrune* shows better visual outputs, outperforming all the baselines, and maintaining high fidelity to the original DMs, both in structure and fine details. More examples can be found in Fig. 2 and Appendix C. Table 1 shows that *HierarchicalPrune* achieves comparable quality to the original model based on both HPSv2 and GenEval. While baselines induce substantial quality degradation of 38.2-45.1%, our approach yields only a minimal drop (\sim 3.2-4.8%), given the same range of memory compression ratio, showcasing the effectiveness of our hierarchical compression strategy.

Importantly, *HierarchicalPrune* achieves 79.5% lower peak memory, a reduction from 15.8 GB to 3.24 GB for SD3.5 Large Turbo model. Notably, the peak memory of our compressed SD3.5 model (3.24 GB) is comparable to SANA-Sprint (3.14 GB), but with substantially better image quality. For FLUX.1-Schnell, our method achieves 80.4% peak memory reduction, indicating its generalisability to larger DMs.

User Study: Our user study with 85 participants reaffirms the effectiveness of *HierarchicalPrune*. Fig. 1 presents the mean opinion score (MOS) across all methods. *HierarchicalPrune* achieves remarkably close MoS to the original SD3.5 Large Turbo model with only minimal reduction (4.8% for text alignment, 5.3% for image quality). In contrast, SANA-Sprint-1.6B shows a noticeable quality drop (14.2% for text alignment, 11.1% for image quality) compared to the original model, while other baselines (*i.e.*, BK-SDM, KOALA) show a substantial 44.0-52.2% degradation.

Model	Method	Memory	GenEval \uparrow	HPSv2 \uparrow	Reduction \downarrow
Linear DiT	SANA-Sprint	3.14 (100%)	0.77	29.61	-
	Original	15.8 (100%)	0.71	30.29	-
SD3.5 Large Turbo	KOALA	12.6 (79.4%)	0.37	19.99	41.2%
	KOALA (+Quant)	3.56 (22.5%)	0.33	18.44	46.4%
	BK-SDM	12.6 (79.4%)	0.38	21.21	38.2%
	BK-SDM (+Quant)	3.56 (22.5%)	0.34	19.83	43.3%
	Ours (HPP+PWP+Q)	3.56 (22.5%)	0.69	28.15	4.8%
	Ours (All)	3.24 (20.5%)	0.62	26.29	13.3%
FLUX.1 Schnell	Original	22.6 (100%)	0.66	29.71	-
	KOALA	15.9 (70.5%)	0.38	25.24	28.7%
	BK-SDM	15.9 (70.5%)	0.45	27.38	19.8%
	Ours (All)	4.44 (19.6%)	0.64	28.69	3.2%

Table 1: Image quality measured in GenEval and HPSv2 scores and the corresponding peak memory measurements in GB and remaining ratios (%) after compression. KOALA and BK-SDM experience substantial degradation of image quality when reducing memory usage by 20-30%. *HierarchicalPrune* achieves significantly reduced memory usage while maintaining the image quality close to the original models.

Training Cost Comparison: Beyond quality preservation, *HierarchicalPrune* requires only 615–1,287 A100 GPU hours for architectural profiling and distillation (Appendix C.1), compared to 140k–200k A100 GPU hours required to pre-train small-scale DMs like SD1.4 and SD2.1 (Pernias et al. 2024). This demonstrates that compressing existing large models is far more cost-efficient than training compact alternatives from scratch, while maintaining superior quality.

Latency: We measure per-step latency of *HierarchicalPrune* and baselines both server- and desktop-grade GPUs with different specifications (A100, A6000, and GTX 3090

Model	Method	Latency ↓	Reduction ↑
Linear DiT	SANA-Sprint	54 ms	-
	Original	823 ms	-
SD3.5	KOALA	642 ms	22.0%
	BK-SDM	642 ms	22.0%
Large Turbo	Ours (HPP, PWP, Quant)	593 ms	27.9%
	Ours (All)		
FLUX.1 Schnell	Original	756 ms	-
	KOALA	432 ms	42.9%
	BK-SDM	432 ms	42.9%
	Ours (All)	469 ms	38.0%

Table 2: Comparison of per-step latency of the DM Transformer measured on an A6000 GPU with 48 GB of VRAM.

with 80, 48, and 25 GB VRAM, respectively, reported in Appendix Table 5). While we report the measurement results from A6000 in Table 2, the reduction rates are similar across GPUs. Our compression pipeline achieves 27.9% and 38.0% latency reduction compared to the original SD3.5 Large Turbo and FLUX.1-Schnell, respectively.

Overall, our findings reveal that respecting the hierarchical sensitivity of diffusion model components enables more effective model compression than approaches that treat all blocks as equally important, establishing a new paradigm for the efficient deployment of large-scale generative models.

3.3 Ablation Study and Analysis

Impact of Each Component of *HierarchicalPrune*: We conducted an ablation study of *HierarchicalPrune* to investigate the contribution of each component: (1) HPP, (2) PWP, (3) HPP+PWP+SGDistill, and (4) our final form, HPP+PWP+SGDistill+Quant. Table 3 shows that using HPP+PWP drastically improves image quality compared to HPP only and prior works (*i.e.*, BK-SDM, KOALA). Moreover, by leveraging intra-block sensitivity, SGDistill substantially reduces image quality degradation from 31.0% to 10.1% compared to HPP+PWP at an aggressive parameter reduction rate of 30%². Our method outperforms all baselines with superior image quality, lower memory, and faster execution.

Impact of Quantisation: We investigate how much W4A16 quantisation in our pipeline affects the final image quality. Table 3 shows the quality metrics with and without quantisation applied. The impact of quantisation is as small as 2.4-3.5% in both GenEval and HPSv2 scores.

Robustness of Text Generation: As shown in Fig. 6, prior works (BK-SDM and KOALA in pruning) and small-scale DMs (SANA-Sprint-1.6B) are limited in synthesising legible texts in the generated images. However, *HierarchicalPrune* demonstrates the superior quality in text generation in Fig. 6 and 2. These results represent the effectiveness of our proposed method in preserving the innate capability of the original model, not only in the aesthetic quality but also in other aspects like text generation.

4 Related Work

T2I Diffusion Models. Since Stable Diffusion (SD) (Romach et al. 2022), there has been rapid community adoption

²We set $r_{thres} = 0.25$ for both models, determined where quality degradation exceeded 15% without SGDistill.

Pruning Ratio	Method	Remaining Memory	GenEval ↑	HPSv2 ↑	Reduction ↓
None (0%)	Original	100%	0.71	30.29	-
Moderate Pruning (20%)	Ours (HPP)	79.4%	0.03	11.08	79.4%
	Ours (+PWP)	79.4%	0.71	28.97	2.5%
	Ours (+Quant)	22.5%	0.69	28.15	4.8%
Aggressive Pruning (30%)	Ours (HPP)	71.5%	0.0	7.00	88.4%
	Ours (+PWP)	71.5%	0.46	21.74	31.9%
	Ours (+SGDistill)	71.5%	0.64	27.29	10.1%
	Ours (+Quant)	20.5%	0.62	26.29	13.3%

Table 3: Ablation study of each component and quantisation in *HierarchicalPrune* on SD3.5 Large Turbo.

and iterative updates in the field. SDv1.4 and v1.5 were released in 2022, enhancing efficiency and enabling specialised fine-tuning (*e.g.*, DreamBooth). By mid-2023, Stable Diffusion XL (SDXL) significantly advanced resolution (1024×1024), text comprehension, and image quality. By the end of 2024, the adoption of MMDiT (Esser et al. 2024) as a backbone in SD3, SD3.5, and FLUX brought a significant boost in image generation quality and alignment with long text input. At the same time, the increasing parameter count, particularly in the backbone Transformer blocks, significantly improved image quality but with excessive resource demands. As SOTA models like SD3.5 scale up to as many as 8B parameters, efficient inference on resource-constrained devices becomes impractical. This underlines the significance of approaches such as our *HierarchicalPrune* framework, to enable DM deployment outside of high-end compute setups.

Compression Techniques. Traditional model compression methods, including distillation (Hinton, Vinyals, and Dean 2015), pruning (Han, Mao, and Dally 2016) (such as depth pruning via block removal (Ghiasi, Lin, and Le 2018; Kim et al. 2024c), width pruning (Kwon et al. 2024)), and quantisation (Jacob et al. 2018) have been applied to DMs (Lee et al. 2024; Kim et al. 2024a; Castells et al. 2024b; Hu et al. 2024a; Fang et al. 2024; Li et al. 2025; Kim et al. 2024b; Zhang et al. 2024), resulting in variable parameter efficiency gains. Specifically, (Kim et al. 2024a) and (Lee et al. 2024) explore block removal in the U-Net backbone of SD1.5 and SDXL models, respectively. More recently, (Li et al. 2025) explored the effectiveness of quantisation on DMs for memory reduction, hence our *HierarchicalPrune* employs adjunct post-training quantisation as part of its design. Orthogonal to compression, Δ -DiT (Chen et al. 2024) identifies hierarchical patterns in DiT (MMDiT predecessor) for inference caching. Our analysis confirms that this generalises to modern MMDiT and further leverages these insights for the compression of SOTA multi-billion-scale DMs, under a unified framework.

5 Conclusion

In this work, we proposed *HierarchicalPrune*, pushing the limit of compressing MMDiT-based large-scale DMs through hierarchical insights. By combining HPP, PWP, and SGDistill with quantisation, *HierarchicalPrune* achieves 77.5-80.4% memory reduction and 27.9-38.0% latency improvements with minimal quality loss, bringing SOTA large-scale DMs (SD3.5 Large Turbo, FLUX.1-Schnell) within reach of resource-constrained environments and democratising access to high-quality T2I generation.

References

- Black Forest Labs. 2024. Flux.1 Model Family. <https://blackforestlabs.ai/announcing-black-forest-labs/>.
- ByteDance. 2025. Seedream 3.0 Technical Report. In <https://arxiv.org/pdf/2504.11346>.
- Castells, T.; Song, H.-K.; Kim, B.-K.; and Choi, S. 2024a. LD-Pruner: Efficient Pruning of Latent Diffusion Models using Task-Agnostic Insights. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.
- Castells, T.; Song, H.-K.; Piao, T.; Choi, S.; Kim, B.-K.; Yim, H.; Lee, C.; Kim, J. G.; and Kim, T.-H. 2024b. EdgeFusion: On-Device Text-to-Image Generation. *CoRR*, abs/2404.11925.
- Chen, J.; Cai, H.; Chen, J.; Xie, E.; Yang, S.; Tang, H.; Li, M.; and Han, S. 2025a. Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models. In *International Conference on Learning Representations (ICLR)*.
- Chen, J.; Xue, S.; Zhao, Y.; Yu, J.; Paul, S.; Chen, J.; Cai, H.; Xie, E.; and Han, S. 2025b. SANA-Sprint: One-Step Diffusion with Continuous-Time Consistency Distillation. *arXiv:2503.09641*.
- Chen, P.; Shen, M.; Ye, P.; Cao, J.; Tu, C.; Bouganis, C.-S.; Zhao, Y.; and Chen, T. 2024. *Delta-DiT: A Training-Free Acceleration Method Tailored for Diffusion Transformers*. *arXiv preprint arXiv:2406.01125*.
- 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. 2023. Emu: Enhancing Image Generation Models Using Photogenic Needles in a Haystack. *arXiv:2309.15807*.
- Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; and Ré, C. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Dettmers, T.; Lewis, M.; Belkada, Y.; and Zettlemoyer, L. 2022. GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *International Conference on Machine Learning (ICML)*.
- Fang, G.; Li, K.; Ma, X.; and Wang, X. 2024. TinyFusion: Diffusion Transformers Learned Shallow. *arXiv preprint arXiv:2412.01199*.
- Fang, G.; Ma, X.; and Wang, X. 2023. Structural Pruning for Diffusion Models. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*.
- Ghiasi, G.; Lin, T.-Y.; and Le, Q. V. 2018. DropBlock: A Regularization Method for Convolutional Networks. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Ghosh, D.; Hajishirzi, H.; and Schmidt, L. 2023. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Han, S.; Mao, H.; and Dally, W. J. 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *International Conference on Learning Representations (ICLR)*.
- He, Y.; Liu, L.; Liu, J.; Wu, W.; Zhou, H.; and Zhuang, B. 2023. PTQD: Accurate Post-Training Quantization for Diffusion Models. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*.
- Hessel, J.; Holtzman, A.; Forbes, M.; Le Bras, R.; and Choi, Y. 2021. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7514–7528. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.
- Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.
- Hu, D.; Chen, J.; Huang, X.; Coskun, H.; Sahni, A.; Gupta, A.; Goyal, A.; Lahiri, D.; Singh, R.; Idelbayev, Y.; et al. 2024a. SnapGen: Taming High-Resolution Text-to-Image Models for Mobile Devices with Efficient Architectures and Training. *arXiv preprint arXiv:2412.09619*.
- Hu, X.; Wang, R.; Fang, Y.; Fu, B.; Cheng, P.; and Yu, G. 2024b. ELLA: Equip Diffusion Models with LLM for Enhanced Semantic Alignment. *arXiv:2403.05135*.
- HuggingFace. 2024. YE-POP Dataset. <https://huggingface.co/datasets/Ejafa/ye-pop>. Accessed: 2025-March.
- Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; and Kalenichenko, D. 2018. Quantization and Training of Neural Networks for Efficient Integer-arithmetic-only Inference. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the Design Space of Diffusion-Based Generative Models. In *Advances in Neural Processing Systems (NeurIPS)*.
- Katharopoulos, A.; Vyas, A.; Pappas, N.; and Fleuret, F. 2020. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In *International Conference on Machine Learning (ICML)*.
- Kim, B.-K.; Song, H.-K.; Castells, T.; and Choi, S. 2024a. BK-SDM: A Lightweight, Fast, and Cheap Version of Stable Diffusion. In *European Conference on Computer Vision (ECCV)*.
- Kim, G.; Kim, B.; Park, E.; and Cho, S. 2024b. Diffusion Model Compression for Image-to-Image Translation. In *Proceedings of the Asian Conference on Computer Vision*, 2105–2123.

- Kim, J.; Halabi, M. E.; Ji, M.; and Song, H. O. 2024c. Layer-Merge: Neural Network Depth Compression through Layer Pruning and Merging. In *International Conference on Machine Learning (ICML)*.
- Kwon, Y. D.; Li, R.; Venieris, S. I.; Chauhan, J.; Lane, N. D.; and Mascolo, C. 2024. TinyTrain: Resource-Aware Task-Adaptive Sparse Training of DNNs at the Data-Scarce Edge. In *International Conference on Machine Learning (ICML)*.
- Lee, Y.; Park, K.; Cho, Y.; Lee, Y.-J.; and Hwang, S. J. 2024. Koala: Empirical Lessons toward Memory-Efficient and Fast Diffusion Models for Text-to-Image Synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Li, M.; Lin, Y.; Zhang, Z.; Cai, T.; Guo, J.; Li, X.; Xie, E.; Meng, C.; Zhu, J.-Y.; and Han, S. 2025. SVDQuant: Absorbing Outliers by Low-Rank Component for 4-Bit Diffusion Models. In *International Conference on Learning Representations (ICLR)*.
- Li, Y.; Wang, H.; Jin, Q.; Hu, J.; Chemerys, P.; Fu, Y.; Wang, Y.; Tulyakov, S.; and Ren, J. 2023. SnapFusion: Text-to-Image Diffusion Model on Mobile Devices within Two Seconds. In <https://arxiv.org/abs/2306.00980>.
- Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2023. Flow Matching for Generative Modeling. In *International Conference on Learning Representations (ICLR)*.
- Lu, Z.; Wang, Z.; Huang, D.; Wu, C.; Liu, X.; Ouyang, W.; and Bai, L. 2024. FiT: Flexible Vision Transformer for Diffusion Model. In *International Conference on Machine Learning (ICML)*.
- Peebles, W.; and Xie, S. 2022. Scalable Diffusion Models with Transformers. In *International Conference on Computer Vision (ICCV)*.
- Pernias, P.; Rampas, D.; Richter, M. L.; Pal, C. J.; and Aubreville, M. 2024. Wüerstchen: An Efficient Architecture for Large-Scale Text-to-Image Diffusion Models. In <https://openreview.net/forum?id=gU58d5QeGv>.
- Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. In <https://arxiv.org/abs/2307.01952>.
- Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Sauer, A.; Lorenz, D.; Blattmann, A.; and Rombach, R. 2023. Adversarial Diffusion Distillation. In <https://arxiv.org/abs/2311.17042>.
- Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; Schramowski, P.; Kundurthy, S.; Crowson, K.; Schmidt, L.; Kaczmarczyk, R.; and Jitsev, J. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Sheynin, S.; Polyak, A.; Singer, U.; Kirstain, Y.; Zohar, A.; Ashual, O.; Parikh, D.; and Taigman, Y. 2023. Emu Edit: Precise Image Editing via Recognition and Generation Tasks. [arXiv:2311.10089](https://arxiv.org/abs/2311.10089).
- Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations (ICLR)*.
- Stability AI. 2023. SDXL-Turbo. <https://huggingface.co/stabilityai/sdxl-turbo>.
- von Platen, P.; Patil, S.; Lozhkov, A.; Cuenca, P.; Lambert, N.; Rasul, K.; Davaadorj, M.; Nair, D.; Paul, S.; Berman, W.; Xu, Y.; Liu, S.; and Wolf, T. 2022. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>.
- Wang, H.; Shang, Y.; Yuan, Z.; Wu, J.; and Yan, Y. 2024. QuEST: Low-bit Diffusion Model Quantization via Efficient Selective Finetuning. In <https://arxiv.org/abs/2402.03666>.
- Wu, X.; Hao, Y.; Sun, K.; Chen, Y.; Zhu, F.; Zhao, R.; and Li, H. 2023a. Human Preference Score v2: A Solid Benchmark for Evaluating Human Preferences of Text-to-Image Synthesis. *arXiv preprint arXiv:2306.09341*.
- Wu, X.; Sun, K.; Zhu, F.; Zhao, R.; and Li, H. 2023b. Human Preference Score: Better Aligning Text-to-Image Models with Human Preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Xie, E.; Chen, J.; Chen, J.; Cai, H.; Tang, H.; Lin, Y.; Zhang, Z.; Li, M.; Zhu, L.; Lu, Y.; and Han, S. 2025. SANA: Efficient High-Resolution Text-to-Image Synthesis with Linear Diffusion Transformers. In *International Conference on Learning Representations (ICLR)*.
- Zhang, D.; Li, S.; Chen, C.; Xie, Q.; and Lu, H. 2024. Laptopdiff: Layer pruning and normalized distillation for compressing diffusion models. *arXiv preprint arXiv:2404.11098*.