Importance-Aware Low-Rank Distillation of Diffusion Transformers

Denis Zavadski*[1,2], Sebastian Heid*[1,2], Damjan Kalšan[1], Stefan Roth[2,3,4], and Carsten Rother[1,2]

[1] Heidelberg University [2] Zuse School ELIZA [3] TU Darmstadt [4] hessian.AI

Abstract. Diffusion Transformers (DiTs) have emerged as a dominant architecture for high-quality text-to-image generation, yet their scale poses challenges for efficient deployment. While truncated singular value decomposition (SVD) is a principled tool for parameter reduction, evidence from large language models (LLMs) suggests that naive low-rank approximation can cause catastrophic failure. In contrast, we find that truncated SVD in DiTs produces smooth degradation even under substantial global compression, with redundancy distributed across projection matrices throughout the whole network rather than concentrated in a few transformer blocks. Building on these insights, we introduce *SVDtrunc*, a two-step block-level compression scheme, first allocating ranks across blocks and compressing the least important ones via truncated SVD under a global parameter budget, and then fine-tuning all blocks with modular knowledge distillation and a rectified-flow objective. We apply SVDtrunc to FLUX.dev across compression levels ranging from 40–90 % of the original parameter count. Across three benchmarks, GenEval, HPSv2, and DPG, we outperform all competing approaches. Notably, and in contrast to prior work, we retain near-full performance at 68 % and remain competitive even at 57 % of the original parameter budget. Furthermore, we show that SVDtrunc complements step distillation and achieves strong results even without fine-tuning, positioning it as a practical continuation of efficiency improvements beyond diffusion step reduction for large-scale generative models.

1 Introduction

Diffusion Transformers [26] (DiTs) have emerged as a dominant backbone for high-quality text-to-image generation. Their strong compositional reasoning and perceptual fidelity are enabled by large-scale Transformer architectures with substantial parameter and memory requirements. As state-of-the-art models continue to be scaled [22, 39, 41], efficient deployment becomes increasingly challenging, motivating research on reducing neural function evaluations and model parameters. On the computational side, step distillation [4, 21, 32, 33] reduces the number of diffusion or flow evaluations, approaching the theoretical limit of single-step generation. Once this limit is reached, further efficiency gains within

* Equal Contribution

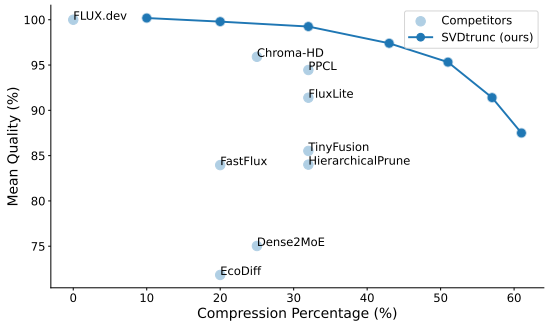

Fig. 1: Quantitative results of our SVDtrunc approach compared to the state of the art. We plot quality relative to the original FLUX.dev model [19] (in %, ↑) against parameter reduction (%, ↑). The quality is averaged across the relative performances on the GenEval [10], DPG [13], and HPSv2 [40] benchmarks. We outperform prior models across all compression-levels.

the same pre-trained architecture have to come from reducing the parameter footprint. Existing parameter reduction methods [8, 24] typically rely on pruning or structural modification of Transformer blocks to achieve compactness.

A principled alternative to pruning and structural modification is low-rank approximation via singular value decomposition (SVD) [1, 6, 12], which provides the optimal rank-constrained approximation of a matrix under standard norms (*i.e.*, all unitarily invariant norms [25]). While attractive for parameter compression, studies in autoregressive LLMs show that naive low-rank approximation can substantially degrade quality unless combined with careful fine-tuning or architectural adaptation [12, 28, 37, 38, 42]. Whether Diffusion Transformers exhibit similar sensitivity remains unclear, as diffusion models iteratively refine latent representations over many steps where structural perturbations may accumulate.

In this work, we show that Diffusion Transformers can be aggressively compressed via truncated SVD while retaining strong generative performance (Fig. 2). In contrast to LLMs, where naive low-rank approximation can cause substantial degradation and catastrophic failure [12, 38], we observe that DiTs remain surprisingly robust to rank reduction (Fig. 1). Even under large parameter reductions, compressed models preserve perceptual fidelity with only minor quality loss. This robustness emerges when compressing projection matrices rather than removing entire blocks, suggesting that redundancy in Diffusion Transformers can be exploited through distributed low-rank compression without architectural modification. To examine whether such redundancy is already present in pretrained DiTs rather than purely a result of distillation, we conduct a side experiment where models are compressed without fine-tuning. Even for substantial parameter reduction, *e.g.* by up to 30 %, low-rank approximation of projection matrices causes only minor quality degradation, whereas coarse architectural modifications lead to substantially stronger degradation [2, 8, 45].

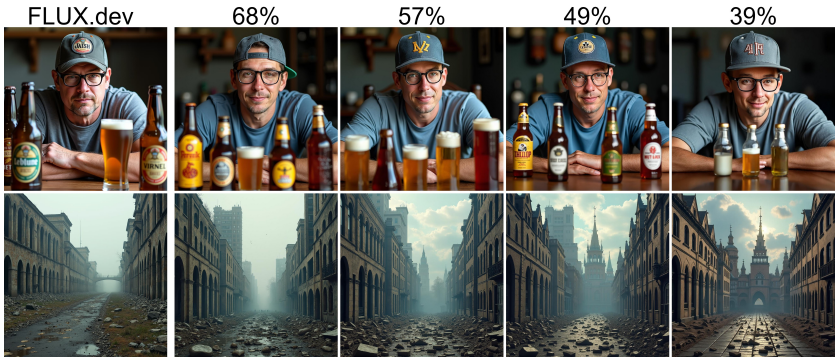

Fig. 2: Qualitative results of applying our SVDtrunc compression schema to FLUX.dev [19], ranging from 68 % to 39 % of the original parameter count. A minor degradation in quality is visible for 39 % of parameters.

Prior work has shown that different Transformer blocks contribute unevenly to image quality, and that sensitivity varies across network depth [2]. Building on this observation, we introduce a compression-based block importance estimation protocol, quantifying the functional contribution of individual blocks through controlled compression probing. We leverage this importance signal to allocate rank reduction non-uniformly given a global parameter budget. This enables to preserve critical components while compressing redundant ones more aggressively. Combined with modular knowledge distillation and rectified-flow training, this importance-aware rank allocation enables stable low-rank compression that preserves high representational capacity without altering the architecture.

Our contributions can be summarised as follows: *(1)* We show that Diffusion Transformers exhibit strong intrinsic tolerance to projection-level low-rank approximation. In contrast to observations in autoregressive LLMs, truncated SVD induces gradual rather than catastrophic degradation, even in a training-free setting. *(2)* We provide empirical evidence that redundancy in DiTs can be exploited across projection matrices, as distributed low-rank compression preserves functionality significantly more gracefully than coarse block removal. *(3)* We introduce a block-level compression probing protocol for estimating block sensitivity and use it for importance-aware rank allocation, improving quality retention under matched budgets. *(4)* Our SVDtrunc establishes a new quality-parameter frontier for compressed DiTs and show that low-rank tolerance persists after step distillation.

2 Related Work

Text-to-Image Generators. Text-to-image generation has evolved from U-Net-based [31] latent diffusion models [30] to large-scale Diffusion Transformers (DiTs) [3, 26, 41]. Early latent diffusion architectures such as SDXL [27]

reached approximately 2.6 B parameters, while recent DiT-based models, including FLUX [19] and Qwen-Image [39], scale from 8 to 20 B parameters. Although Transformer scaling improves generation quality and semantic fidelity, such model sizes impose substantial memory and runtime constraints, making tuning and deployment challenging without specialised engineering. Reducing computational and memory cost has, therefore, become a central research topic.

Temporal Distillation. Diffusion and rectified-flow models generate images iteratively over tens of steps. A large body of work focuses on compressing this trajectory by reducing the number of model evaluations [16,21,32,33]. Step distillation and consistency-based methods commonly compress multi-step generation into 4–8 steps, substantially lowering latency. However, such temporal distillation leaves the backbone architecture unchanged. The parameter count and VRAM footprint remain identical, and inference cost is fundamentally bounded by single-step evaluation. Consequently, temporal distillation does not address memory constraints and is complementary to structural compression.

Structural Compression. Several works directly modify diffusion backbones to reduce parameters. EcoDiff [45] introduces differentiable neuron masking with an end-to-end objective spanning all denoising steps. It removes attention heads and FFN neurons via hard-concrete relaxation and achieves up to 20 % sparsity on FLUX with lightweight mask optimisation. Pruned neurons are permanently removed from all forward passes, yielding structural sparsity and reduced memory. HierarchicalPrune [18] performs block-level removal based on a position-aware sensitivity analysis, reducing model depth. PPCL [24] identifies redundant layer intervals and prunes or replaces them during inference, reducing memory and computation by eliminating contiguous representational stages. FastFLUX [2] applies branch-level substitution: instead of deleting blocks, it replaces the residual branch of selected ResBlocks with lightweight linear layers while preserving shortcut connections, combining architectural modification with localised re-training. Dense2MoE [46] converts dense DiTs into mixture-of-experts architectures, reducing the number of activated parameters per forward pass. While this lowers effective computation, the total parameter count remains largely unchanged, and sparse routing introduces additional gating networks and training complexity.

Across these approaches, compression is achieved by removing or replacing architectural components (neurons, heads, residual branches, or entire blocks), thereby altering network structure and reducing representational capacity at a coarse granularity.Under aggressive compression regimes, such structural interventions may lead to abrupt performance degradation due to irreversible capacity reduction. In contrast, we use low-rank approximation to compress selected projection matrices while preserving the original architectural structure and pretrained capabilities for all blocks, allowing for granular control.

Low-Rank Compression. Low-rank factorisation has been widely explored for CNNs and large language models (LLMs) [12, 23, 28, 37, 38, 42]. In LLMs, SVD-based compression and related techniques are known to be highly sensitive. Naive truncated SVD can even lead to catastrophic failure [38], requiring

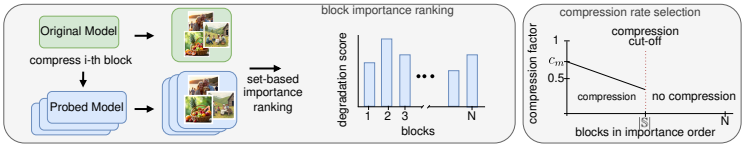

Fig. 3: SVDtrunc Overview. *(left)* We first generate N probed models, by individually compressing each of the N blocks. Then, each probed model generates a set of images. By comparing these sets with the generations of the original model, we obtain a quality-based ranking of block importance. We use this ranking *(right)* to assign importance-aware compression factors to each block of the final compressed model.

carefully tuned truncation schedules [12, 23]. In contrast, we find that for Diffusion Transformers truncated SVD-based compression is not sensitive. Applying SVD compression of projection matrices yields graceful degradation even at aggressive compression rates while preserving the original architecture.

3 Importance-Aware Low-Rank Distillation

We base SVDtrunc on two hypotheses: *(1)* Individual building blocks of the network are not fully redundant. Removing an entire block eliminates its functionality, which must then be compensated by the remaining blocks and may limit their performance. *(2)* Building blocks do not contribute equally to the final generated image, implying that the degree of parameter reduction should vary across the network. Both hypotheses are empirically validated in Sec. 4.

Typical parameter reduction methods prune entire Transformer blocks [2, 47], a highly aggressive form of parameter reduction that violates our first hypothesis. Instead, to preserve block functionality, we compress individual blocks using truncated singular value decomposition (SVD) rather than removing them entirely. To address our second hypothesis, we introduce a block-importance-aware schedule that assigns different compression rates to blocks based on their estimated contribution to generation quality. Finally, we employ a modular knowledge distillation (MKD), similar to [14], to reproduce the features of all teacher modules. Figure 3 provides an overview of the truncated SVD procedure.

3.1 Block Importance Analysis

Previous work has studied how different parts of generative networks contribute to the generation and understanding of concepts and styles [5, 18, 43]. In the context of compressing pre-trained text-to-image models, we focus on the importance of individual network blocks for the final output. To this end, we randomly sample K images from an open-world dataset, caption them using a trained image captioner [9], and generate one image per caption with the original, uncompressed model. We then probe the network by compressing one block at a time while keeping all others unchanged (compression details are given below).

Fig. 4: Ranking metric justification. Depending on which block is compressed, the generated image of the compressed model *(right)* can deviate from the image generated by the uncompressed one *(left)*. However, the pixel-wise change is not necessarily correlated with quality, making pixel-wise similarity scores less suitable for block ranking.

Images are generated again with identical prompts and seeds. For each of the N probed blocks, we compare the quality of the resulting images to the original images using a quality metric $\mathcal{M}$.

Since all images are generated under identical conditions, the generated sets contain corresponding images. This admits two choices for the metric $\mathcal{M}$: pairwise measures that compare each probing image with its original counterpart, *e.g.* LPIPS [44], or set-level measures such as FID [11] or CMMD [15]. We empirically observe that compressing certain blocks can alter attributes such as pose or appearance, which affects pairwise scores like LPIPS. Figure 4 illustrates this effect for two compressed blocks of FLUX.dev [19]. As such changes do not necessarily imply lower image quality, we argue that distribution-based metrics like CMMD or FID, which compare feature distributions across sets, are better suited for our importance analysis. Due to its greater stability with respect to the number of images, we adopt CMMD over FID for evaluation. This procedure yields a ranking of blocks according to their importance for generation quality, enabling stronger compression of blocks whose modification causes the least degradation.

3.2 Importance-Aware Low-Rank Compression

Similar to previous approaches that aim to adapt or modify model architectures in a minimally destructive way to increase efficiency [2, 24, 43], we reduce the parameter size of the model by reducing parameters on a whole model-block level. This means for modern DiT-based architectures [7, 19, 20, 22, 39] that the considered blocks are single- and double-stream blocks, as shown in the supplement. These blocks are constructed from a variety of different projection layers and normalisations. When we reduce the parameter size, we always compress all of the projections within a block with the same block-dependent compression ratio. However, unlike other approaches [2, 24], we explicitly do not remove or substitute the whole block through a linear layer [2], in order to retain the nonlinear interactions within it.

For model compression, we propose to use truncated SVD, removing the least dominant singular values. Let $W \in \mathbb{R}^{n \times m}$ be a linear projection matrix.

We compute its SVD and retain only the top r singular values and vectors for our truncated approximation as

$$W = U\Sigma V^T \approx U_r \Sigma_r V_r^T = BA, \tag{1}$$

where $B = U_r\sqrt{\Sigma_r} \in \mathbb{R}^{n\times r}$ and $A = \sqrt{\Sigma_r}V_r^T \in \mathbb{R}^{r\times m}$ for numerical stability. With the original parameter count $P_0 = mn$ and the truncated parameter count $P_r = r(m+n)$, the compression ratio is $c = 1 - P_r/P_0$. Imposing a target compression ratio $c \in (0,1)$ gives $r(n+m) \leq (1-c)nm$, which determines the rank for the truncation as

$$r \leq \frac{(1-c)nm}{n+m}. \tag{2}$$

In practice, r is chosen as the largest integer satisfying this constraint. Since not all transformer blocks contribute equally to the final model quality [5, 18, 43], we compress only a subset of blocks. Let the model consist of blocks indexed by $i \in \{1,\ldots,N\}$ and let $\mathbb{S} \subseteq \{1,\ldots,N\}$ denote the subset of blocks selected for compression according to the importance ranking described in Sec. 3.1. Each block i contains projection matrices with total parameter count P_0^i. For compressing a block $i \in \mathbb{S}$, we apply truncated SVD with a block-specific compression ratio $c_i \in (0,1)$, while all blocks $i \notin \mathbb{S}$ remain unchanged.

For a compressed block $i \in \mathbb{S}$, the parameter count changes after compression as $P_0^i \longrightarrow (1-c_i)P_0^i$. For the block-specific rank-assignment, let the compressed blocks be ordered according to increasing importance. We assign compression ratios using a linearly decreasing schedule

$$c_i = c_m - \alpha \ \text{order}(i), \tag{3}$$

with c_m being the largest compression of a block and $\alpha > 0$ the slope parameter. Blocks with higher importance receive less compression (*i.e.* smaller c_i). We choose the slope α such that we achieve the desired global parameter budget

$$P \coloneqq \frac{\sum_{i\notin\mathbb{S}} P_0^i + \sum_{j\in\mathbb{S}}(1-c_j)P_0^j}{\sum_k P_0^k}. \tag{4}$$

See Figure 3 (right) for an illustration. By compressing only a subset of all blocks, we maintain potentially critical blocks unaltered and only focus on compressing blocks with the highest expected redundancy.

3.3 Fine-Tuning

We now fine-tune the compressed student model f_θ with two complementary objectives in mind: *(1)* preserving task performance for text-to-image generation, and *(2)* transferring knowledge from the uncompressed teacher f.

Rectified Flow Objective. To preserve generative performance, we train the student with the same rectified-flow objective as the teacher. Following

FLUX, we train the student with the rectified flow velocity prediction objective. Let $x \sim p_{\text{data}}, \epsilon \sim \mathcal{N}(0, I)$, and $t \sim \mathcal{U}(0, 1)$. We define the linear interpolation $z_t = (1-t)x + t\epsilon$ with target velocity $v^* = \epsilon - x$. The student model predicts a velocity field $v_\theta(z_t, t, y)$ conditioned on text y and we optimise

$$\mathcal{L}_{\text{RF}} = \mathbb{E}_{x,\epsilon,t,y}\left[\left\|v_\theta(z_t, t, y) - (\epsilon - x)\right\|_2^2\right]. \tag{5}$$

Modular Knowledge Distillation. To leverage the uncompressed model's knowledge, we additionally employ knowledge distillation. Instead of simply calculating the loss on the final generated images [32, 33], we use a more complete variant. In contrast to substituting or pruning of whole blocks, we apply redundancy-ranked compression, which preserves the block structure of the original model. This enables us to perform module-wise alignment of intermediate representations, similar to [14]. The modular knowledge distillation (MKD) loss combines velocity-level distillation

$$\mathcal{L}_{\text{KD}} = \mathbb{E}_{x,\epsilon,t,y}\left[\left\|v_{\text{T}}(z_t, t, y) - v_\theta(z_t, t, y)\right\|_2^2\right] \tag{6}$$

where $v_{\text{T}}(z_t, t, y)$ is the prediction of the teacher, with module-wise feature distillation. Let f^i and f_θ^i denote the i-th transformer block of the uncompressed teacher and the compressed student, respectively, and let

$$h_i = f^i(h_{i-1}; t, y), \qquad \hat{h}_i = f_\theta^i(\hat{h}_{i-1}; t, y), \qquad h_0 = \hat{h}_0 = \text{embed}(z_t, y), \tag{7}$$

be the intermediate feature maps. We penalise deviations of intermediate features after each block with

$$\mathcal{L}_{\text{F}} = \sum_i \lambda_i \mathbb{E}_{x,\epsilon,t,y}\left[\text{MSE}(h_i, \hat{h}_i)\right]. \tag{8}$$

We set λ_i to equalise the magnitude of block-wise contributions. The MKD loss is then given as

$$\mathcal{L}_{\text{MKD}} = \lambda_{\text{F}}\mathcal{L}_{\text{F}} + \lambda_{\text{KD}}\mathcal{L}_{\text{KD}}, \tag{9}$$

with the complete fine-tuning objective being

$$\mathcal{L} = \lambda_{\text{RF}}\mathcal{L}_{\text{RF}} + \mathcal{L}_{\text{MKD}} = \lambda_{\text{RF}}\mathcal{L}_{\text{RF}} + \lambda_{\text{F}}\mathcal{L}_{\text{F}} + \lambda_{\text{KD}}\mathcal{L}_{\text{KD}}. \tag{10}$$

4 Experiments

We first provide the experimental setup in Sec. 4.1. Before presenting the main comparison to the state of the art in Sec. 4.4, we present two experiments that motivate our overall methodology. First, in Sec. 4.2 we validate that redundancy in DiTs is distributed across projection matrices throughout the whole network, rather than at the level of entire Transformer blocks. Second, in Sec. 4.3 we show that purely applying truncated SVD to DiTs without re-training does not result in catastrophic degradation as for LLMs. The section concludes with an application to a temporally-distilled model in Sec. 4.5 and an ablation study in Sec. 4.6.

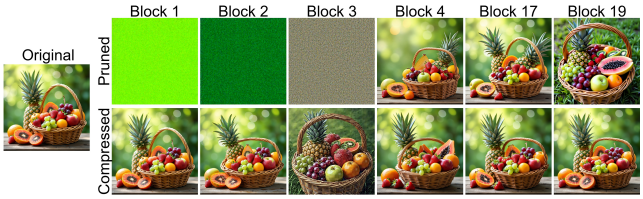

Fig. 5: Block sensitivity comparison. Generated images after probing double-stream blocks of FLUX.dev [19] with pruning *(top) vs.* truncated SVD *(bottom)*.

4.1 Experimental Setup

Generative Model. We compress FLUX.dev [19] using our SVDtrunc approach. This preserves the original block structure and non-linearities. The VAE and text-encoder are kept frozen and are not included in reported parameter budgets. We report the remaining parameter budget P with respect to the DiT backbone parameters for the Diffusion Transformer weights only.

Importance Analysis. To estimate block importance for our final models (Sec. 3.1), we sample $K_{\text{probe}} = 1000$ images from LAION [34] and generate captions using JoyCaption [9]. For each candidate block i, we construct a probed model by compressing only block i with a fixed block compression ratio of $c_m = 0.6$ while keeping all other blocks uncompressed, and generate paired image sets using identical prompts.

Importance metric. We quantify degradation using CMMD [15] between the probed and uncompressed image sets, and rank blocks by their induced degradation. Lower degradation suggests higher redundancy, hence we perform stronger compression.

Fine-tuning Settings. For distillation, we train on $K_{\text{train}} = 190\,000$ images generated by FLUX.dev conditioned on captions from JoyCaption on LAION images. We optimise the student using the combined objective in Eq. (10). Fine-tuning is done for 40 k steps with a batch size of 4 using Adafactor [35] with learning rate 1.8×10^{-6} on one H200 GPU. This is used to achieve a parameter budget of 68 %, as in alternative approaches [24, 29, 36]. To compress further, we re-evaluate the block importance and train subsequently for 20 k more steps.

Compression configuration. Unless stated otherwise, we compress $|\mathbb{S}| = 30$ blocks using variable maximum block-compression ratios c_m, which we specify in the supplement, and determine the linear schedule slope α to satisfy the target global budget P (Eq. 4).

Evaluation. We report results on three public benchmarks: GenEval [10], which evaluates object-centric compositional text–image alignment; DPG [13], which measures faithfulness to dense prompts; and HPSv2 [40], which combines prompt–image alignment with approximated human preference. All metrics are evaluated using the same inference protocol, *i.e.* a resolution of 1024×1024, 50 steps, classifier-free guidance scale 3.5, and fixed seeds per prompt. We additionally report peak VRAM usage with bf16 precision under the same settings.

Method	P[%] ↓	GenEval↑	HPSv2↑	DPG↑	R[%]↓
FLUX.dev [19]	100	0.647	31.70	83.9	0.00
EcoDiff [45]	90	0.452	29.67	78.0	14.52
Pruning	90	**0.646**	**31.02**	**83.3**	**1.00**
SVDtrunc (ours)	90	0.638	29.03	78.9	5.24
EcoDiff [45]	80	0.187	23.63	49.1	45.99
Pruning	80	0.381	24.88	65.2	28.31
SVDtrunc (ours)	80	**0.617**	**28.86**	**78.4**	**6.69**
EcoDiff [45]	70	0.002	7.26	17.0	85.50
Pruning	70	0.000	4.37	2.2	94.52
SVDtrunc (ours)	70	**0.580**	**28.48**	**76.6**	**9.86**

Table 1: Training-free compression. While importance-aware Pruning and EcoDiff collapse even under moderate compression, our SVDtrunc maintains stable generation quality even at 70% of the original parameter count.

For clarity, we report the average quality reduction across those benchmarks in relation to the respective uncompressed model as R.

4.2 Block Sensitivity Probing

We first investigate how Diffusion Transformers react to structural changes at the block level. For each block i, we consider two variants: *(1)* Pruning, where the entire block is replaced by an identity mapping, similarly to [17], and *(2)* single-block compression, where block i is compressed using truncated SVD with fixed block-compression ratio $c_m = 0.6$, while all other blocks remain unchanged. Furthermore, the blocks are not fine-tuned at this stage. For each probed model, we generate images using the same prompts as the uncompressed model. We quantify degradation using CMMD between the generated image sets, while the CMMD score is normalised by the number of reduced parameters.

Pruning *vs*. Compression. Fig. 5 shows that pruning frequently leads to severe degradation and, for early blocks, catastrophic failure of the generation process. In contrast, single-block compression results in substantially smaller degradation across all blocks. Quantitatively, pruning yields a mean CMMD of 1.18 ± 3.36 per probed block, whereas block compression achieves 0.23 ± 0.65. This indicates that rank reduction preserves functional behaviour far more gracefully than removing entire blocks. In Sec. 4.3, this trend is further confirmed by comparing our compression schedule with pruning under equal parameter budgets in the training-free setting. These results suggest that redundancy is distributed across projection matrices within blocks throughout the network, while complete block removal disrupts essential transformations. We use this setup to derive the block-importance ranking used in Eq. (3).

4.3 Training-Free Model Compression

To evaluate the intrinsic behaviour of Diffusion Transformers under low-rank approximation, we evaluate our importance-aware SVD compression without any

Method	Venue	Params. [%] ↓	VRAM [%] ↓	GenEval ↑	HPSv2 ↑	DPG ↑	R [%]↓	Rank ↓
FLUX.dev [19]	–	100	100.0	0.647	31.70	83.9	0.00	–
Chroma-HD [29]	HF	75	82.5	0.593	–	**84.0**	4.09	3
FluxLite [36]	HF'24	68	78.8	0.523	**31.31**	79.3	8.61	5
TinyFusion [8]	CVPR'25	68	74.4	0.511	–	77.2	14.48	6
Dense2MoE [46]	ICCV'25	75	–	0.403	–	73.6	24.97	9
FastFlux [2]	AAAI'26	80	–	0.530	27.26	–	16.04	8
PPCL [24]	CVPR'26	68	74.4	0.605	–	80.0	5.55	4
Hier.Prune [18]	AAAI'26	68	74.4	0.503	–	75.7	15.99	7
EcoDiff [45]	ICLR'26	80	–	0.399	25.99	–	28.17	10
SVDtrunc-m	–	68	78.2	**0.645**	31.27	83.4	**0.75**	1
SVDtrunc-s	–	**57**	**70.7**	0.616	31.22	82.6	2.60	2

Table 2: Quantitative results for compression of FLUX.dev [19]. The final Rank is based on the average quality reduction R. Our medium-sized model SVDtrunc-m achieves first rank. Notably, our small-sized model SVDtrunc-s, which has considerably fewer parameters than all alternative approaches, is runner-up.

Method	Params. [%] ↓	GenEval ↑	HPSv2 ↑	DPG ↑	R [%] ↓
PixArt-Σ [3]	100 (0.6B)	0.535	30.50	78.8	0.00
SVDtrunc	90	0.543	29.34	78.5	0.22
SVDtrunc	80	0.528	29.79	78.0	1.62
SVDtrunc	70	0.532	29.64	78.2	1.38
SVDtrunc	60	0.528	29.34	78.0	2.04
SVDtrunc	50	0.496	28.28	76.6	5.79

Table 3: Results for PixArt-Σ model. The degradation in quality of our SVDtrunc results are gradual, without abrupt failure.

subsequent fine-tuning. In this setting, truncated SVD is applied once according to the importance-guided schedule from Eq. (3), and the resulting model is evaluated directly. The results are reported in Tab. 1. At 90 % parameter count, our SVDtrunc degrades only 5.24 % in average performance, compared to 14.52 % for EcoDiff, a state-of-the-art training-free pruning method. At 80 %, SVDtrunc degrades by 6.69 %, whereas EcoDiff drops by 45.99 %. At approximately 70 % parameters, EcoDiff collapses almost entirely with $R = 85.50\,\%$, while our method drops only by $R = 9.86\,\%$.

These results indicate that low-rank compression on projection-matrix level preserves the functional structure of Diffusion Transformers to a surprising extent, even without fine-tuning, while leading to potential catastrophic failure in LLMs [12,38]. In contrast, pruning-based removal of blocks [17] leads to high performance at minimal parameter reduction but rapidly collapses ($R = 94.52\,\%$) under a moderate parameter budget of 70 %. While fine-tuning further improves alignment with the uncompressed teacher (Sec. 4.4), these findings show that the effectiveness of our approach is not solely driven by fine-tuning, but reflects an inherent tolerance of Diffusion Transformers to structured rank reduction.

4.4 Comparison to State-of-the-Art

Quantitative Results. We compare our approach with recent compression methods at matched parameter budgets after fine-tuning in Tab. 2. Unless stated

otherwise, models are re-evaluated under a unified inference protocol as described in Sec. 4.1. For TinyFusion [8], Dense2MoE [46], Chroma-HD [29], and HierarchicalPrune [18], we report numbers from PPCL [24] due to the absence of publicly available checkpoints or inference code.

At a parameter budget of 68 %, our model (SVDtrunc-m) achieves near full performance across GenEval, HPSv2, and DPG, with a degradation of only $R = 0.75\,\%$ compared to FLUX.dev. Notably, this exceeds competing methods at the same (or even larger) parameter budgets, while preserving strong compositional reasoning on GenEval. At a more aggressive 57 % parameter budget, our model (SVDtrunc-s) maintains competitive performance, with only $R = 2.6\,\%$. Despite operating at a substantially smaller parameter footprint than competing approaches at 68 %, it outperforms them on GenEval and remains competitive on HPSv2 and DPG. These results establish a new quality–parameter frontier for compressed Diffusion Transformers, demonstrating that importance-aware low-rank distillation enables substantial parameter reduction without catastrophic degradation. Full quantitative tables are provided in the supplement.

Memory Efficiency. In addition to parameter count, we report peak VRAM usage under identical inference settings. SVDtrunc achieves competitive or superior memory efficiency compared to pruning-based methods at comparable parameter budgets, highlighting the practical benefits of truncated SVD.

Progressive Compression. To analyse the trade-off between parameter budget and generation quality, we evaluate models across a range of global parameter budgets $P \in [0.9, 0.4]$. The results are shown in Fig. 1. For mild compression (*e.g.*, $P \geq 0.68$), performance remains nearly indistinguishable from the uncompressed model across all evaluated metrics, with $< 1\,\%$ quality reduction on average. As compression increases, quality degrades gradually and predictably. Notably, substantial performance deterioration only becomes apparent beyond approximately 40–50 % parameter reduction. This indicates that DiTs contain significant structured redundancy that can be removed without immediate degradation. However, beyond a certain compression level, the reduction in representational capacity begins to affect generation quality more noticeably. Since FLUX.1-dev contains 12B parameters, its robustness to compression could partly stem from overparameterization. To test whether this behavior extends beyond large-scale models, we additionally evaluate SVDtrunc on PixArt-Σ [3] in Tab. 3. PixArt-Σ shows the same quantitative trend of graceful degradation under increasing compression, suggesting that the redundancy exploited by SVDtrunc is not specific to FLUX.1-dev but is present across pretrained DiTs of different sizes and architectures.

Qualitative Results and Domain Drifts. In all our tests, our compressed models remain competitive with, or outperform, existing compression approaches. Qualitative examples are shown in Fig. 2. In visual inspection, we noticed an occasional domain drift, predominatly at stronger compression levels where the generated images move towards slightly different stylistic domains while remaining visually coherent. Fig. 6 illustrates such a domain drift from cartoonish style to realistic portrait. Importantly, this stylistic drift is not con-

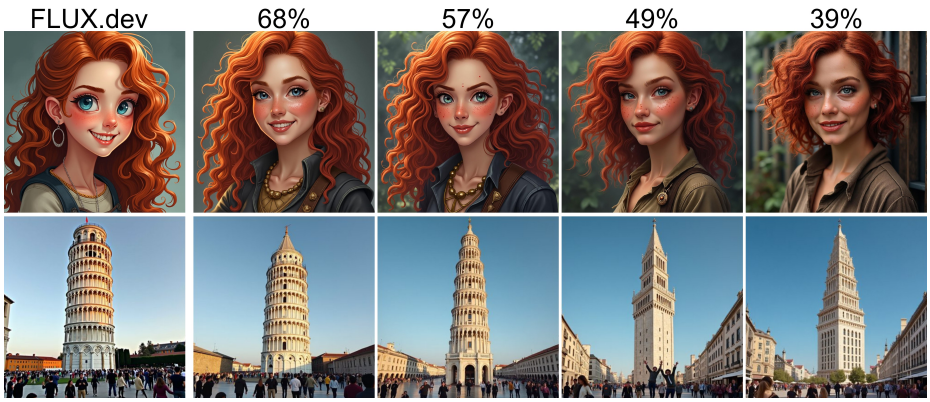

Fig. 6: Example of domain drift (top) and memorisation loss (bottom) under different parameter budgets.

sistently accompanied by a measurable degradation in visual quality and benchmark scores remain relatively stable. We conjecture that this behaviour indicates that SVDtrunc preserves the parameters most critical for generation quality, while stylistic characteristics are more susceptible to changes under compression. As a result, the model retains its core generative capabilities even under substantial parameter reduction.

Beyond the stylistic domain drift, decreasing the parameter budget can reduce the model's ability to reproduce specific characters or highly distinctive concepts. In Fig. 6, we show examples illustrating this behaviour. Well-known instances such as the tower of Pisa (Fig. 6) or Bugs Bunny, Darth Vader, or the Minions (supplement) are no longer reproduced faithfully at stronger compression levels. Instead, the model generates high-quality images resembling the general concept but lacking the distinctive identity of the original entity.

This behaviour suggests that part of the model capacity is used to memorise specific instances or highly detailed concepts, whereas general visual quality can be maintained with substantially fewer parameters. Intuitively, generating plausible images primarily requires learning reusable visual features and compositional patterns, whereas representing highly specific identities or characters requires additional capacity. Consequently, under strong compression, models may begin to discard parameters associated with memorised instances while retaining those necessary for general image synthesis. This observation indicates that substantial parameter redundancy in large diffusion transformers may be related to storing highly specific concepts rather than the core mechanisms required for image generation. We refer to the supplement for further examples.

4.5 Application to Temporally-Distilled Model

Step distillation reduces the number of denoising steps, compressing the generation process along the temporal dimension [21, 32, 33]. One might therefore

Method	P[%] ↓	GenEval↑	HPSv2↑	DPG↑	R[%]↓
FLUX.schnell [19]	100	0.667	30.85	85.0	0.00
SVDtrunc (ours)	90	0.666	29.59	85.2	1.34
SVDtrunc (ours)	80	0.668	29.48	84.9	1.48
SVDtrunc (ours)	70	0.648	29.30	84.8	2.72
SVDtrunc (ours)	60	0.621	29.05	84.3	4.53

Table 4: Results for temporally-distilled FLUX.schnell [19] model. The degradation in quality of our SVDtrunc results are gradual, without abrupt failure.

expect temporally distilled models to exhibit reduced internal redundancy and increased sensitivity to parameter compression. We test this hypothesis on the official temporally-distilled model FLUX.schnell [19], which generates images in four steps. We recompute the block-importance ranking and apply projection-level low-rank compression using the same importance-aware allocation strategy from Sec. 3.1. Results are summarised in Tab. 4. At a 90 % parameter budget, GenEval performance remains virtually unchanged and DPG stays comparable. Even at 80 %, the compressed model maintains near-baseline performance across all metrics. As compression increases to 70 % and 60 %, degradation remains gradual and controlled, without abrupt failure. However, the degradation is slightly higher, with $R = 4.53\,\%$ at $P = 60\,\%$, compared to FLUX.dev with $R = 2.60\,\%$ at $P = 57\,\%$ (see Tab. 2). These results indicate that tolerance to projection-level low-rank compression persists even after temporal distillation. Redundancy therefore appears to be structurally embedded in the Transformer backbone rather than arising solely from multi-step diffusion dynamics.

4.6 Ablations

We ablate the rank-allocation strategy, which is the central design choice of SVDtrunc. This experiment analyses how importance-aware rank allocation affects generation quality under decreasing parameter budgets. Additional ablations on the distribution of compression across blocks and the fine-tuning objective are provided in the supplement.

Importance-Aware *vs*. Uniform Allocation. Tab. 5 compares uniform, random block, and importance-aware compression across multiple parameter budgets. With uniform, we compress all blocks equally, while the random schedule has a random ranking order for Eq. (3). All three strategies exhibit smooth degradation as compression increases, indicating that Diffusion Transformers tolerate projection-level rank reduction even under naive allocation. At moderate compression (68 %), performance differences remain small, with SVDtrunc retaining near-teacher performance. As compression becomes more aggressive (57 % and 49 %), importance-aware allocation consistently outperforms both uniform and random schedules across GenEval, HPSv2, and DPG, yielding improved quality retention and lower relative degradation. These results indicate that redundancy is broadly distributed across the network, enabling stable low-

Method	P[%] ↓	GenEval↑	HPSv2↑	DPG↑	R[%]↓
FLUX.dev	100	0.647	31.70	83.9	0.00
Random	68	0.633	31.19	82.9	1.63
Uniform	68	0.633	**31.38**	82.8	1.46
Ours	68	**0.645**	31.27	**83.4**	**0.75**
Random	57	0.588	30.22	81.1	5.68
Uniform	57	0.613	30.92	82.3	3.19
Ours	57	**0.616**	**31.22**	**82.6**	**2.60**
Random	49	0.542	29.16	79.8	9.69
Uniform	49	0.582	29.96	81.2	6.23
Ours	49	**0.587**	**31.01**	**81.7**	**4.68**

Table 5: Compression Schedule Ablation. We evaluate uniform, random and importance-aware (our) SVD compression schedules after training.

rank compression even under uniform allocation, while importance-aware rank allocation becomes increasingly beneficial under stronger constraints.

5 Summary and Conclusion

We investigated truncated SVD for compressing large-scale Diffusion Transformers and find that, unlike autoregressive LLMs, DiTs exhibit substantial tolerance to low-rank approximation on the projection-matrix level. Even under significant parameter reduction, generative performance degrades gradually rather than catastrophically, and this behaviour is already observable in a training-free setting. Across benchmarks, our approach achieves a favourable quality–parameter trade-off and remains robust even when applied to models after temporal distillation, positioning low-rank compression on the projection-matrix level as a promising direction for further efficiency gains. At aggressive compression levels, domain drift can occur, highlighting limits of global representational capacity and motivating future work on preserving stylistic diversity under stronger parameter constraints.

Acknowledgments

Denis Zavadski and Sebastian Heid are supported by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA) through the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the German Federal Ministry of Education and Research. The authors gratefully acknowledge the support by the Ministry of Science, Research and the Arts Baden-Württemberg (MWK) through bwHPC, SDS@hd and the German Research Foundation (DFG) through the grants INST 35/1597-1 FUGG and INST 35/1503-1 FUGG. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 866008), and from the DFG under Germany's Excellence Strategy (EXC-3057/1 "Reasonable Artificial Intelligence", Project No. 533677015). We thank Tim Küchler for the discussions.

References

1. Ben Noach, M., Goldberg, Y.: Compressing pre-trained language models by matrix decomposition. In: A.-P. Assoc. Comput. Ling. and the 10th International Joint Conference on Natural Language Processing. pp. 884–889 (2020) 2
2. Cai, F., Guo, Y., Li, J., Li, W., Chen, J., Fang, X.: FastFLUX: Pruning FLUX with block-wise replacement and sandwich training. AAAI p. 2507–2515 (2026) 2, 3, 4, 5, 6, 11
3. Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., Li, Z.: PixArt-σ: Weak-to-strong training of diffusion transformer for 4K text-to-image generation. In: Eur. Conf. Comput. Vis. pp. 74–91. Springer (2024) 3, 11, 12
4. Chen, J., Xue, S., Zhao, Y., Yu, J., Paul, S., Chen, J., Cai, H., Han, S., Xie, E.: SANA-Sprint: One-step diffusion with continuous-time consistency distillation. In: Int. Conf. Comput. Vis. pp. 16185–16195 (2025) 1
5. Chen, P., Shen, M., Ye, P., Cao, J., Tu, C., Bouganis, C.S., Zhao, Y., Chen, T.: Δ-DiT: A training-free acceleration method tailored for diffusion transformers. arXiv preprint arXiv:2406.01125 (2024) 5, 7
6. Denton, E., Zaremba, W., Bruna, J., LeCun, Y., Fergus, R.: Exploiting linear structure within convolutional networks for efficient evaluation. In: Adv. Neural Inform. Process. Syst. vol. 27. Curran Associates, Inc. (2014) 2
7. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: Int. Conf. Mach. Learn. (2024) 6
8. Fang, G., Li, K., Ma, X., Wang, X.: TinyFusion: Diffusion transformers learned shallow. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 18144–18154 (2025) 2, 11, 12
9. fpgaminer: JoyCaption. https://github.com/fpgaminer/joycaption (2024) 5, 9
10. Ghosh, D., Hajishirzi, H., Schmidt, L.: GenEval: An object-focused framework for evaluating text-to-image alignment. In: Adv. Neural Inform. Process. Syst. vol. 36, pp. 52132–52152. Curran Associates, Inc. (2023) 2, 9
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Adv. Neural Inform. Process. Syst. vol. 30. Curran Associates, Inc. (2017) 6
12. Hsu, Y.C., Hua, T., Chang, S., Lou, Q., Shen, Y., Jin, H.: Language model compression with weighted low-rank factorization. In: Int. Conf. Learn. Represent. (2022) 2, 4, 5, 11
13. Hu, X., Wang, R., Fang, Y., Fu, B., Cheng, P., Yu, G.: ELLA: Equip diffusion models with LLM for enhanced semantic alignment. arXiv preprint arXiv:2403.05135 (2024) 2, 9
14. Huang, T., Zhang, Y., Zheng, M., You, S., Wang, F., Qian, C., Xu, C.: Knowledge diffusion for distillation. In: Adv. Neural Inform. Process. Syst. vol. 36, pp. 65299–65316. Curran Associates, Inc. (2023) 5, 8
15. Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., Kumar, S.: Rethinking FID: Towards a better evaluation metric for image generation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9307–9315 (2024) 6, 9
16. Kang, M., Zhang, R., Barnes, C., Paris, S., Kwak, S., Park, J., Shechtman, E., Zhu, J.Y., Park, T.: Distilling diffusion models into conditional GANs. In: Eur. Conf. Comput. Vis. pp. 428–447. Springer (2024) 4

17. Kim, B.K., Song, H.K., Castells, T., Choi, S.: BK-SDM: A lightweight, fast, and cheap version of Stable Diffusion. In: Eur. Conf. Comput. Vis. pp. 381–399. Springer (2024) 10, 11
18. Kwon, Y.D., Li, R., Li, S., Li, D., Bhattacharya, S., Venieris, S.I.: HierarchicalPrune: Position-aware compression for large-scale diffusion models. AAAI p. 22716–22724 (2026) 4, 5, 7, 11, 12
19. Labs, B.F.: FLUX. https://github.com/black-forest-labs/flux (2024) 2, 3, 4, 6, 9, 10, 11, 14
20. Li, S., Kallidromitis, K., Gokul, A., Liao, Z., Kato, Y., Kozuka, K., Grover, A.: OmniFlow: Any-to-any generation with multi-modal rectified flows. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 13178–13188 (2025) 6
21. Lin, S., Wang, A., Yang, X.: SDXL-Lightning: Progressive adversarial diffusion distillation. arXiv preprint arXiv:2402.13929 (2024) 1, 4, 13
22. Liu, B., Akhgari, E., Visheratin, A., Kamko, A., Xu, L., Shrirao, S., Lambert, C., Souza, J., Doshi, S., Li, D.: Playground v3: Improving text-to-image alignment with deep-fusion large language models. arXiv preprint arXiv:2409.10695 (2024) 1, 6
23. Liu, K., Zhang, Y., Cheng, N., Li, Z., Wang, S., Xiao, J.: GRASP: Replace redundant layers with adaptive singular parameters for efficient model compression. In: Conf. Empir. Meth. Nat. Lang. Process. pp. 26344–26359 (2025) 4, 5
24. Ma, J., Peng, Q., Zhu, X., Xie, P., Chen, C., Lu, H.: Pluggable pruning with contiguous layer distillation for diffusion transformers. arXiv preprint arXiv:2511.16156 (2025) 2, 4, 6, 9, 11, 12
25. Mirsky, L.: Symmetric gauge functions and unitarily invariant norms. Quart. J. Math. **11**(1), 50–59 (1960) 2
26. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Int. Conf. Comput. Vis. pp. 4195–4205 (2023) 1, 3
27. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: Improving latent diffusion models for high-resolution image synthesis. In: Int. Conf. Learn. Represent. (2024) 3
28. Qinsi, W., Ke, J., Tomizuka, M., Keutzer, K., Xu, C.: Dobi-SVD: Differentiable SVD for LLM compression and some new perspectives. In: Int. Conf. Learn. Represent. (2025) 2, 4
29. Rock, L.: Chroma1-HD. https://huggingface.co/lodestones/Chroma1-HD (2025) 9, 11, 12
30. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 10684–10695 (2022) 3
31. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Int. Conf. Med. Image Comput. Comput.-Assist. Interv. pp. 234–241. Springer (2015) 3
32. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: Int. Conf. Learn. Represent. (2022) 1, 4, 8, 13
33. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. In: Eur. Conf. Comput. Vis. pp. 87–103. Springer (2024) 1, 4, 8, 13
34. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C.W., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S.R., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: LAION-5B: An open large-scale dataset for training next generation image-text models. In: Adv. Neural Inform. Process. Syst. vol. 35, pp. 25278–25294. Curran Associates, Inc. (2022) 9

35. Shazeer, N., Stern, M.: Adafactor: Adaptive learning rates with sublinear memory cost. In: Int. Conf. Mach. Learn. pp. 4596–4604. PMLR (2018) 9
36. Verdú, D., Martín, J.: Flux.1 Lite: Distilling FLUX1.dev for efficient text-to-image generation (2024) 9, 11
37. Wang, X., Alam, S., Wan, Z., Shen, H., Zhang, M.: SVD-LLM V2: Optimizing singular value truncation for large language model compression. In: Nation. Am. Assoc. for Comput. Ling. pp. 4287–4296 (2025) 2, 4
38. Wang, X., Zheng, Y., Wan, Z., Zhang, M.: SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In: Int. Conf. Learn. Represent. (2025) 2, 4, 11
39. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-Image technical report. arXiv preprint arXiv:2508.02324 (2025) 1, 4, 6
40. Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., Li, H.: Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. arXiv preprint arXiv:2306.09341 (2023) 2, 9
41. Xie, E., Chen, J., Zhao, Y., YU, J., Zhu, L., Lin, Y., Zhang, Z., Li, M., Chen, J., Cai, H., Liu, B., Zhou, D., Han, S.: SANA 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. In: Int. Conf. Mach. Learn. (2025) 1, 3
42. Yuan, Z., Shang, Y., Song, Y., Yang, D., Wu, Q., Yan, Y., Sun, G.: ASVD: Activation-aware singular value decomposition for compressing large language models. arXiv preprint arXiv:2312.05821 (2023) 2, 4
43. Zavadski, D., Feiden, J.F., Rother, C.: ControlNet-XS: Rethinking the control of text-to-image diffusion models as feedback-control systems. In: Eur. Conf. Comput. Vis. pp. 343–362. Springer (2024) 5, 6, 7
44. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 586–595 (2018) 6
45. Zhang, Y., Jin, E., Liang, W., Dong, Y., Khakzar, A., Torr, P., Stegmaier, J., Kawaguchi, K.: Learnable sparsity for vision generative models. In: Int. Conf. Learn. Represent. (2026) 2, 4, 10, 11
46. Zheng, Y., Ren, Y., Xia, X., Xiao, X., Xie, X.: Dense2MoE: Restructuring diffusion transformer to moe for efficient text-to-image generation. In: Int. Conf. Comput. Vis. pp. 18661–18670 (2025) 4, 11, 12
47. Zhu, H., Tang, D., Liu, J., Lu, M., Zheng, J., Peng, J., Li, D., Wang, Y., Jiang, F., Tian, L., Tiwari, S., Sirasao, A., Yong, J., Wang, B., Barsoum, E.: DiP-GO: A diffusion pruner via few-step gradient optimization. In: Adv. Neural Inform. Process. Syst. vol. 37, pp. 92581–92604. Curran Associates, Inc. (2024) 5