
SlimDiff: Training-Free, Activation-Guided Hands-free Slimming of Diffusion Models

Arani Roy* **Shristi Das Biswas*** **Kaushik Roy**
 Purdue University
 {roy173, sdasbisw, kaushik}@purdue.edu

Abstract

Diffusion models (DMs), lauded for their generative performance, are computationally prohibitive due to their billion-scale parameters and iterative denoising dynamics. Existing efficiency techniques, such as quantization, timestep reduction, or pruning, offer savings in compute, memory, or runtime but are strictly bottle-necked by reliance on fine-tuning or retraining to recover performance. In this work, we introduce SlimDiff, an automated activation-informed structural compression framework that reduces both attention and feedforward dimensionalities in DMs, while being entirely gradient-free. SlimDiff reframes DM compression as a spectral approximation task, where activation covariances across denoising timesteps define low-rank subspaces that guide dynamic pruning under a fixed compression budget. This activation-aware formulation mitigates error accumulation across timesteps by applying module-wise decompositions over functional weight groups: query–key interactions, value–output couplings, and feedforward projections — rather than isolated matrix factorizations, while adaptively allocating sparsity across modules to respect the non-uniform geometry of diffusion trajectories. SlimDiff achieves up to 35% acceleration and $\sim 100M$ parameter reduction over baselines, with generation quality on par with uncompressed models without any backpropagation. Crucially, our approach requires only about 500 calibration samples, over $70\times$ fewer than prior methods. To our knowledge, this is the first closed-form, activation-guided structural compression of DMs that is entirely training-free, providing both theoretical clarity and practical efficiency.

1 Introduction

Diffusion models (DMs) (Rombach et al. (2022); Ramesh et al. (2022); Saharia et al. (2022)) have become the dominant paradigm in generative modeling, achieving remarkable performance. Their power, however, comes at a steep computational cost: every sample requires hundreds of denoising iterations, each iteration invoking a billion-parameter U-Net architecture (Dhariwal & Nichol (2021)). The sequential reliance on such high-dimensional operators leads to substantial latency, memory, and energy demands, making real-time or resource-constrained deployment prohibitive.

Structural slimming offers a direct avenue for reducing both parameters and MACs (Shen et al. (2025)), yet applying it to DMs exposes fundamental challenges that prior work has largely overlooked. Attention (Vaswani et al. (2017)), a primary building block of the diffusion U-Net, illustrates this difficulty: pruning individual weights neglects the coupled nature of effective computations, which arise from products like query-key interactions (QK), value-output couplings (VO), and feed-forward projections (FFN). Since the rank of a product is bounded by the smallest rank among its factors (Kolter (2007)), maximal compression can be attained only when these products are treated as functional units (Lin et al. (2024)). As shown in App A.6, DM weights are largely high-rank with heavy-tailed spectra, and truncation errors accumulate through sequential denoising. Effective compression is achieved when the weight structure aligns with activation correlations, which define

the active subspaces during denoising. Thus, optimal compression requires *data awareness* (Lin et al. (2024)). Ignoring these structural dependencies leads to subpar pruning decisions.

The second issue is that compressibility in DMs is inherently timestep-dependent. Activation correlations reveal a far richer low-rank structure than weights alone, but their covariance evolves dramatically over the denoising trajectory (Wang et al. (2024)). The activation distribution differs not only across timesteps but also across functional modules, each interacting with the weights in distinct ways. Approximating weights with a static, timestep-agnostic basis therefore collapses this evolving geometry and leads to poor preservation of fine details (Yao et al. (2024b)).

A third issue is error propagation. In diffusion, pruning errors are not local; distortions introduced at one step are passed through every subsequent denoising update. Small deviations in early layers compound multiplicatively, creating irreversible degradation (Zeng et al. (2025)). Existing methods allocate sparsity myopically, without accounting for this sequential amplification, and therefore, rely on costly fine-tuning or retraining with large datasets to recover lost performance. Such dependence on retraining undermines the very motivation for slimming. (Zhang et al. (2024a))

Finally, data-aware slimming requires collecting activations across timesteps and prompts (Lin et al. (2024)), and exhaustive sampling is computationally infeasible. Since naïve calibration over thousands of prompts is cumbersome, a principled strategy is needed to select a compact yet representative subset of prompts that spans the relevant activation subspace (Nguyen & He (2025)).

Prior works (Zhang et al. (2024b); Gao et al. (2024)) exemplify these limitations: although they reduce parameter counts, they remain functional module-agnostic and rely on finetuning or distillation on a large dataset to correct for timestep-dependent distortions and error propagation. Other approaches, such as (Lu et al. (2022)) and (Chen et al. (2025)), design compact models from scratch but at the cost of prohibitively expensive retraining. Alternatively, a parallel line of work orthogonally looks at inference-time accelerations such as (Wang et al. (2024); Bolya & Hoffman (2023); Fang et al. (2023)) that reduce computation by truncating denoising timesteps or merging tokens at run-time. These methods incur runtime overhead at every invocation and yield savings that fluctuate across runs - making both the effective cost and the achievable compression level unpredictable. Other methods explore quantization (Li et al. (2023); Zeng et al. (2025)) strategies to accelerate inference, which can be used in parallel with our structural slimming method.

In this paper, we introduce SlimDiff, the first training-free, activation-guided framework that addresses structural, temporal, and propagation-aware challenges of DM slimming in a unified platform. Our contributions are:

- **Principled compression design:** We introduce *module-aligned decompositions* that compress functionally related weight groups instead of isolated matrices, ensuring the compressed model remains structurally consistent with the diffusion computation graph.
- **Data- and process-aware compression:** To align compression with the dynamics of denoising, we propose *timestep-aware compressibility*, leveraging activation statistics stratified by timestep, and *propagation-aware rank allocation*, which globally distributes sparsity under an explicit model of error amplification.
- **Efficient calibration:** We design *SlimSet*, a compact semantic-aware calibration set of only 500 prompts—over 70× fewer than prior works—that spans representative compressible subspaces, making the entire activation collection pipeline lightweight and practical.
- **Comprehensive validation:** We evaluate SlimDiff on MS-COCO (Lin et al. (2014)), LAION Aesthetics (Schuhmann et al. (2022)), ImageReward (Xu et al. (2023)), and PartiPrompts (Yu et al. (2023)) across DMs SDv1.5 and SDv1.4 (Rombach & Esser (2022b,a)). SlimDiff reduces ~ 100M parameters, reduces FLOPs by 22%, and speeds up inference by 35% while preserving quality. We also confirm robustness via human preference scoring on HPS v2.1(Wu et al. (2023)), ImageReward, and Pic-a-Pic v1(Kirstain et al. (2023)).

2 Methodology

We aim to compress a pretrained stable diffusion model Θ into a compact model $\hat{\Theta}$ that satisfies a full parameter budget B while preserving output quality. Formally, we pose this as a constrained optimization problem:

$$\min_{\hat{\Theta}} \mathcal{L}_{\text{qual}}(\hat{\Theta}) \quad \text{s.t.} \quad \text{params}(\hat{\Theta}) \leq B \quad (1)$$

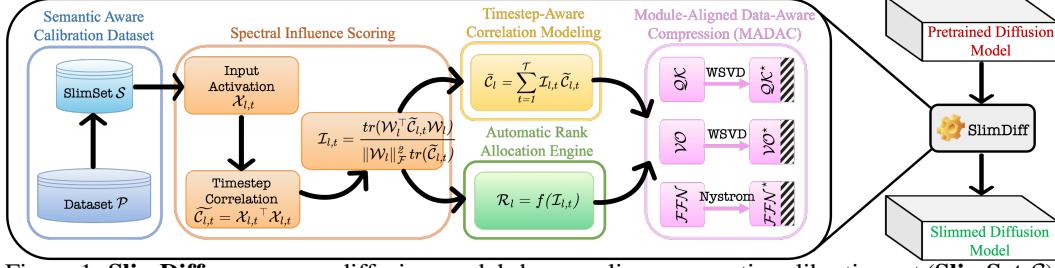


Figure 1: **SlimDiff** compresses diffusion models by sampling a semantic calibration set (**SlimSet** \mathcal{S}), Spectral Influence Scoring each module’s alignment with input anisotropy, which drives Timestep-Aware Correlation Modeling and an Automatic Rank Allocator under a global budget. Finally, MADAC applies whitening–SVD to $\mathcal{QK}/\mathcal{VN}$ and Nyström reduction to \mathcal{FFN} .

SlimDiff addresses this challenge through *structural, activation-guided compression*. Instead of pruning or factorizing individual matrices, it operates on functional weight groups (\mathcal{QK} , \mathcal{VN} , \mathcal{FFN}) and aligns compression with activation statistics across denoising steps. Figure 1 outlines four key components: **(i) Spectral Influence Scoring:** For each module, we quantify alignment with dominant activation directions to measure its relative importance (Sec. 2.1). **(ii) Semantic Calibration Dataset (SlimSet):** From the full prompt pool \mathcal{P} , we select a compact subset \mathcal{S} that spans activations across timesteps, using geometric-median clustering with furthest-point sampling (FPS) in embedding space (Nguyen & He (2025)) (Sec. 2.2). **(iii) Timestep-aware Correlation Modeling:** Using SlimSet activations, we compute per-timestep correlations for \mathcal{QK} , \mathcal{VN} , and \mathcal{FFN} modules. These are aggregated into fidelity-weighted mixtures, assigning greater weight to timesteps most salient to output quality (Sec. 2.3). **(iv) Module-Aligned Data-Aware Compression (MADAC) and Rank Allocation:** Aggregated correlations drive a modular compression objective with tailored decompositions for \mathcal{QK} , \mathcal{VN} , and \mathcal{FFN} blocks (Sec. 2.4). Influence scores then guide automatic per-layer rank selection under a user-specified parameter budget (Sec. 2.5). Together, these stages yield a closed-form, training-free pipeline that ‘slims’ diffusion models without performance degradation, delivering substantial reductions in parameters and FLOPs without retraining.

2.1 Anchoring Metric: Spectral Influence Score

At the core of SlimDiff is an anchoring metric we call the *Spectral Influence Score*, which quantifies how strongly each module aligns with the anisotropy of its input activations. This score serves as the foundation for both timestep-aware correlation accumulation and rank allocation, ensuring that compression decisions remain faithful to the evolving geometry of diffusion activations.

Formally, we adopt the trace-normalized Rayleigh quotient (TRQ) (Chen (2020)) as the spectral influence score. For a weight matrix per layer l , \mathcal{W}_l and pre-activation covariance $\tilde{\mathcal{C}}_{l,t}$ at timestep t , we define our influence score $\mathcal{I}_{l,t}$ as:

$$\mathcal{I}_{l,t} = \text{TRQ}_{l,t}(\mathcal{W}_l) = \frac{\mathbb{E} \nabla (\mathcal{W}_l^\top \tilde{\mathcal{C}}_{l,t} \mathcal{W}_l)}{\|\mathcal{W}_l\|_F^2 \mathbb{E} \nabla (\tilde{\mathcal{C}}_{l,t})} = \frac{\|\tilde{\mathcal{C}}_{l,t}\|_F^{1/2} \mathcal{W}_l}{\|\mathcal{W}_l\|_F^2 \text{tr}(\tilde{\mathcal{C}}_{l,t})} \quad (2)$$

This formulation is variance-invariant: normalizing by $\text{tr}(\tilde{\mathcal{C}}_{l,t})$ cancels stepwise scaling, and scale-invariant, as normalizing by $\|\mathcal{W}_l\|_F^2$ removes dependence on parameter norms. Importantly, it isolates directional alignment: in the eigenbasis of $\tilde{\mathcal{C}}_{l,t}$, the score evaluates how strongly \mathcal{W}_l projects onto high-variance directions, focusing on anisotropy rather than raw energy. Aggregating TRQ across timesteps with a convex mixture (Sec. 2.3) yields a single spectral influence score per module $\mathcal{I}_{l,t}$, which anchors our pipeline by guiding both Data-Aware Compression and Global Rank Allocation.

2.2 SlimSet: Semantic-Aware Calibration Dataset Formation

Activation-guided compression requires estimating correlations $\Sigma_{l,t} = \mathbb{E}[X_{l,t}^\top X_{l,t}]$ across timesteps and modules. However, collecting these statistics over the full prompt corpus \mathcal{P} is computationally expensive. We therefore introduce **SlimSet**, a semantic coresset \mathcal{S} that preserves the statistical geometry of \mathcal{P} while reducing calibration cost by more than $70\times$. Note that, SlimSet construction is lightweight, taking only a few minutes, and follows the semantic coresset selection strategy introduced in SCDP (Nguyen & He (2025)).

Semantic embedding. Each prompt $p_i \in \mathcal{P}$ is embedded into $E_i \in \mathbb{R}^d$ using CLIP, providing a semantic space where distances reflect prompt similarity. We compute the geometric median c of the embedding cloud, and assign each prompt a distinctiveness score $f_i = \|E_i - c\|_2$, so that prompts farther from the center capture more diverse semantics.

Bin allocation and sampling. To balance frequent and rare concepts, we stratify prompts into B quantile bins of $\{f_i\}$ and assign each bin a quota q_b proportional to corpus size. Within each bin, we apply farthest-point sampling (FPS): $\mathcal{S}_b = \arg \max_{|\mathcal{S}_b|=q_b} \min_{i \neq j \in \mathcal{S}_b} (1 - \cos(E_i, E_j))$, ensuring that selected prompts are well-spread and mutually diverse. We then perform cosine-based de-duplication to remove redundancy.

Resulting calibration set. The final SlimSet \mathcal{S} retains only $J \ll |\mathcal{P}|$ prompts ($J = 500$ vs. 35k), yet yields activation covariances satisfying $\Sigma \text{Cov}_{l,t}^{\mathcal{S}} \approx \Sigma \text{Cov}_{l,t}^{\mathcal{P}}$ across layers l and timesteps t . This compactness permits accurate estimation of module-level covariance structure at a fraction of the sampling cost, while preserving both semantic diversity and statistical fidelity. Fig. 3 and Sec. 4(i) detail the SlimSet size ablation and stability evaluation.

2.3 Timestep-Aware Correlation Modeling

DM activations evolve over the denoising trajectory, with correlation structure strongly tied to timestep. A single, timestep-agnostic covariance thus misrepresents the statistics: early steps are near-isotropic noise, while later ones show anisotropic, perceptually aligned variance (App. Sec A.5, Fig. 4). To capture this, we compute *timestep-aware correlations* per module and form a fidelity-weighted mixture emphasizing steps relevant to output quality. For activations $x_{l,t} \in \mathbb{R}^d$ collected at layer l and timestep t with N_t samples, we form the second moment $\hat{\mathcal{C}}_{l,t} = \frac{1}{N_t} \mathcal{X}_{l,t}^\top \mathcal{X}_{l,t}$. Next, to ensure numerical stability (especially with a few samples), we use a simple regularized estimate, denoted $\tilde{\mathcal{C}}_{l,t}$, and then aggregate across timesteps using convex weights $w_{l,t} \geq 0$, $\sum_t w_{l,t} = 1$ chosen via the spectral influence score $\mathcal{I}_{l,t}$ (Sec. 2.1): $\bar{\mathcal{C}}_l = \sum_{t=1}^T w_{l,t} \tilde{\mathcal{C}}_{l,t}$, $\bar{\mathcal{R}}_l = \bar{\mathcal{C}}_l^{1/2}$. For cross-attention, the text features are time-invariant, so their statistics are computed once over SlimSet and reused.

Per-module variants. We apply spectral influence score $\mathcal{I}_{l,t}$ consistently across functional groups: (i) **Self-attention (SA):** QK logits use $\bar{\mathcal{R}}^{\text{sa}}$; VO maps use $\bar{\mathcal{R}}^{\text{sa}}$. (ii) **Cross-attention (CA):** Queries use step-mixtures $\bar{\mathcal{R}}^q$, while keys and values use cached text statistics $\bar{\mathcal{R}}^{\text{text}}$. (iii) **Feed-forward (FFN):** Down-projection \mathcal{W}_d is scored with FFN intermediate correlations $\bar{\mathcal{K}}^{\text{ffn}}$. We show the variation of $\mathcal{I}_{l,t}$ across layers, timesteps and functional modules in App. Sec A.5, Fig. 3.

2.4 Module-Aligned Data-Aware Compression (MADAC)

A core challenge in DM compression is that conventional per-matrix factorization treats weights in isolation, overlooking the functional coupling across attention and feed-forward modules. MADAC addresses this by treating each module as an integrated unit and learning a joint, data-aware decomposition that preserves the end-to-end mapping under empirically observed activation distributions.

In attention, the query–key interaction $\mathcal{X}\mathcal{W}_q\mathcal{W}_k^\top \mathcal{X}^\top$ and the value–output map $\mathcal{X}\mathcal{W}_v\mathcal{W}_o$ are *bilinear* in the weights, so compression must respect cross-matrix coupling rather than factorizing each weight in isolation. Likewise, feed-forward blocks compute $\mathcal{Z} = (\mathcal{X}\mathcal{W}_x) \odot \sigma(\mathcal{X}\mathcal{W}_g)$ and $\mathcal{Y} = \mathcal{Z}\mathcal{W}_D$, where the up-projection \mathcal{W}_U is split into a *content* branch \mathcal{W}_x and a *gate* branch \mathcal{W}_g , with down-projection \mathcal{W}_D . The elementwise gating in FFN breaks linearity, making per-matrix factorization inadequate. Denoting module input activations as \mathcal{X} and attention projections as $\{\mathcal{W}_q, \mathcal{W}_k, \mathcal{W}_v, \mathcal{W}_o\}$, we represent each functional module as $f(\mathcal{X}; \mathcal{W}_1, \mathcal{W}_2)$ and compress weight groups by minimizing the data-driven reconstruction loss:

$$\min_{\widehat{\mathcal{W}}_1, \widehat{\mathcal{W}}_2} \sum_{i=1}^N \left\| f(\mathcal{X}_i; \mathcal{W}_1, \mathcal{W}_2) - f(\mathcal{X}_i; \widehat{\mathcal{W}}_1, \widehat{\mathcal{W}}_2) \right\|_F^2 \quad (3)$$

We solve this optimization using the timestep-aware correlation mixtures $\bar{\mathcal{C}}_l$ from Section 2.3, ensuring compression aligns with the activation geometry encountered during inference. For each functional group, we derive specialized decompositions that respect both the computational structure and activation statistics: Nyström approximation for FFN modules and whitened SVD for attention (QK, VO) modules. They are detailed as follows:

Type-I: \mathcal{FFN} via Nyström approximation. The feedforward module consists of gated up-projections $\mathcal{W}_x, \mathcal{W}_g \in \mathbb{R}^{d \times 4d}$ followed by a down-projection $\mathcal{W}_D \in \mathbb{R}^{4d \times d}$. Since \mathcal{W}_g resides within the nonlinearity $\sigma(\cdot)$, we constrain $\mathcal{W}_g, \mathcal{W}_k$'s compressed forms to share a column selection matrix $M_k \in \mathbb{R}^{4d \times k}$ for tractable optimization of Eq. 3: $\mathcal{W}'_x = \mathcal{W}_x M_k, \mathcal{W}'_g = \mathcal{W}_g M_k$. For \mathcal{W}_D , we ensure dimensional compatibility with compressed up-projections by searching over $\mathbb{R}^{k \times d}$. Our theoretical analysis (App. Sec. A.1) reveals that when a single column selection matrix is used, Eq. 3 reduces to a Nyström approximation of the intermediate activation correlation matrix.

Theorem 1 Let $\widehat{\mathcal{W}}_x, \widehat{\mathcal{W}}_g$ be constrained to the form $\mathcal{W}_x M_k, \mathcal{W}_g M_k$ where M_k is a k -column selection matrix, and let $\widehat{\mathcal{W}}_D$ be searched over $\mathbb{R}^{k \times d}$. The optimal $\widehat{\mathcal{W}}_D^*$ is given by:

$$\widehat{\mathcal{W}}_D^* = (M_k^\top \mathcal{K} M_k)^\dagger M_k^\top \mathcal{K} \mathcal{W}_D \quad (4)$$

where $\mathcal{K} = \sum_{i=1}^N \mathcal{Z}_i^\top \mathcal{Z}_i$ is the intermediate activation correlation matrix and $\mathcal{Z}_i = (\mathcal{X}_i \mathcal{W}_x) \odot \sigma(\mathcal{X}_i \mathcal{W}_g)$. The Type-I reconstruction error satisfies:

$$\mathcal{V}_I \leq \|\mathcal{W}_D\|_2^2 \|\mathcal{K}^{-1}\|_2 E_{\text{Nys}}(\mathcal{K}) \quad (5)$$

where $E_{\text{Nys}}(\mathcal{K})$ denotes the Nyström approximation error of \mathcal{K} using the same M_k . Theorem 1 shows that effective Type-I compression can be achieved through a well-designed Nyström approximation of the intermediate correlation matrix \mathcal{K} . We implement this via Algorithm 1, which normalizes \mathcal{K} to correlation form, computes a randomized dominant basis, and selects informative columns using column-pivoted QR (CPQR). The optimal down-projection \mathcal{W}'_D is then solved in closed form on the selected subspace, ensuring both up-projections respect the gated nonlinearity while adapting to the compressed intermediate space.

Type-II: \mathcal{QK} via whitening SVD (WSVD). We now focus on the query-key interactions within multi-head attention mechanisms. The \mathcal{QK} computation corresponds to the bilinear form $f(\mathcal{X}; \mathcal{W}_q, \mathcal{W}_k) = (\mathcal{X} \mathcal{W}_q)(\mathcal{W}_k^\top \mathcal{X}^\top)$, which depends on how query and key directions align with the input distribution. We compress this bilinear operation by factorizing the query-key cross-product $\mathcal{W}_q \mathcal{W}_k^\top$. To ensure the compression respects the anisotropy of the input distribution rather than treating all directions equally, we first whiten both query and key matrices using their respective timestep-aware covariance roots $\bar{\mathcal{R}}_q, \bar{\mathcal{R}}_k$ (Sec. 2.3): $\widetilde{\mathcal{W}}_q = \bar{\mathcal{R}}_q \mathcal{W}_q, \widetilde{\mathcal{W}}_k = \bar{\mathcal{R}}_k \mathcal{W}_k$.

Theorem 2 (*Query-Key compression by whitening SVD*). Let $\widehat{\mathcal{W}}_q, \widehat{\mathcal{W}}_k$ be the rank- r compressed matrices obtained by applying SVD to the whitened cross-product $\widetilde{\mathcal{W}}_q \widetilde{\mathcal{W}}_k^\top = \mathcal{U} \Sigma \mathcal{V}^\top$ and unwhitening: $\widehat{\mathcal{W}}_q = \bar{\mathcal{R}}_q^{-1} \mathcal{U}_r, \widehat{\mathcal{W}}_k = \bar{\mathcal{R}}_k^{-1} \mathcal{V}_r \Sigma_r$. Then the Type-II reconstruction error in Eq. 3 satisfies:

$$\mathcal{V}_{II} \leq \sum_{i=r+1}^{\min(d_q, d_k)} \sigma_i^2(\widetilde{\mathcal{W}}_q \widetilde{\mathcal{W}}_k^\top) \quad (6)$$

where σ_i are the singular values of the whitened cross-product in descending order. This theorem shows that whitening SVD provides the optimal rank- r approximation of the \mathcal{QK} bilinear operator under the covariance-normalized Frobenius norm, and preserves the most important interaction patterns between queries and keys. The procedure is applied independently to each attention head as detailed in Algorithm 2. The full theoretical analysis is provided in App. Sec. A.1.

Type-III: \mathcal{VO} via Whitening SVD (WSVD). Finally, we focus on the Type-III module, which involves the value-output matrices. The module has is expressed as: $f(\mathcal{X}) = \mathcal{X} \mathcal{W}_v \mathcal{W}_o$, so we seek general low-rank matrices for compression: $\widehat{\mathcal{W}}_v \in \mathbb{R}^{d_h \times k}, \widehat{\mathcal{W}}_o \in \mathbb{R}^{k \times d_h}$ such that $\widehat{\mathcal{W}}_v \widehat{\mathcal{W}}_o \approx \mathcal{W}_v \mathcal{W}_o$. The subsequent theorem reveals that the reconstruction can be solved optimally by applying SVD to the whitened composite transformation.

Theorem 3 (*Value-Output compression by whitening SVD*). If we search $\widehat{\mathcal{W}}_v$ and $\widehat{\mathcal{W}}_o$ over $\mathbb{R}^{d_h \times k}$ and $\mathbb{R}^{k \times d_h}$, respectively, the optimum in Eq. 3 is $\widehat{\mathcal{W}}_v = \mathcal{C}^{-1/2} \mathcal{U}_k$ and $\widehat{\mathcal{W}}_o = \Sigma_k \mathcal{V}_k^\top$. Here, $\mathcal{U} \Sigma \mathcal{V}^\top$ and $\mathcal{C} = \sum_{i=1}^N \mathcal{X}_i^\top \mathcal{X}_i$ are the SVD of $\mathcal{C}^{1/2} \mathcal{W}_v \mathcal{W}_o$ and input correlation, respectively. The corresponding Type-III reconstruction error in Eq. 3 is exactly the SVD approximation error relative to $\mathcal{C}^{1/2} \mathcal{W}_v \mathcal{W}_o$:

$$\mathcal{V}_{III} = E_{\text{SVD}}^2(\mathcal{C}^{1/2} \mathcal{W}_v \mathcal{W}_o) \quad (7)$$

In practice (Alg. 3), we use the timestep-aware value correlation $\bar{\mathcal{C}}_v$ (SA: $\bar{\mathcal{C}}^{\text{sa}}$; CA: cached text), compute the SVD of $\bar{\mathcal{C}}_v^{1/2} \mathcal{W}_v \mathcal{W}_o$, keep rank r , and unwhiten; we apply this per head and concatenate. This yields the optimal rank- r approximation under the covariance-weighted Frobenius norm, aligning compression with the anisotropy of value activations (Details in App. Sec. A.1).

Algorithm 1 Type-I \mathcal{FFN} compression via Nyström approximation

Require: $\mathcal{W}_x, \mathcal{W}_g \in \mathbb{R}^{d \times d_{\text{int}}}$, $\mathcal{W}_D \in \mathbb{R}^{d_{\text{int}} \times d}$, intermediate activations $\mathcal{Z}_i = (\mathcal{X}_i \mathcal{W}_x) \odot \sigma(\mathcal{X}_i \mathcal{W}_g)$, correlation $\mathcal{K} = \sum_{i=1}^N \mathcal{Z}_i^\top \mathcal{Z}_i$, target rank $k = \lceil (1 - \text{sparsity}) d_{\text{int}} \rceil$

- 1: $(Q, R, \text{pivot_idx}) \leftarrow \text{CPQR}(\mathcal{K})$ \triangleright Column-pivoted QR; pivot_idx is the column order
- 2: $M_k \leftarrow I_{d_{\text{int}}}[:, \text{pivot_idx}[1:k]]$ \triangleright Select the first k pivot columns
- 3: **return** $(\widehat{\mathcal{W}}_x, \widehat{\mathcal{W}}_g, \widehat{\mathcal{W}}_D) \leftarrow (\mathcal{W}_x M_k, \mathcal{W}_g M_k, (M_k^\top \mathcal{K} M_k)^\dagger M_k^\top \mathcal{K} \mathcal{W}_D)$ \triangleright Nyström-approximated branches and closed-form down-projection

Algorithm 2 Type-II \mathcal{QK} compression via whitening SVD

Require: head-specific QK matrices: $\{W_{q,j} \in \mathbb{R}^{d_q \times d_h}, W_{k,j} \in \mathbb{R}^{d_k \times d_h}\}_{j=1}^H$, correlations $\{C_q, C_k\}$, target rank $r = \lceil (1 - \text{sparsity}) d_{\text{int}} / H \rceil$

- 1: $R_q \leftarrow C_q^{1/2}; R_k \leftarrow C_k^{1/2}$ \triangleright Compute whitening transforms
- 2: **for** $j = 1, \dots, H$ **do**
- 3: $\widetilde{\mathcal{W}}_{q,j} \leftarrow R_q \mathcal{W}_{q,j}; \widetilde{\mathcal{W}}_{k,j} \leftarrow R_k \mathcal{W}_{k,j}, \mathcal{T} \leftarrow \widetilde{\mathcal{W}}_{q,j} \widetilde{\mathcal{W}}_{k,j}^\top$ \triangleright Whiten Q,K; get whitened composite
- 4: $(\mathcal{U}, \Sigma, \mathcal{V}) \leftarrow \text{SVD}(\mathcal{T})$; truncate $\mathcal{U}_r, \Sigma_r, \mathcal{V}_r$ \triangleright SVD and rank truncation
- 5: $\mathcal{W}_{q,j} \leftarrow R_q^{-1} \mathcal{U}_r, \mathcal{W}_{k,j} \leftarrow R_k^{-1} \mathcal{V}_r \Sigma_r$ \triangleright Unwhiten compressed matrices
- 6: **end for**
- 7: **return** $(W_q, W_k) \leftarrow ([W_{q,1}, \dots, W_{q,H}], [W_{k,1}, \dots, W_{k,H}])$ \triangleright Concatenate the heads

2.5 Automatic Rank Allocation Engine

Given a parameter budget B , we choose per-block ranks $\{r_\ell\}$ to maximize fidelity with total parameters $\leq B$. Since estimating block-wise utility is infeasible, we use the *spectral influence* $\mathcal{I}_{\ell,t}$ (Sec. 2.3) as a surrogate and allocate capacity $\propto \sum_t \mathcal{I}_{\ell,t}$, so higher-influence blocks retain more rank. Formal mapping details are in App. Sec. A.2.

Softmax-style Allocation. We convert influence scores into retention fractions via a temperature-controlled softmax (App. A.2), which normalizes importance across layers and concentrates capacity on influential blocks while keeping allocations smooth. Intuitively, each block’s retained rank depends not only on its own importance but also on its relative standing among all blocks, yielding a propagation-aware allocation under the global budget.

Mapping to Ranks. Each block’s retention fraction is multiplied by its effective width (d for \mathcal{QKVO} per head, $4d$ for \mathcal{FFN} intermediates), rounded to hardware-friendly multiples (of 8), and clipped by a minimum rank for stability. The global average sparsity is then adjusted by a simple bisection search to ensure the final parameter count exactly meets the budget B . This convex allocation distributes sparsity in a propagation-aware manner: high-influence blocks retain more rank, while less influential ones are slimmed, all under a unified parameter budget.

This allocation engine is convex, closed-form, and propagation-aware: blocks with high spectral influence automatically keep more capacity, while less critical ones are slimmed more aggressively. All mathematical details are provided in the App. Sec. A.2.

3 Evaluation and Analyses

We evaluate SlimDiff on SDv1.4 and SDv1.5, comparing against both uncompressed models and competitive compression baselines. Our study is structured around key research questions, detailed in the following subsections, while the full experimental setup is deferred to the App. Sec. A.4.

3.1 Does SlimDiff preserve generation quality under compression? To evaluate SlimDiff’s ability to maintain generation quality while achieving significant compression, we conduct comprehensive experiments on the MS-COCO 2014 validation dataset (Lin et al. (2014)). We benchmark against state-of-the-art diffusion compression methods (BK-SDM, Small Stable Diffusion, LD-Pruner) and also report autoregressive baselines (DALL-E, CogView).

Table 1 presents quantitative results on generation quality. SlimDiff achieves competitive performance with only minimal degradation: FID of 13.12 (vs. 13.07 for SD v1.5) and CLIP score of

Algorithm 3 Type-III \mathcal{VO} compression via whitening SVD

Require: head-specific VO matrices: $\{W_{v,j} \in \mathbb{R}^{d_v \times d_h}, W_{o,j} \in \mathbb{R}^{d_h \times d_q}\}_{j=1}^H$, value correlation C_v , target rank $r = \lceil (1 - \text{sparsity}) d_{int}/H \rceil$

- 1: $R_v \leftarrow C_v^{1/2}$ ▷ Compute whitening transform
- 2: **for** $j = 1, \dots, H$ **do**
- 3: $\tilde{\mathcal{W}}_{v,j} \leftarrow R_v \mathcal{W}_{v,j}, \mathcal{T} \leftarrow \tilde{\mathcal{W}}_{v,j} \mathcal{W}_{o,j}$ ▷ Whiten value matrix, get whitened composite
- 4: $(\mathcal{U}, \Sigma, \mathcal{V}) \leftarrow \text{SVD}(\mathcal{T}); \text{truncate } \mathcal{U}_r, \Sigma_r, \mathcal{V}_r$ ▷ SVD and rank truncation
- 5: $\mathcal{W}_{v,j} \leftarrow R_v^{-1} \mathcal{U}_r, \mathcal{W}_{o,j} \leftarrow \Sigma_r \mathcal{V}_r^\top$ ▷ Unwhiten
- 6: **end for**
- 7: **return** $(W_v, W_o) \leftarrow ([W_{v,1}, \dots, W_{v,H}], [W_{o,1}; \dots; W_{o,H}])$ ▷ Concat across heads

Table 1: Comparison on MS-COCO (512×512 , 50 denoising steps, CFG=8). Lower FID and higher IS/CLIP is better. ‘BP-free’ indicates training-free compression, methods using BP are grayed out. LD-Pruner* is not open-sourced; results reported from its paper may not be directly comparable.

Model	BP-free	# Params	FID↓	IS↑	CLIP↑	Data Size (M)	A100 Days
SD v1.5 (Rombach & Esser (2022b))	–	1.04B	13.07	33.49	0.322	> 2000	6250
SD v1.4 (Rombach & Esser (2022a))	–	1.04B	13.05	36.76	0.296	> 2000	6250
Small Stable Diffusion (OFA-Sys (2022))	×	0.76B	12.76	30.27	0.303	229	–
BK-SDM-Base (Kim et al. (2024))	×	0.76B	14.71	31.93	0.314	0.22	13
LD-Pruner* (Zhang et al. (2024b))	×	0.71B	12.37	35.77	0.289	0.22	–
SlimDiff (Ours, v1.5)	✓	0.76B	13.12	32.61	0.319	0.0005	4
SlimDiff (Ours, v1.4)	✓	0.76B	13.21	31.96	0.289	0.0005	4
<i>Autoregressive baselines</i>							
DALL-E (Ramesh et al. (2021))	×	12B	27.5	17.9	–	250	8334
CogView (Ding et al. (2021))	×	4B	27.1	18.2	–	30	–

0.319 (vs. 0.322 for SD v1.5). Notably, SlimDiff is the only method that achieves this performance while being completely backpropagation-free (BP-free), requiring only 500 data points and 4 A100 days compared to 6250 days for the original model. This represents a significant practical advantage over competing methods that require retraining, such as BK-SDM-Base (13 A100 days) and Small Stable Diffusion (extensive retraining on 229M samples). Note that, for LD-Pruner, results are taken from the paper since the code is not open-sourced, and may not be directly comparable. Figure 2 demonstrates SlimDiff’s ability to preserve semantic content and visual quality across diverse prompts. For the “cute Shiba Inu in a cabbage” prompt, SlimDiff maintains the key semantic elements: the dog’s distinctive features and the cabbage setting, while achieving 27% parameter reduction (760M vs. 1.04B). Artistic prompts like “Van Gogh Starry Night” maintain characteristic brushstrokes and color palettes, while samples for prompts “man staring ahead” and “astronaut bears” highlight robustness across photorealistic and creative domains. More results in App. A.7

3.2 How efficient is SlimDiff compared to baselines? SlimDiff not only preserves generational quality but also delivers substantial efficiency gains. We assess efficiency along two axes: *inference cost* (MACs, GPU/CPU latency) and *training cost*. As shown in Table 2, SlimDiff reduces U-Net MACs by 34% per forward pass (112.0G vs. 169.5G) and likewise for full 50-step generation (5.6T vs. 8.5T). This translates into a 45% end-to-end GPU latency reduction (0.87s vs. 1.57s; $\sim 1.8 \times$



Figure 2: Visual comparison with contemporaries shows that SlimDiff maintains higher perceptual quality post-compression. Methods that rely on BP for model slimming are grayed out.

Table 2: **Efficiency comparison on MS-COCO** (512×512 , 50 steps, CFG= 8). MACs are reported per image generation. UNet (1): one U-Net forward pass; Whole (50): $50 \times$ UNet (1). Latency measured with batch size = 1, fp16 for GPU, fp32 for CPU, and an identical scheduler.

Model	Params (B)	MACs		GPU Latency (s)		CPU Latency (s)	
		UNet(1)	Whole(50)	UNet(1)	Whole(50)	UNet(1)	Whole(50)
SD v1.5	1.04	169.5G	8.5T	0.032	1.57	1.90	85.60
BK-SDM-Base	0.76	112.0G	5.6T	0.020	0.92	0.77	34.56
Small Stable Diffusion	0.76	112.0G	5.6T	0.019	0.84	0.85	38.21
SlimDiff (Ours)	0.76	112.0G	5.6T	0.019	0.87	0.92	42.30

Table 3: Evaluation on human preference metrics (higher is better).

Dataset	Model	Params	Score
HPS v2.1	SD v1.5	1.04B	24.45
	SlimDiff (Ours)	0.76B	24.41
ImageReward	SD v1.5	1.04B	0.51
	SlimDiff (Ours)	0.76B	0.56
Pick-a-Pic v1	SD v1.5	1.04B	21.30
	SlimDiff (Ours)	0.76B	21.22

Table 4: Cross-dataset CLIPScore. Rows denote calibration datasets; columns denote evaluation datasets, with diagonals showing in-domain performance.

Calib \rightarrow Eval	COCO	LAION	IRDB	Parti
COCO	0.302	0.309	0.305	0.301
LAION	0.319	0.323	0.308	0.299
IRDB	0.301	0.311	0.314	0.300
Parti	0.300	0.309	0.307	0.303

faster) and a 51% CPU reduction (42.3s vs. 85.6s; $\sim 2.0 \times$ faster). On training cost (Table 1), unlike baselines that require massive retraining, SlimDiff performs compression *without retraining* using only 500 **prompts and 4 A100-days** - over three orders of magnitude lighter than training from scratch, and dramatically smaller than data-hungry baselines.

3.3 How transferable is SlimDiff across datasets? A critical question for practical deployment is whether SlimDiff’s compression strategy generalizes across different datasets and domains. To assess this transferability, we conduct cross-dataset evaluation experiments where we calibrate SlimDiff on one dataset and evaluate its performance on different target datasets, in Table 4. We consider four diverse benchmarks: MS-COCO (Lin et al. (2014)) (natural scene descriptions), LAION (Schuhmann et al. (2022)) (web-scale image–text pairs), IRDB (Xu et al. (2023)) (human preference data), and PartiPrompts (Yu et al. (2023)) (challenging compositional prompts). This setup probes whether SlimDiff’s semantic slimming captures patterns that remain robust when transferred to new distributions. The strong cross-dataset results highlight SlimDiff’s practical value: a model calibrated once on a readily available dataset such as COCO can be deployed across diverse domains without repeating the compression process. This makes the approach especially useful when target-domain data is limited, while also indicating that SlimDiff captures broad semantic structures that transfer reliably across different visual and textual distributions.

3.4 Are Human Preference Metrics Robust to SlimDiff? Beyond standard image quality metrics like FID and CLIP score, we evaluate SlimDiff’s performance on human preference metrics to ensure that compression does not compromise perceptual quality or aesthetic appeal. We assess three established human preference benchmarks, using their own scoring methods: HPS v2.1 (holistic preference scoring) (Wu et al. (2023)), ImageRewardDB (Xu et al. (2023)) (reward-based preference), and Pick-a-Pic v1 (pairwise preference comparisons) (Kirstain et al. (2023)).

Table 3 confirms that SlimDiff preserves alignment with human preferences despite a 27% parameter reduction. Across three distinct benchmarks, SlimDiff delivers similar scores to SD v1.5 on HPS v2.1 and Pic-a-Pic, while achieving a clear improvement on ImageReward. This consistency shows that our compression strategy not only maintains subjective quality but can also strengthen alignment with human judgments, underscoring SlimDiff’s reliability for practical, user-facing deployment.

4 Ablation

In this section, we ask: *How do design choices affect performance?* We ablate SlimDiff’s core components to identify which factors drive quality and efficiency. Specifically, we study (i) SlimSet size, (ii) weighting strategies for timestep correlations, and (iii) the impact of compressing different module types. These analyses clarify why SlimDiff works and where its efficiency gains arise.

Table 5: Ablation on weighting strategies for timestep-aware correlation accumulation. Lower FID is better.

Weighting Strategy	$\text{FID} \downarrow$
Uniform over steps	17.90
Input activation diversity	14.55
Combined (diversity + weights)	13.28
Spectral Influence Score (Ours)	13.12

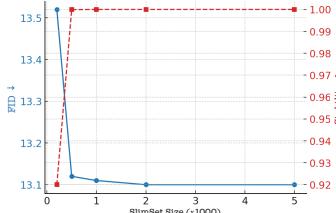


Figure 3: FID and stability as a function of SlimSet size.

Table 6: Module ablation on SD v1.5 using MS-COCO. ✓ = Compressed; ✕ = Uncompressed. Quality degrades most with \mathcal{FFN} compression

SA	CA	\mathcal{FFN}	$\text{FID} \downarrow$	$\text{CLIP} \uparrow$
✓	✓	✗	13.14	0.317
✓	✗	✓	13.18	0.318
✗	✓	✓	13.26	0.317
✓	✓	✓	13.12	0.319

(i) SlimSet Size and Calibration Efficiency. The size of the calibration set determines how well activation statistics are captured. Sets of very few prompts underrepresent semantic diversity and lead to performance drop, while larger sets yield diminishing returns beyond a certain size. Ablating SlimSet sizes from 500 up to 5,000 prompts in Fig. 3 show that around 500 prompts are already sufficient to match the quality of using tens of thousands, offering an effective balance between cost and fidelity. Different 500-prompt SlimSets produce near-identical activation statistics: the subspace overlap between these SlimSets is ≈ 1.0 . We refer to this property ‘Stability’, which ensures the calibration statistics are insensitive to a particular prompt sample - yielding reproducible, deployment-robust models rather than artifacts of one random subset. This effect is consistent across LAION-2B, COCO, PartiPrompts, and ImageRewardDB, confirming that a 500 prompt SlimSet provides a robust and stable calibration set. Accordingly, we adopt 500 as the standard SlimSet size.

(ii) Weighting Strategies for Timestep-Aware Correlation Accumulation. Not all timesteps contribute equally to perceptual fidelity. To evaluate our weighting design for accumulating correlations $\mathcal{C}_{l,t} \rightarrow \mathcal{C}_l$, we test four alternatives: (i) uniform averaging across timesteps, (ii) input-activation diversity only, (iii) spectral influence weighting (ours), and (iv) a combined scheme. As shown in Table 5, uniform weighting performs worst, while spectral influence yields the best FID (13.12). The combined scheme slightly improves over diversity-only but still falls short of spectral influence. These results highlight that fidelity-aware weighting, captured by the spectral influence score, is essential for effective timestep-aware accumulation.

(iii) Module-Specific Compression Contributions. We ablate compression across self-attention (SA), cross-attention (CA), and feedforward (\mathcal{FFN}) modules to assess their relative contributions (Table 6). Note that, if a particular module is uncompressed, the compression budget is distributed over compressed modules. As our Nyström approximation for \mathcal{FFN} s constrains both gate and linear projections to share the same column selection matrix and relies on intermediate activations that are altered by compression, it creates architectural bottlenecks and circular dependencies that make \mathcal{FFN} compression the most sensitive to quality loss. CA shows the next largest impact, while SA remains the most robust. Importantly, compressing all three modules jointly yields the best trade-off between quality and efficiency, highlighting the need for module-aware rather than uniform strategies.

5 Conclusion

We present **SlimDiff**, the first training-free framework that compresses diffusion models by aligning slimming with activation geometry. SlimDiff reduces the model by $\sim 100M$ parameters and achieves up to 35% faster inference versus the original Stable Diffusion Models, all while consistently maintaining generation quality and human preference alignment across diverse benchmarks and datasets. Remarkably, it achieves these gains with only 500 calibration prompts – over 70× fewer than prior work – and without any finetuning, through module-aware decompositions, timestep-weighted correlations, and a compact semantic coresset. Our analyses highlight three principles: effective compression depends more on functional structure than raw capacity, fidelity-aware weighting is critical to prevent error accumulation across timesteps, and module-aware strategies (especially cross-attention and feedforward) drive the best efficiency-quality trade-off. SlimDiff thus provides a principled, training-free compression alternative to retraining-based methods, demonstrating that Diffusion Models can be made both efficient and reliable without a single gradient step.

References

- Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefer, and James Hensman. Slicept: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*, 2024.
- Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4599–4603, 2023.
- Guangliang Chen. Lecture 4: The rayleigh quotient. <https://www.sjsu.edu/faculty/guangliang.chen/Math253S20/lec4RayleighQuotient.pdf>, 2020. San Jose State University, Math 253.
- Jierun Chen, Dongting Hu, Xijie Huang, Huseyin Coskun, Arpit Sahni, Aarush Gupta, Anujraaj Goyal, Dishani Lahiri, Rajesh Singh, Yerlan Idelbayev, et al. Snapgen: Taming high-resolution text-to-image models for mobile devices with efficient architectures and training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7997–8008, 2025.
- Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in neural information processing systems*, 34:19822–19835, 2021.
- Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/35c1d69d23bb5dd6b9abcd68be005d5c-Paper-Conference.pdf.
- Jiarui Gao et al. Block pruning for efficient text-to-image diffusion models. In *ECCV*, 2024.
- Alex Gittens and Michael W Mahoney. Revisiting the nyström method for improved large-scale machine learning. *The Journal of Machine Learning Research*, 17(1):3977–4041, 2016.
- Ming Gu and Stanley C Eisenstat. Efficient algorithms for computing a strong rank-revealing qr factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996.
- Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *ECCV*, 2024.
- Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS) 2023*, 2023. URL <https://arxiv.org/abs/2305.01569>.
- Zico Kolter. Cs229 linear algebra review and reference. Technical report, Stanford University, 2007. URL <https://cs229.stanford.edu/section/cs229-linalg.pdf>. Accessed: 2025-09-19.
- Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17535–17545, 2023.
- Chi-Heng Lin, Shangqian Gao, James Seale Smith, Abhishek Patel, Shikhar Tuli, Yilin Shen, Hongxia Jin, and Yen-Chang Hsu. Modegpt: Modular decomposition for large language model compression. *arXiv preprint arXiv:2408.09632*, 2024.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 740–755. Springer, 2014. URL <https://arxiv.org/abs/1405.0312>.
- Chengqiang Lu, Jianwei Zhang, Yunfei Chu, Zhengyu Chen, Jingren Zhou, Fei Wu, Haiqing Chen, and Hongxia Yang. Knowledge distillation of transformer-based language models revisited. *ArXiv*, abs/2206.14366, 2022.

Binh-Nguyen Nguyen and Yang He. Swift cross-dataset pruning: Enhancing fine-tuning efficiency in natural language understanding. *arXiv preprint arXiv:2501.02432*, 2025.

OFA-Sys. Small stable diffusion. <https://huggingface.co/OFA-Sys/small-stable-diffusion-v0>, 2022.

Farhad Pourkamali-Anaraki and Stephen Becker. Improved fixed-rank nyström approximation via qr decomposition: Practical and theoretical aspects. *Neurocomputing*, 363:261–272, 2019.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Robin Rombach and Patrick Esser. Stable diffusion v1-4. <https://huggingface.co/CompVis/stable-diffusion-v1-4>, 2022a. Model release, CompVis. Accessed: 2025-09-22.

Robin Rombach and Patrick Esser. Stable diffusion v1-5. <https://huggingface.co/runwayml/stable-diffusion-v1-5>, 2022b. Model release, RunwayML. Accessed: 2025-09-22.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CVPR*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Raghunathan, Gaurav Karanam, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. URL <https://arxiv.org/abs/2210.08402>. The LAION-Aesthetics dataset is a filtered subset of LAION-5B using an aesthetic predictor model.

Hui Shen, Jingxuan Zhang, Boning Xiong, Rui Hu, Shoufa Chen, Zhongwei Wan, Xin Wang, Yu Zhang, Zixuan Gong, Guangyin Bao, et al. Efficient diffusion models: A survey. *arXiv preprint arXiv:2502.06805*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hongjie Wang, Difan Liu, Yan Kang, Yijun Li, Zhe Lin, Niraj K Jha, and Yuchen Liu. Attention-driven training-free efficiency enhancement of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16080–16089, 2024.

Yining Wang and Aarti Singh. Provably correct algorithms for matrix column subset selection with selectively sampled data. *Journal of Machine Learning Research*, 18(156):1–42, 2018.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. URL <https://arxiv.org/abs/2306.09341>.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *NeurIPS 2023*, pp. 15903–15935, 2023. URL <https://arxiv.org/abs/2304.05977>.

Yuzhe Yao, Feng Tian, Jun Chen, Haonan Lin, Guang Dai, Yong Liu, and Jingdong Wang. Timestep-aware correction for quantized diffusion models. In *European Conference on Computer Vision*, pp. 215–232. Springer, 2024a.

- Yuzhe Yao et al. Timestep-aware correction for quantized diffusion models. In *ECCV*, 2024b.
- Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Burcu Karagol Ayan, Hans Zhang, et al. Parti: Scaling autoregressive models for content-rich text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL <https://arxiv.org/abs/2206.10789>.
- Qian Zeng, Chenggong Hu, Mingli Song, and Jie Song. Diffusion model quantization: A review. *arXiv preprint arXiv:2505.05215*, 2025.
- Dingkun Zhang, Sijia Li, Chen Chen, Qingsong Xie, and Haonan Lu. Laptop-diff: Layer pruning and normalized distillation for compressing diffusion models. *arXiv preprint arXiv:2404.11098*, 2024a.
- Wei Zhang et al. Ld-pruner: Towards compact text-to-image diffusion models without retraining. *arXiv preprint arXiv:2404.11936*, 2024b.

A Appendix

A.1 Proofs

Preliminaries: \mathcal{X} = input activation to the functional module; $\mathcal{W}_q, \mathcal{W}_k$ = query, key weight matrices; $\mathcal{W}_v, \mathcal{W}_o$ = value, output weight matrices; \mathcal{W}_U = feedforward up-matrix, $\mathcal{W}_x, \mathcal{W}_g$ = part of the up-matrix divided into two, content matrix, gate matrix, \mathcal{W}_D = feedforward down-matrix, M_k = k-column selection matrix

Type-I: FFN via Nyström approximation Proof Sketch

We perform FFN data-aware compression as described in Section 2.4. The module uses gated up-projections within a GeGLU nonlinearity followed by a down-projection. Since traditional low-rank compression of the joint matrices is ineffective, we constrain both up-projections to share a column selection matrix M_k , reducing the objective to a Nyström approximation of the intermediate activation correlation matrix.

What is a Column Selection Matrix? A k -column selection matrix $M_k \in \{0, 1\}^{d \times k}$ has exactly one nonzero per column, indicating the selected indices. For any $A \in \mathbb{R}^{m \times d}$, the product AM_k consists of the k selected columns of A . (Lin et al. (2024); Wang & Singh (2018)). Note that, for any input matrix X and any column selection matrix M , the nonlinearity σ satisfies $\sigma(X)M = \sigma(XM)$. In other words, column selection commutes with σ . This assumption holds for all activation functions that act elementwise (Pourkamali-Anaraki & Becker (2019); Gittens & Mahoney (2016)).

Theorem 1 Proof Details We constrain $\mathcal{W}_x, \mathcal{W}_g \in \mathcal{W}_U$ to be of the form $\mathcal{W}_U M_k$. We define FFN intermediate activation $\mathcal{Z} = f(\mathcal{X}_i[\mathcal{W}_x|\mathcal{W}_g], \mathcal{W}_D) = (\mathcal{X}_i \mathcal{W}_x) \odot \sigma(\mathcal{X}_i \mathcal{W}_g)$, empirical correlation matrix of intermediate FFN features $\mathcal{K} = \mathcal{Z}^\top \mathcal{Z}$.

We can simplify equation 3 as:

$$\begin{aligned}
& \min_{M_k, \widehat{\mathcal{W}}_D} \sum_{i=1}^N \left\| f(\mathcal{X}_i; [\mathcal{W}_x|\mathcal{W}_g], \mathcal{W}_D) - f(\mathcal{X}_i; [\mathcal{W}_x M_k|\mathcal{W}_g M_k], \widehat{\mathcal{W}}_D) \right\|_F^2 \\
&= \min_{M_k, \widehat{\mathcal{W}}_D} \sum_{i=1}^N \left\| ((\mathcal{X}_i \mathcal{W}_x) \odot \sigma(\mathcal{X}_i \mathcal{W}_g)) \mathcal{W}_D - \mathcal{Z}_i M_k \widehat{\mathcal{W}}_D \right\|_F^2 \\
&= \min_{M_k, \widehat{\mathcal{W}}_D} \sum_{i=1}^N \text{Tr} \left((\mathcal{W}_D - M_k \widehat{\mathcal{W}}_D)^\top \mathcal{Z}_i^\top \mathcal{Z}_i (\mathcal{W}_D - M_k \widehat{\mathcal{W}}_D) \right) \\
&= \min_{M_k, \widehat{\mathcal{W}}_D} \left\| \left(\sum_{i=1}^N \mathcal{Z}_i^\top \mathcal{Z}_i \right)^{1/2} (\mathcal{W}_D - M_k \widehat{\mathcal{W}}_D) \right\|_F^2 \\
&= \min_{M_k, \widehat{\mathcal{W}}_D} \left\| \left(\mathcal{K}^{1/2} (\mathcal{W}_D - M_k \widehat{\mathcal{W}}_D) \right) \right\|_F^2,
\end{aligned} \tag{8}$$

Note, we use the following properties from column section matrix for our equation- (i) *Column selection commutes with elementwise nonlinearities*: $\sigma(\mathcal{X}_i \mathcal{W}_g M_k) = \sigma(\mathcal{X}_i \mathcal{W}_g) M_k$. (ii) *Column extraction for linear terms*: $\mathcal{X}_i \mathcal{W}_x M_k = (\mathcal{X}_i \mathcal{W}_x) M_k$. (iii) *Columnwise compatibility of the Hadamard product with a shared selector*: $((A M_k) \odot (B M_k)) = (A \odot B) M_k$.

Optimal Down-Projection. Setting the gradient of Eq. 8 with respect to $\widehat{\mathcal{W}}_D$ to zero yields the normal equations: $M_k^\top \mathcal{K} M_k \widehat{\mathcal{W}}_D = M_k^\top \mathcal{K} \mathcal{W}_D$. The minimum-norm solution is therefore:

$$\widehat{\mathcal{W}}_D^* = (M_k^\top \mathcal{K} M_k)^\dagger M_k^\top \mathcal{K} \mathcal{W}_D. \tag{9}$$

Reduction to Nyström Approximation. Plugging Eq. 9 back into Eq. 8, we obtain

$$\min_{M_k} \left\| \left(\mathcal{K}^{1/2} - \mathcal{K}^{1/2} M_k (M_k^\top \mathcal{K} M_k)^\dagger M_k^\top \mathcal{K} \right) \mathcal{W}_D \right\|_F^2 = \min_{M_k} \left\| \mathcal{W}_D \right\|_2^2 \left\| \mathcal{K}^{-1/2} \right\|_2^2 \left\| \left(\mathcal{K} - \mathcal{K} M_k (M_k^\top \mathcal{K} M_k)^\dagger M_k^\top \mathcal{K} \right) \right\|_F^2 \tag{10}$$

$$\leq \left\| \mathcal{W}_D \right\|_2^2 \left\| \mathcal{K}^{-1} \right\|_2 E_{\text{Nys}}^2(\mathcal{K}) \tag{11}$$

where $E_{\text{Nys}}(\mathcal{K})$ denotes the Nyström approximation error (Gittens & Mahoney (2016) of \mathcal{K} using the same column selection M_k .

While our derivation in Eq. 8 holds for any column selection strategy, in practice we adopt CPQR to construct S_k . Strong rank-revealing QR Gu & Eisenstat (1996) ensures that the selected columns span a well-conditioned rank- k subspace of \mathcal{K} , with Nyström reconstruction error bounded by a modest multiple of the optimal low-rank approximation error Gittens & Mahoney (2016).

Type-II: \mathcal{QK} via Whitening-SVD Proof Sketch

We now turn to the query-key bilinear operator. We compress it via *whitening SVD*, which first rescales queries and keys by their input activation correlation roots, then applies SVD to the whitened cross-product. This yields the optimal rank- r approximation of $\mathcal{W}_q \mathcal{W}_k^\top$ under the correlation-normalized Frobenius norm.

Theorem 2 Proof Details: For input \mathcal{X} , the \mathcal{QK} interaction is $f(\mathcal{X}; \mathcal{W}_q, \mathcal{W}_k) = (\mathcal{X} \mathcal{W}_q)(\mathcal{W}_k^\top \mathcal{X}^\top)$, with rank at most $\min(d_q, d_k)$. We whiten queries and keys using their correlation roots $C_q^{1/2}, C_k^{1/2}$, apply SVD to the whitened cross-product, and truncate to rank r (Eckart–Young–Mirsky). Unwhitening gives the compressed matrices: $\widehat{\mathcal{W}}_q = C_q^{-1/2} \mathcal{U}_r$, $\widehat{\mathcal{W}}_k = C_k^{-1/2} \mathcal{V}_r \Sigma_r$.

We can obtain Eq. 3 as:

$$\begin{aligned}
& \min_{\widehat{\mathcal{W}}_q, \widehat{\mathcal{W}}_k} \sum_{i=1}^N \| f(\mathcal{X}_i; \mathcal{W}_q, \mathcal{W}_k) - f(\mathcal{X}_i; \widehat{\mathcal{W}}_q, \widehat{\mathcal{W}}_k) \|_F^2 \\
&= \min_{\widehat{\mathcal{W}}_q, \widehat{\mathcal{W}}_k} \sum_{i=1}^N \| (\mathcal{X}_i \mathcal{W}_q)(\mathcal{W}_k^\top \mathcal{X}_i^\top) - (\mathcal{X}_i \widehat{\mathcal{W}}_q)(\widehat{\mathcal{W}}_k^\top \mathcal{X}_i^\top) \|_F^2 \\
&= \min_{\widehat{\mathcal{W}}_q, \widehat{\mathcal{W}}_k} \sum_{i=1}^N \| \mathcal{X}_i (\mathcal{W}_q \mathcal{W}_k^\top - \widehat{\mathcal{W}}_q \widehat{\mathcal{W}}_k^\top) \mathcal{X}_i^\top \|_F^2 \\
&= \min_{\widehat{\mathcal{W}}_q, \widehat{\mathcal{W}}_k} \left\| \underbrace{\left(\sum_{i=1}^N \mathcal{X}_i^\top \mathcal{X}_i \right)^{1/2}}_{C_q^{1/2}} (\mathcal{W}_q \mathcal{W}_k^\top - \widehat{\mathcal{W}}_q \widehat{\mathcal{W}}_k^\top) \underbrace{\left(\sum_{i=1}^N \mathcal{X}_i^\top \mathcal{X}_i \right)^{1/2}}_{C_k^{1/2}} \right\|_F^2 \\
&= \min_{\text{rank } \leq r} \| C_q^{1/2} \mathcal{W}_q \mathcal{W}_k^\top C_k^{1/2} - C_q^{1/2} \widehat{\mathcal{W}}_q \widehat{\mathcal{W}}_k^\top C_k^{1/2} \|_F^2.
\end{aligned} \tag{12}$$

Optimal Query, Key matrices. Let $\Delta = \mathcal{W}_q \mathcal{W}_k^\top - \widehat{\mathcal{W}}_q \widehat{\mathcal{W}}_k^\top$. From Eq. 12 we have

$$\mathcal{V}_{\text{II}} = \| C_q^{1/2} \Delta C_k^{1/2} \|_F^2 = \text{Tr}((C_q^{1/2} \Delta C_k^{1/2})^\top (C_q^{1/2} \Delta C_k^{1/2})).$$

Differentiating w.r.t. $\widehat{\mathcal{W}}_q$ and $\widehat{\mathcal{W}}_k$ and setting to zero yields the normal equations:

$$C_q \widehat{\mathcal{W}}_q (\widehat{\mathcal{W}}_k^\top C_k \widehat{\mathcal{W}}_k) = C_q \mathcal{W}_q \mathcal{W}_k^\top C_k \widehat{\mathcal{W}}_k, \quad C_k \widehat{\mathcal{W}}_k (\widehat{\mathcal{W}}_q^\top C_q \widehat{\mathcal{W}}_q) = C_k \mathcal{W}_k \mathcal{W}_q^\top C_q \widehat{\mathcal{W}}_q. \tag{13}$$

Introducing whitened variables: $A = C_q^{1/2} \widehat{\mathcal{W}}_q$, $B = C_k^{1/2} \widehat{\mathcal{W}}_k$, and $\mathcal{M} = C_q^{1/2} \mathcal{W}_q \mathcal{W}_k^\top C_k^{1/2}$, the system becomes: $A(B^\top B) = \mathcal{M}B$, $B(A^\top A) = \mathcal{M}^\top A$.

For the SVD $\mathcal{M} = U \Sigma V^\top$, the minimum-norm solution is $A^* = U_r$, $B^* = V_r \Sigma_r$, yielding the optimal rank- r approximation $U_r \Sigma_r V_r^\top$. Unwhitening gives:

$$\widehat{\mathcal{W}}_q^* = C_q^{-1/2} U_r, \quad \widehat{\mathcal{W}}_k^* = C_k^{-1/2} V_r \Sigma_r. \tag{14}$$

The Type-II reconstruction error is bounded by the spectral tail:

$$\mathcal{V}_{\text{II}} \leq \sum_{i=r+1}^{\min(d_q, d_k)} \sigma_i^2 (C_q^{1/2} \mathcal{W}_q \mathcal{W}_k^\top C_k^{1/2}). \tag{15}$$

Type-III: VO via Whitening-SVD Proof Sketch

We next analyze the value–output module, which, unlike QK, requires whitening only on the value side. The objective reduces to approximating the composite $\mathcal{W}_v \mathcal{W}_o$ under the metric induced by the input correlation matrix $\mathcal{C} = \sum_{i=1}^N \mathcal{X}_i^\top \mathcal{X}_i$.

Theorem 3 Proof Details: For input \mathcal{X} , the \mathcal{VO} interaction is $f(\mathcal{X}; \mathcal{W}_v, \mathcal{W}_o) = \mathcal{X} \mathcal{W}_v \mathcal{W}_o$. By whitening with $\mathcal{C}^{1/2}$, the problem becomes SVD of $\mathcal{C}^{1/2} \mathcal{W}_v \mathcal{W}_o$. Truncating to rank r yields $\widehat{\mathcal{W}}_v = \mathcal{C}^{-1/2} \mathcal{U}_r$, $\widehat{\mathcal{W}}_o = \Sigma_r \mathcal{V}_r^\top$, with reconstruction error $\sum_{i=r+1}^N \sigma_i^2 (\mathcal{C}^{1/2} \mathcal{W}_v \mathcal{W}_o)$.

We solve our Eq 3 for VO as follows:

$$\begin{aligned}
& \min_{\widehat{\mathcal{W}}_v, \widehat{\mathcal{W}}_o} \sum_{i=1}^N \| f(\mathcal{X}_i; \mathcal{W}_v, \mathcal{W}_o) - f(\mathcal{X}_i; \widehat{\mathcal{W}}_v, \widehat{\mathcal{W}}_o) \|_F^2 \\
&= \min_{\widehat{\mathcal{W}}_v, \widehat{\mathcal{W}}_o} \sum_{i=1}^N \| \mathcal{X}_i (\mathcal{W}_v \mathcal{W}_o - \widehat{\mathcal{W}}_v \widehat{\mathcal{W}}_o) \|_F^2 \\
&= \min_{\widehat{\mathcal{W}}_v, \widehat{\mathcal{W}}_o} \sum_{i=1}^N \text{Tr}((\mathcal{W}_v \mathcal{W}_o - \widehat{\mathcal{W}}_v \widehat{\mathcal{W}}_o)^\top \mathcal{X}_i^\top \mathcal{X}_i (\mathcal{W}_v \mathcal{W}_o - \widehat{\mathcal{W}}_v \widehat{\mathcal{W}}_o)) \quad (16) \\
&= \min_{\substack{\widehat{\mathcal{W}}_v, \widehat{\mathcal{W}}_o \\ \text{rank } \leq r}} \left\| \underbrace{\left(\sum_{i=1}^N \mathcal{X}_i^\top \mathcal{X}_i \right)^{1/2} (\mathcal{W}_v \mathcal{W}_o - \widehat{\mathcal{W}}_v \widehat{\mathcal{W}}_o)} \right\|_F^2 \\
&= \min_{\text{rank } \leq r} \| \mathcal{C}^{1/2} \mathcal{W}_v \mathcal{W}_o - \mathcal{C}^{1/2} \widehat{\mathcal{W}}_v \widehat{\mathcal{W}}_o \|_F^2.
\end{aligned}$$

Optimal Value, Output Matrices: By the Eckart–Young–Mirsky theorem, the optimal solution is given by truncating the SVD of $\mathcal{C}^{1/2} \mathcal{W}_v \mathcal{W}_o$ to rank r . This yields

$$\widehat{\mathcal{W}}_v = \mathcal{C}^{-1/2} \mathcal{U}_r, \quad \widehat{\mathcal{W}}_o = \Sigma_r \mathcal{V}_r^\top, \quad (17)$$

with reconstruction error

$$\sum_{i=r+1}^N \sigma_i^2 (\mathcal{C}^{1/2} \mathcal{W}_v \mathcal{W}_o). \quad (18)$$

A.2 Automatic Rank Allocation Engine (Extended)

Under a fixed parameter budget B , we must distribute ranks $\{r_\ell\}_{\ell=1}^L$ across blocks to maximize compression fidelity. This constitutes a constrained optimization problem: maximize utility subject to $\sum_\ell c_\ell(r_\ell) \leq B$, where $c_\ell(\cdot)$ represents the cost model for block ℓ . Directly estimating utility curves for each block is computationally prohibitive, requiring extensive sensitivity analysis across rank choices.

Instead, we leverage our trace-normalized Rayleigh quotient (TRQ) scores $\{s_\ell\}$ as tractable surrogates for block importance. TRQ captures how well each block’s weights align with the dominant directions of their input activations. The trace normalization provides scale invariance across layers, while optional family-specific offsets can be added to s_ℓ (boosting \mathcal{FFN} or cross-attention) to reflect their empirically higher contribution to fidelity within diffusion architectures.

Convex Surrogate Formulation. Let $\rho_\ell \in [0, 1]$ denote the retention fraction (preserved rank relative to effective width) and $\phi_\ell = 1 - \rho_\ell$ the sparsity level. Following the entropy-regularized allocation framework of Lin et al. (2024), we solve:

$$\min_{\{\phi_\ell \in [0, 1]\}} \sum_{\ell=1}^L (s_\ell \phi_\ell + \varepsilon \phi_\ell \log \phi_\ell) \quad \text{s.t.} \quad \frac{1}{L} \sum_{\ell=1}^L \phi_\ell = \bar{\phi} \quad (19)$$

where $\bar{\phi} \in [0, 1]$ is the target average sparsity (determined by budget B) and $\varepsilon > 0$ is a temperature parameter. The linear term $s_\ell \phi_\ell$ penalizes sparsifying high-importance blocks, while the entropy regularizer $\phi_\ell \log \phi_\ell$ prevents winner-take-all collapse by encouraging smooth allocation.

Closed-Form Solution. Problem equation 19 is strictly convex since $\phi \mapsto \phi \log \phi$ has positive second derivative. The Lagrangian optimality conditions yield:

$$s_\ell + \varepsilon(1 + \log \phi_\ell) + \frac{\lambda}{L} = 0$$

Solving and applying the sparsity constraint gives the unique softmax solution:

$$\phi_\ell = L \bar{\phi} \cdot \frac{\exp(-s_\ell/\varepsilon)}{\sum_{j=1}^L \exp(-s_j/\varepsilon)}, \quad \rho_\ell = 1 - \phi_\ell \quad (20)$$

This exponential weighting automatically concentrates capacity on blocks with high TRQ scores while maintaining smooth allocation controlled by temperature ε .

Rank Mapping and Budget Enforcement. Each block has effective width d_ℓ^{eff} (d for $\mathcal{QK}/\mathcal{VO}$ per head, $4d$ for \mathcal{FFN} intermediates). We convert retention fractions to hardware-friendly ranks:

$$r_\ell = \max \left\{ r_{\min}, 8 \cdot \left\lfloor \frac{\rho_\ell d_\ell^{\text{eff}} + 4}{8} \right\rfloor \right\}$$

The rounding ensures tensor core alignment while $r_{\min} = 8$ prevents numerical instability.

Using cost model $c_\ell(r) = a_\ell r + b_\ell$ (where a_ℓ captures GEMM complexity), we find the target sparsity $\bar{\phi}$ via bisection search on the monotonic function $\bar{\phi} \mapsto \sum_\ell c_\ell(r_\ell(\bar{\phi}))$ until the total cost exactly meets budget B .

Properties and Guarantees. Our allocation is *convex* (unique global optimum), *propagation-aware* (high-influence blocks retain more capacity), and *budget-exact* (bisection ensures precise cost targeting). Unlike heuristic approaches, the entropy regularization provides principled smoothness while TRQ scores enable cross-family comparison without manual rescaling.

A.3 Related Work

We enumerate related works in this subsection, describing the popular structural pruning methods, alternative methods like reduction of timestep sampling or dynamic token pruning. We also show works on LLMs which use data-aware compression in their pipeline.

Structural compression of diffusion models. Several methods reduce the parameter footprint of diffusion models through pruning or architectural redesign. Block- and layer-pruning approaches compress the U-Net backbone but typically rely on distillation or finetuning to recover fidelity (Kim et al. (2024); Zhang et al. (2024b,a)). Complementary efforts prune at the timestep or module granularity (Fang et al. (2023); Yao et al. (2024a)), highlighting the role of sequential error propagation. Our work differs by providing a closed-form, training-free pipeline that operates over *functional groups* (\mathcal{QK} , \mathcal{VO} , \mathcal{FFN}) with module-aligned objectives.

Training-free acceleration at inference. A separate thread accelerates sampling without changing model size. Attention-driven step reduction (Wang et al. (2024)) modulates compute over timesteps, while token and cache pruning seek runtime savings (Bolya & Hoffman (2023)). These methods improve wall-clock latency but add per-run heuristics and do not permanently reduce parameters. SlimDiff is orthogonal: it permanently shrinks dimensions/ranks and can be combined with such inference-time techniques.

Quantization for diffusion models. Post-training quantization (PTQ) has been adapted to diffusion pipelines to reduce precision while preserving sample quality (Zeng et al. (2025)). Recent works study timestep-aware calibration, noise-schedule sensitivity, and stability at low bit-widths; surveys synthesize progress and open challenges. Quantization is complementary to SlimDiff: the former reduces *precision*, while SlimDiff reduces *structure*; together they offer stacked efficiency gains.

Activation-/data-aware compression and modular views. Beyond diffusion, activation-aligned and module-aware compression in large transformers (Lin et al. (2024); Ashkboos et al. (2024)) shows that respecting activation geometry and functional coupling outperforms naive matrix-wise

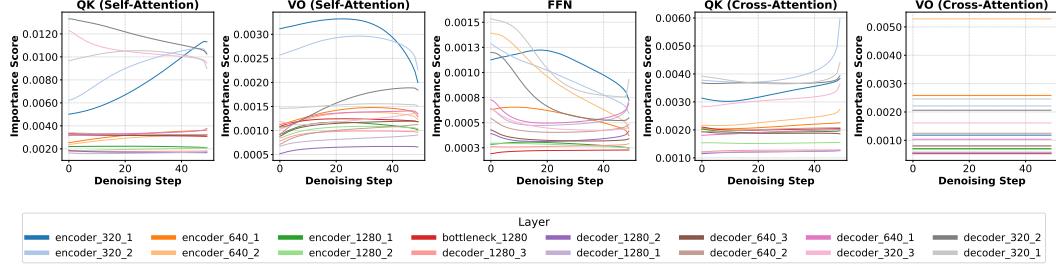


Figure 3: Spectral Influence Score Distribution across different functional modules

pruning. SlimDiff adapts this perspective to diffusion’s evolving activations: it models per-timestep correlations, weights them by spectral influence, and applies whitening–SVD (QK/VO) and Nyström FFN reductions under a global rank allocator.

A.4 Experimental Setup

Models and Code. We evaluate on Stable Diffusion v1.5 and v1.4 using the publicly released U-Net, VAE, and text-encoder weights, and include two public compressed baselines: *BK-SDM-Base* from nota-ai and *Small Stable Diffusion* from OFA-Sys (all checkpoints obtained from their HuggingFace model hubs).

Datasets. Unless noted, we report results on MS-COCO 2014 val at 512×512 . For calibration, we construct a 500-prompt *SlimSet* by sampling text queries from LAION-Aesthetics V2 subset ($\sim 212k$ pairs); only the text side is used. For cross-domain robustness and human preference evaluation, we additionally test with prompts from LAION-Aesthetics, COCO, PartiPrompts, and ImageRewardDB. The same SlimSet is reused across models/datasets with no per-dataset tuning.

Implementation details. All activation collection and compression/optimization procedures run on a single NVIDIA A100 80GB. Unless specified, inference uses 50 denoising steps of the UNet with classifier-free guidance **CFG=8**. We use the default latent resolution ($H = W = 64$) yielding 512×512 images, and keep scheduler and sampling settings fixed across experiments. Code is based on Diffusers and PyTorch, with minor utilities for data-aware factorization.

Metrics. Quality: FID (COCO), Inception Score, CLIPScore; Human preference: HPS v2.1, ImageReward, and Pick-a-Pic scoring on their own datasets. For cross-dataset evaluation, calibration and evaluation prompts come from different corpora as indicated in the main text.

Latency and MACs. MACs are reported per image for one UNet forward and for a full 50-step trajectory. GPU and CPU latencies use identical schedulers with batch size 1 and fp16(GPU)/fp32(CPU). All wall-clock measurements are averaged over multiple runs after a warm-up pass.

A.5 Spectral Influence Score and Diversity Results across layers and modules

We measure the pruning sensitivity of each block using the *trace-normalized Rayleigh quotient (TRQ)* influence score. TRQ evaluates how strongly a block’s weight operator aligns with the dominant eigenspaces of its activation covariance. Intuitively, it quantifies how effectively a block exploits the signal-rich directions of its input. Because TRQ is normalized by the input trace, the score is directly comparable across blocks of different widths and even across module families (\mathcal{QK} , \mathcal{VO} , \mathcal{FFN}).

Figure 3 highlights several consistent trends. Across all module types, the widest layers (1280 channels) tend to have the lowest importance. For \mathcal{QK} blocks, decoder layers dominate encoder layers, reflecting their exposure to greater variance. In contrast, \mathcal{FFN} blocks show higher importance in the encoder, especially at early timesteps. For \mathcal{VO} , cross-attention blocks exhibit stable importance

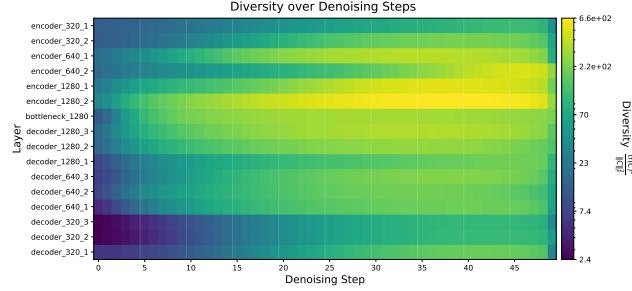


Figure 4: Diversity distribution of input activation across different functional modules

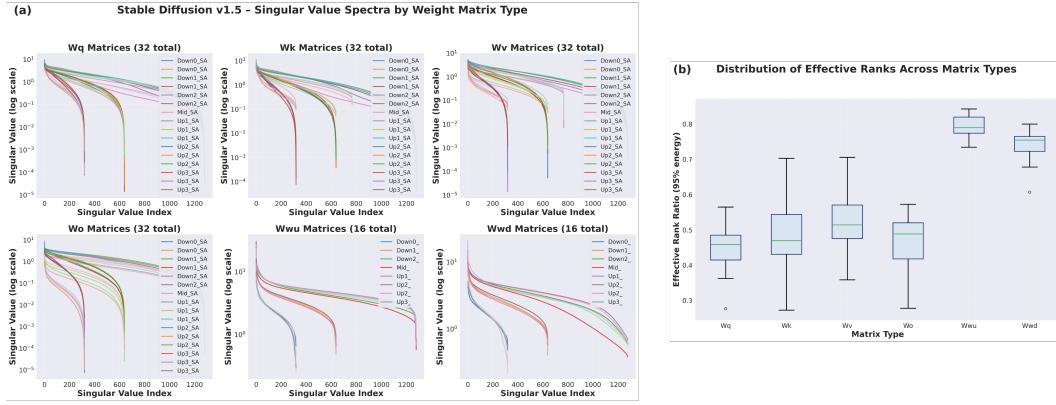


Figure 5: Singular value analysis of Stable Diffusion v1.5 weights. (a) Singular value spectra across attention and feedforward matrices reveal heavy-tailed distributions without clear low-rank cutoffs. (b) Effective rank ratios (95% energy) show that most weights remain in a high-rank regime for FFN layers, while QK/VO module weights average around 50% rank. Importantly, effective ranks must fall below 50% to yield meaningful compression gains.

across timesteps, since they depend only on text-side correlations. These observations guide our pruning strategy, where rank allocation is driven directly by TRQ.

We also examine *diversity*, defined as the spectral spread of input activations. Diversity is highest at early timesteps and decreases as denoising progresses, with mid-U-Net layers achieving higher diversity in fewer steps. This agrees with prior findings that greater diversity reflects less noisy, more semantically aligned activations Wang et al. (2024). However, diversity alone does not reliably predict pruning sensitivity. As confirmed in Table 5, TRQ provides a stronger and more actionable signal for structural pruning, while diversity remains useful primarily as a diagnostic measure.

A.6 Spectral Analysis of SDM Weights

Most attention and feedforward weights in Stable Diffusion fall within a high-rank regime, as evident in Fig. 5. While 40-60% compressibility may appear possible, this is insufficient: reductions below $\sim 50\%$ have little impact on overall parameter count, while more aggressive pruning causes sharp quality degradation due to sequential error propagation across denoising steps (Lin et al. (2024)).

Unlike prior works in diffusion models, that apply SVD independently to each weight matrix, we propose a *joint matrix decomposition* strategy. By respecting the functional couplings within attention (e.g., $W_q W_k^\top$, $W_v W_o$) and integrating data-aware statistics, our method achieves meaningful compression without sacrificing generation quality.

A.7 Additional Visual Results

As shown in Fig. 6 and Fig. 7, SlimDiff consistently preserves generation quality while matching or surpassing structurally compressed baselines (BK-SDM, SSD) across diverse prompts. Importantly,

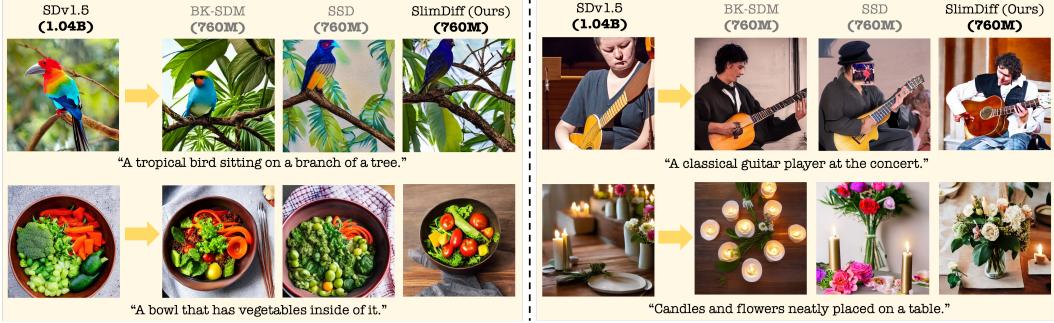


Figure 6: Additional visual comparison with contemporaries on SDv1.5 demonstrates that SlimDiff maintains higher perceptual quality post-compression. Methods that rely on BP for model slimming are grayed out.

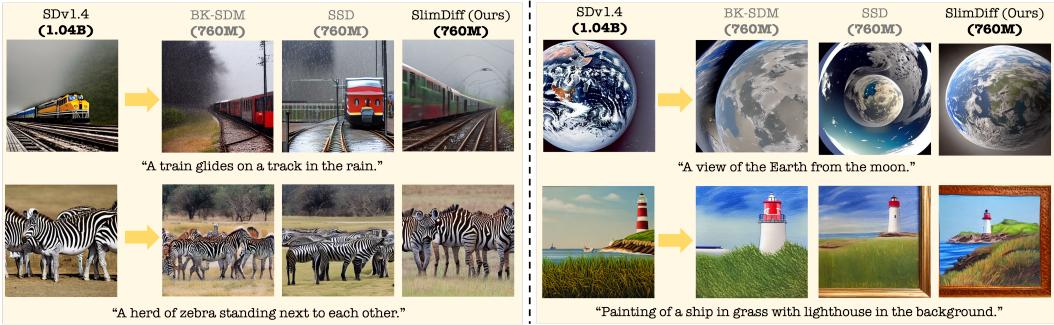


Figure 7: Qualitative comparison across baselines on SDv1.4 highlights SlimDiff’s ability to retain generative performance under compression.

this holds for both SDv1.5 and SDv1.4 backbones, demonstrating that our method is not tied to a particular model variant. SlimDiff thus achieves high fidelity under significant parameter reduction, highlighting its robustness and generality across architectures.