SFA: A STATISTICAL PHYSICS-INSPIRED DIFFUSION TRANSFORMER FOR EFFICIENT AND HIGH-QUALITY IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion transformers have achieved remarkable success in image generation, but accelerating attention while preserving generation quality remains a major challenge at high resolutions. Inspired by statistical physics and the widespread occurrence of scale-free laws in nature, we introduce Scale-Free Attention (SFA), which preserves fine local interactions while summarizing distant regions at progressively coarser scales. For N tokens, mean pooling, token-count correction, and a shared softmax over a scale-free spatial hierarchy give SFA $O(N \log N)$ complexity in both forward and backward passes. Our Triton implementation achieves $357.3\times$ forward and $265.6\times$ backward attention speedups over FlashAttention at 262,144 tokens. After 40 epochs of PixelFlow training from scratch on ImageNet at 256×256 resolution, SFA achieves the best generation quality among five attention mechanisms, including full attention. Fréchet inception distance evaluated on 50,000 generated samples (FID-50k) improves from 13.25 for full attention to 12.51, and the advantage over full attention persists through 80 epochs. SFA also achieves the best generation quality among the compared efficient-attention methods on Flickr-Faces-HQ (FFHQ). Beyond training from scratch, Stable Diffusion 3.5 Medium with attention replaced by SFA retains the generation quality of its full-attention counterpart under matched fine-tuning budgets. We obtain similar results when fine-tuning pretrained PixelFlow on ImageNet. SFA establishes scale-free structure as an effective design principle for efficient, high-quality diffusion transformers.

1 INTRODUCTION

Generating coherent visual content at high resolution requires models that capture both fine detail and long-range structure. Diffusion transformers (DiTs) have achieved strong image-generation performance by combining scalable transformer backbones with iterative denoising (Peebles & Xie, 2023; Esser et al., 2024). However, the quadratic cost of all-to-all attention becomes a major constraint as spatial resolution grows, particularly for direct pixel-space generation with PixelFlow (Chen et al., 2025). Substantially reducing this cost while matching or exceeding the generation quality of full attention remains an open problem.

Existing approaches address different parts of this challenge. FlashAttention improves exact attention through memory-efficient tiling and parallel execution (Dao et al., 2022; Dao, 2024). Its arithmetic cost remains quadratic in token count. Linear attention uses feature-map aggregation to reduce this cost to linear complexity (Katharopoulos et al., 2020; Choromanski et al., 2021). However, the finite feature representations underlying this efficiency trade computational cost against approximation fidelity, making preservation of pretrained softmax interactions a separate challenge. Sparse methods such as Log-linear Sparse Attention (LLSA) (Zhou et al., 2026) reduce interactions through hierarchical selection. Selection adds runtime overhead and omits fine interactions outside the selected regions, which can limit generation quality. Hierarchical attention offers structured global approximations, with H-Transformer-1D (Zhu & Soricut, 2021) developed for one-dimensional sequences. Fast Multipole Attention (FMA) (Kang et al., 2025) extends multilevel computation to image classification tasks using learned pooling and separate softmax at each scale, which complicates efficient kernel implementation and precludes direct resolution extrapolation and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

drop-in fine-tuning from full-attention checkpoints. These limitations restrict its applicability to long sequences and large pretrained generative models. The remaining challenge is to combine subquadratic cost, efficient execution, and generation quality competitive with full attention in both training from scratch and adaptation.

Scale-free laws, widely studied in statistical physics, appear in natural images as approximate scale invariance and power spectra close to an inverse-square law (Field, 1987; Ruderman & Bialek, 1994). Consistent with this spatial prior, we observe an approximate power law in the aggregate radial attention of pretrained PixelFlow, whereas random initialization gives a nearly flat profile (Figure 1(C)). Motivated by this observation, we introduce **Scale-Free Attention (SFA)**, which preserves exact local interactions and partitions the remaining image into progressively larger blocks, each represented by its mean key and value. SFA adds the logarithm of each block's token count to its logit and applies a single softmax to all local tokens and block summaries. Our contributions are threefold.

Scale-free attention. By combining mean pooling, a $\log |B|$ correction for spatial blocks B containing $|B|$ tokens, and a shared softmax over a two-dimensional scale-free hierarchy, SFA retains the original attention projections without additional parameters or hyperparameters. For N tokens, both forward and backward passes have $O(N \log N)$ complexity. In our experiments, SFA achieves higher generation quality than the three efficient-attention baselines and matches or exceeds the performance of full attention (Figure 2(C,D)).

Fast Triton kernels. We implement SFA in Triton (Tillet et al., 2019), using recursive pooling and kernels that fuse attention across spatial scales. Figures 4 and 5 in Appendix B illustrate the forward and backward kernel architectures, respectively. At 262,144 tokens, SFA achieves $357.3\times$ forward and $265.6\times$ backward speedups over FlashAttention on an RTX 5090 (Figure 2(A,B)).

Approximation guarantees. For a fixed spatial partition, we prove that SFA uniquely minimizes reverse Kullback–Leibler (KL) divergence to full attention among blockwise uniform distributions, and bound attention-mass and output errors under logarithmic spatial logits.

2 Related Work

Generative modeling. Denoising diffusion models learn to reverse gradual corruption (Ho et al., 2020), while score-based models describe generation through reverse-time stochastic dynamics (Song et al., 2021). Flow matching learns vector fields along prescribed probability paths (Lipman et al., 2023), and rectified flow uses straight interpolations between noise and data (Liu et al., 2023). These objectives can operate in different representations: latent diffusion reduces spatial computation through learned compression (Rombach et al., 2022), whereas PixelFlow generates directly in pixel space through a resolution cascade (Chen et al., 2025). Diffusion transformers apply scalable transformer backbones to latent patches (Vaswani et al., 2017; Peebles & Xie, 2023), improving generation as model capacity and token count increase, and multimodal rectified flow transformers extend attention to joint image–text interactions (Esser et al., 2024). Across these settings, repeated evaluation during sampling makes the cost of spatial interactions central to generation efficiency.

Efficient diffusion attention. Trainable efficient attention reduces interaction cost while allowing the generator to adapt to the resulting operator. For pretrained generators, compatibility with existing projections and output resolution further supports adaptation through fine-tuning. Recent approaches differ in how they allocate computation: Combiner (Ren et al., 2021) extends sparse attention patterns with indirect attention to regional summaries, providing a global receptive field. FMMformer (Nguyen et al., 2021) computes near- and far-field interactions separately, using an approach similar to linear attention for the latter. Hourglass Diffusion Transformer (HDiT) (Crowson et al., 2024) uses low-resolution global self-attention for coherence. Piecewise Sparse Attention (PISA) (Li et al., 2026) uses dynamic block selection and Taylor approximation of weakly correlated blocks. Video Sparse Attention (VSA) (Zhang et al., 2025) selects important blocks using pooled features, and Sparse-Linear Attention (SLA) (Zhang et al., 2026) combines sparse interactions with a linear component. Log-linear Sparse Attention (LLSA) (Zhou et al., 2026) performs hierarchical selection, enriching fine tokens with pooled keys and values from multiple scales. SFA likewise uses

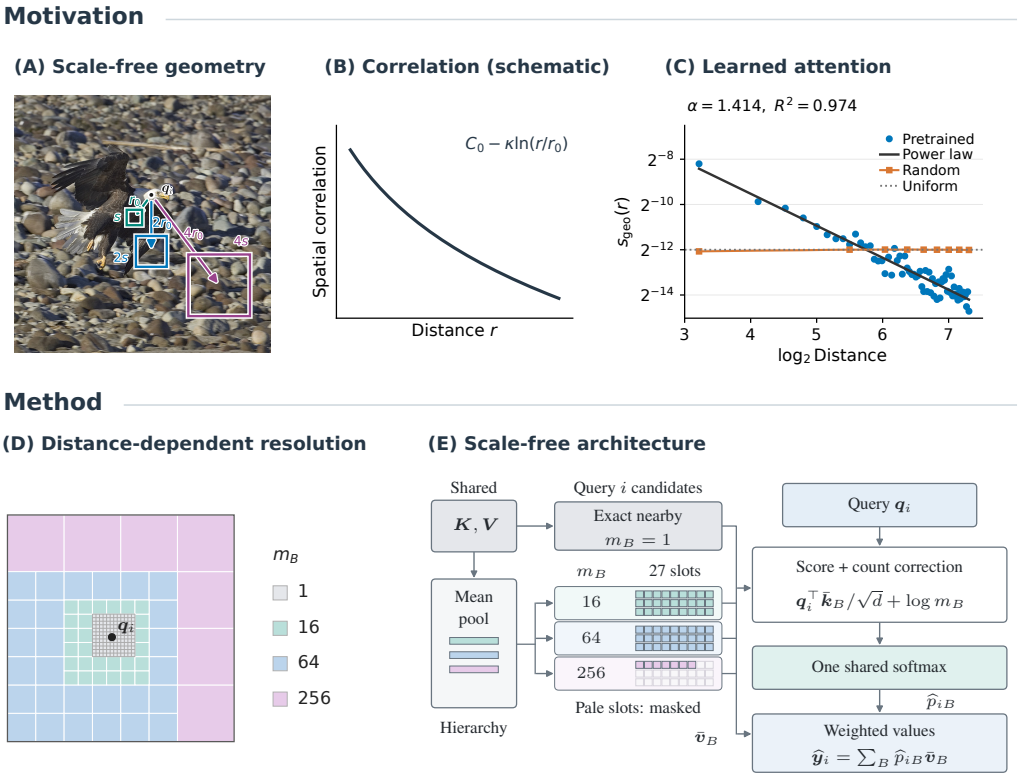


Figure 1: Motivation and construction of Scale-Free Attention (SFA). **(A)** Image regions at center distances $r_0, 2r_0, 4r_0$ from query $\boldsymbol{q}_i$ have side lengths $s, 2s, 4s$. The smallest block lies on the bird, the intermediate block crosses its outline, and the largest contains only background. **(B)** Schematic logarithmic spatial correlation $C_0 - \kappa \ln(r/r_0)$, where r is separation, $r_0 > 0$ is the reference distance, C_0 is an offset, and $\kappa > 0$ controls decay. **(C)** The geometric mean $s_{\text{geo}}(r)$ of layer–head radial profiles follows an approximate power law in pretrained PixelFlow-XL, with decay exponent α and coefficient of determination R^2; random initialization lies near the uniform reference. The horizontal axis shows $\log_2 \text{Distance}$, with distance in pixels (Appendix C.7). **(D)** The partition covers a 64×64 token grid with exact nearby interactions and progressively coarser distant blocks, preserving each original query. **(E)** Key/value matrices $\boldsymbol{K}, \boldsymbol{V}$ supply exact local tokens and a shared mean-pooling hierarchy. Each query uses the same 27-slot layout per pooled scale, with inactive slots masked; active counts are 27, 27, and 7 for (D). Scoring each representative with query $\boldsymbol{q}_i$, adding $\log m_B$, and applying one softmax across all scales yields probabilities $\widehat{p}_{iB}$ for weighted-value aggregation. Here $m_B = |B|$ counts original tokens in block B, and d is the head dimension.

recursive means and shared attention computation. Its fixed spatial interaction lists maintain global coverage through a content-independent allocation of resolution. Appendix D.1 further compares SFA with Sana (Xie et al., 2025) and Native Sparse Attention (NSA) (Yuan et al., 2025).

Multiscale approximation. This fixed spatial construction connects SFA to multiscale attention approximations. FMA (Kang et al., 2025) uses learned pooling kernels and separate softmax at each scale. H-Transformer-1D (Zhu & Soricut, 2021) operates on one-dimensional sequences and also coarse-grains queries. Multi Resolution Analysis (MRA) (Zeng et al., 2022) coarsens queries and uses data-driven criteria to determine where coarsening is appropriate. Focal Attention (Yang et al., 2021) uses multiscale overlapping partitions and introduces learnable pooling, whereas Radial Attention (Li et al., 2025) reduces interactions with spatiotemporally distant video patches. Token Merging (Bolya et al., 2023) combines similar tokens in vision transformers, reducing the sequence length passed to subsequent layers and adding a log token-count bias to the attention logits. Further comparisons appear in Appendix D.

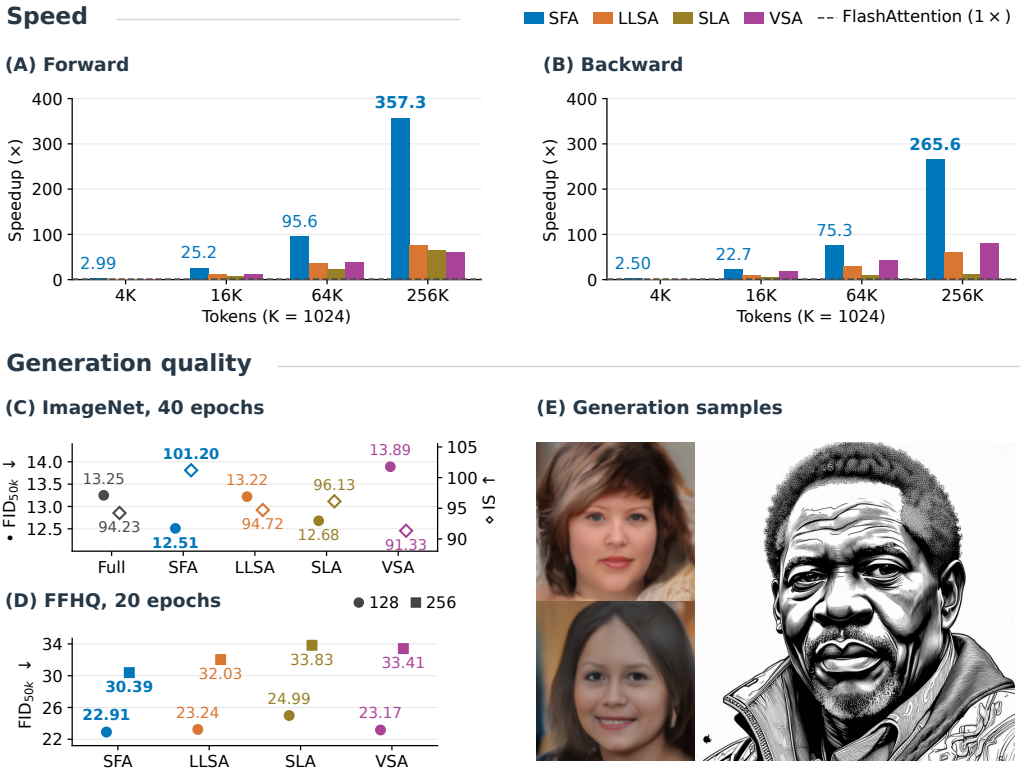


Figure 2: Attention efficiency and generation quality of SFA. **(A,B)** Forward and backward attention speedups over FlashAttention for 4,096–262,144 tokens on an RTX 5090. **(C)** PixelFlow trained from scratch on ImageNet (Deng et al., 2009) at 256×256 resolution for 40 epochs. Circles show Fréchet inception distance (FID; Heusel et al., 2017) on the left axis (lower is better), and diamonds show inception score (IS; Salimans et al., 2016) on the right axis (higher is better), both evaluated on 50,000 generated samples. **(D)** FFHQ (Karras et al., 2019) FID-50k after 20 epochs at each resolution: circles and squares denote 128×128 and 256×256, respectively. Protocols and full results are in Appendices C.2–C.4. **(E)** Generated face samples at 128×128 (left) and a text-to-image sample from Stable Diffusion 3.5 Medium (Stability AI, 2024) at 1024×1024 (right). The text prompt is reproduced in Appendix C.9.

3 Scale-Free Attention

Deterministic scalars are italic, random scalars upright, vectors bold, and matrices bold capitals; plain indexed letters denote coordinates. Logarithms are natural unless specified. Consider one head on an $L \times L$ grid, with power-of-two side length $L \geq 8$ and $N = L^2$ tokens. Indices $i, j \in \{1, \ldots, N\}$ identify positions $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^2$, queries and keys $\boldsymbol{q}_i, \boldsymbol{k}_j \in \mathbb{R}^d$ after positional encoding, and values $\boldsymbol{v}_j \in \mathbb{R}^{d_v}$. Thus, d and d_v are the query/key and value dimensions. Full attention computes logits z_{ij}, row normalizers Z_i, weights a_{ij}, and outputs $\boldsymbol{y}_i \in \mathbb{R}^{d_v}$:

$$z_{ij} = \frac{\boldsymbol{q}_i^\top \boldsymbol{k}_j}{\sqrt{d}}, \qquad Z_i = \sum_{j=1}^{N} e^{z_{ij}}, \qquad a_{ij} = \frac{e^{z_{ij}}}{Z_i}, \qquad \boldsymbol{y}_i = \sum_{j=1}^{N} a_{ij} \boldsymbol{v}_j. \tag{1}$$

Here $\top$ denotes transpose, $\| \cdot \|_2$ is the Euclidean norm, and $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iN})^\top$ and $\boldsymbol{a}_i = (a_{i1}, \ldots, a_{iN})^\top$ collect each query's logits and weights.

3.1 Spatial Scale as a Guide to Attention

To choose the spatial resolution of attention, we use the scale structure of natural images illustrated in Figure 1(A,B). An isotropic finite-band approximation to their inverse-square power spectra

(Ruderman & Bialek, 1994) gives logarithmic spatial correlation over intermediate distances in two dimensions (Appendix A.1). Define the distance between distinct tokens as $r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 > 0$ and fix a reference distance $r_* > 0$. If query–key interactions inherit this dependence, their logits admit a radial model with query-dependent offset b_i, decay exponent $\alpha_i \geq 0$, and content residual u_{ij}:

$$z_{ij} = b_i - \alpha_i \log(r_{ij}/r_*) + u_{ij}, \qquad \alpha_i \geq 0. \tag{2}$$

The radial weights $\exp(z_{ij} - u_{ij}) \propto r_{ij}^{-\alpha_i}$ vary by the same factor over any fixed distance ratio. Section 5.2 and Appendix C.7 present the corresponding empirical measurements. This power law motivates the growth of block sizes in proportion to distance. The pretrained attention profile in Figure 1(C) supports this spatial allocation, within which content-dependent dot products allow the model to learn interaction strengths.

3.2 A COMPLETE MULTISCALE PARTITION

SFA realizes this allocation through a scale-free two-dimensional hierarchical partition similar to that in FMA (Kang et al., 2025) and H-Transformer (Zhu & Soricut, 2021). Each query belongs to a 4×4 base block whose 3×3 block neighborhood, clipped to the image boundary, supplies at most 144 exact singleton tokens. Distant regions are represented through a quadtree. At each pooled scale, we take the children of the 3×3 neighborhood around the query ancestor's parent and exclude the ancestor's own 3×3 neighborhood. This leaves at most 27 blocks per scale, with side lengths doubling at successive scales.

This construction covers the entire image. Together with the exact local tokens, the pooled blocks form a complete, disjoint partition $\mathcal{P}_i$ for query i: every original token belongs to exactly one represented block. The 16 queries in each base block share this partition while retaining their original resolution and computing their own attention weights. Figure 1(D) illustrates this coverage; Appendix B.1 specifies the interaction lists and boundary handling.

3.3 COUNT-CORRECTED AGGREGATION ACROSS SCALES

Each block $B \in \mathcal{P}_i$ of $m_B = |B|$ tokens is represented by its mean key $\bar{\boldsymbol{k}}_B$ and mean value $\bar{\boldsymbol{v}}_B$:

$$\bar{\boldsymbol{k}}_B = \frac{1}{m_B} \sum_{j \in B} \boldsymbol{k}_j, \qquad \bar{\boldsymbol{v}}_B = \frac{1}{m_B} \sum_{j \in B} \boldsymbol{v}_j. \tag{3}$$

For a singleton, these means recover the original key and value, so local interactions and distant summaries use the same rule. Pooling follows positional encoding, including rotary position embeddings (RoPE; Su et al., 2024), preserving the mean logit through linearity: $\bar{z}_{iB} = \boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_B / \sqrt{d} = m_B^{-1} \sum_{j \in B} z_{ij}$. SFA assigns block probability mass $\widehat{p}_{iB}$ and computes output $\widehat{\boldsymbol{y}}_i$ through

$$\widehat{p}_{iB} = \frac{\exp(\bar{z}_{iB} + \log m_B)}{\sum_{C \in \mathcal{P}_i} \exp(\bar{z}_{iC} + \log m_C)}, \qquad \widehat{\boldsymbol{y}}_i = \sum_{B \in \mathcal{P}_i} \widehat{p}_{iB} \bar{\boldsymbol{v}}_B. \tag{4}$$

The $\log m_B$ correction gives each representative the population of its original block: exponentiation multiplies its contribution by m_B, while the pooled dot product determines content-dependent relevance. Related corrections appear in token merging (Bolya et al., 2023). A single softmax jointly normalizes representatives from all scales. Together, mean pooling, count correction, shared normalization, and complete, disjoint coverage define an approximation of full attention on the original token space (Section 4).

3.4 LINEAR PREPROCESSING AND LOG-LINEAR ATTENTION

After pooling base blocks, each coarser mean averages four child means, allowing summaries to be constructed once and shared across queries. The geometric decrease in summary counts gives $O(N(d + d_v))$ preprocessing, while constant representatives per scale give $O(N(d + d_v) \log N)$ forward and backward cost.

Our Triton kernels reuse shared data and fuse attention across scales, with forward and backward architectures detailed in Appendix B (Figures 4 and 5). SFA retains the original projections, head dimensions, and output shape, enabling direct reuse of pretrained weights.

4 APPROXIMATION AND SCALE STRUCTURE

We characterize SFA's approximation to full attention and relate its error to the relative spatial resolution of the partition. For probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ with coordinates p_j and q_j, write their Kullback–Leibler (KL) divergence as $D_{\mathrm{KL}}(\boldsymbol{p}\|\boldsymbol{q}) = \sum_j p_j \ln(p_j/q_j)$.

Optimality for a fixed partition. For a query i, expand each SFA block probability over its constituent tokens: $\widehat{a}_{ij} = \widehat{p}_{iB}/m_B$ for $j \in B$. Let $\widehat{\boldsymbol{a}}_i = (\widehat{a}_{i1}, \ldots, \widehat{a}_{iN})^\top$ and let $\mathcal{U}_i$ be the family of probability distributions uniform within each block of $\mathcal{P}_i$.

Theorem 4.1. *For any finite logits and fixed partition $\mathcal{P}_i$, SFA uniquely minimizes reverse KL within $\mathcal{U}_i$:*

$$\widehat{\boldsymbol{a}}_i = \operatorname*{arg\,min}_{\boldsymbol{b}\in\mathcal{U}_i} D_{\mathrm{KL}}(\boldsymbol{b}\|\boldsymbol{a}_i), \qquad \widehat{\boldsymbol{y}}_i = \sum_j \widehat{a}_{ij}\boldsymbol{v}_j. \tag{5}$$

Mean pooling, token-count correction, and shared normalization jointly realize this optimum. If logits are constant within each block, SFA recovers full attention exactly for arbitrary values. More generally, it approximates the distribution induced by the original queries and keys, providing a basis for reusing pretrained weights. Partition refinement enlarges $\mathcal{U}_i$ and therefore cannot increase the minimum divergence, providing control over the tradeoff between accuracy and computation. Appendix A.2 proves these projection and refinement properties.

Approximation quality depends on the key variation seen by each query. For $j \in B$, define the projected residual $\epsilon_{ij} = \boldsymbol{q}_i^\top(\boldsymbol{k}_j - \bar{\boldsymbol{k}}_B)/\sqrt{d}$. Variations orthogonal to the query leave its logits unchanged, so substantial within-block differences in key features can coexist with accurate attention. For nonzero residuals, output error also depends on their association with the values. Appendix A.6 bounds these contributions, connecting the approximation to learned content-dependent interactions beyond the radial model.

Accuracy across spatial scales. For a pooled block B, let ℓ_B be its diameter and $r_{iB}^- = \min_{j\in B} r_{ij} > 0$ its minimum distance from the query. Their ratio measures the relative spatial resolution. Write $p_{iB} = \sum_{j\in B} a_{ij}$ for its full-attention probability mass. Define the global and within-block value ranges as $R_V = \max_{j,t} \|\boldsymbol{v}_j - \boldsymbol{v}_t\|_\infty$ and $R_{V,B} = \max_{j,t\in B} \|\boldsymbol{v}_j - \boldsymbol{v}_t\|_\infty$, where $\|\cdot\|_\infty$ denotes the maximum absolute coordinate.

Theorem 4.2. *Suppose every pooled block follows Eq. (2) with $u_{ij} = 0$ and satisfies $\ell_B \le \eta r_{iB}^-$ for $\eta \ge 0$. Let $\delta_i = \alpha_i \ln(1+\eta)$. Then*

$$\begin{aligned} &D_{\mathrm{KL}}(\widehat{\boldsymbol{a}}_i\|\boldsymbol{a}_i) \le \frac{\delta_i^2}{8}, \\ &\|\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i\|_\infty \le R_V \tanh\left(\frac{\delta_i^2}{32}\right) + \tanh\left(\frac{\delta_i}{4}\right)\sum_B p_{iB} R_{V,B}. \end{aligned} \tag{6}$$

The two output terms separate changes in probability mass across blocks from changes in value weighting within blocks. Their dependence on relative block size allows larger distant blocks to retain the same accuracy control, independently of absolute distance and token count when α_i and R_V are uniformly bounded. If $R_{V,B} = O(\eta)$ uniformly across blocks, both terms are $O(\eta^2)$. Appendix A.5 proves these bounds and extends them to logits with within-block content variation.

For $r > 0$ in the continuous two-dimensional radial model, a complete annulus between r and $2r$ has token population proportional to r^2 under uniform density. Its unnormalized attention mass therefore scales as $r^{2-\alpha_i}$, so distant tokens can collectively remain important despite weak individual interactions. The $\log m_B$ correction retains this population factor when the region is pooled (Appendix A.7).

A preference for coherent far-field signals. The same aggregation also shapes the information available during training. At fixed block masses and with weights independent of the value noise, uniform aggregation minimizes output noise variance under independent, equal-variance noise. This favors coherent far-field signals while exact local interactions retain fine detail (Appendix A.8).

Pooling after RoPE further acts as a low-pass filter on position-modulated keys, with bandwidth decreasing as block size grows (Appendix A.9). These properties give SFA a spatial inductive bias for image generation alongside its controlled approximation of pretrained attention.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Our baselines are LLSA (Zhou et al., 2026), SLA (Zhang et al., 2026), and VSA (Zhang et al., 2025), using their official Triton implementations and matching the selected-block budgets of SLA and VSA to LLSA. Full attention provides an additional reference for ImageNet generation and efficiency measurements. Figure 2(E) shows representative generated images.

We evaluate PixelFlow (Chen et al., 2025) on ImageNet at 256×256 resolution (Deng et al., 2009), replacing attention in its highest-resolution stage of 4096 tokens and retaining full attention in the other three stages. For longer spatial sequences, we use the small DiT variant (DiT-S; Peebles & Xie, 2023) in pixel space on FFHQ (Karras et al., 2019) at resolutions 128, 256, and 512, following the resolution cascade of LLSA in which each resolution continues from the preceding one (Table 3). FID (Heusel et al., 2017) and inception score (IS; Salimans et al., 2016) use 50,000 generated samples, and all throughput measurements use one NVIDIA RTX 5090 GPU. Appendix C gives the detailed configurations, evaluation protocols, and fixed-rate baseline comparisons.

5.2 SPATIAL STATISTICS OF PRETRAINED ATTENTION

In pretrained PixelFlow-XL, geometric-mean radial attention $s_{\mathrm{geo}}(r)$ at distance r follows $s_{\mathrm{geo}}(r) \propto r^{-\alpha}$ with $\alpha = 1.414$ over 8–160 pixels (Figure 1(C)). The power-law fit achieves $R^2 = 0.974$, compared with 0.589 for logarithmic decay, both evaluated on the original attention scale. By contrast, random initialization produces a nearly uniform radial profile, indicating that the spatial scaling emerges through training and supporting SFA's prior. Appendix C.7 gives the measurement procedure and aggregate and individual fit comparisons.

5.3 ATTENTION AND END-TO-END EFFICIENCY

Compared with FlashAttention, SFA's forward and backward speedups increase from $2.99\times$ and $2.50\times$ at 4096 tokens to $357.3\times$ and $265.6\times$ at 262,144 tokens (Figure 2(A,B)). At the largest scale, SFA's forward and backward speedups over LLSA are $4.70\times$ and $4.45\times$, respectively.

End-to-end gains are bounded by projections, feed-forward networks, and other computations outside attention. The kernel gains above yield $25.75\times$ faster training and $22.54\times$ faster generation on FFHQ at 512×512 resolution, relative to full attention. In Stable Diffusion 3.5 Medium at 2048×2048, replacing both image self-attention and image-to-image interactions in joint attention (preserving all other interactions) gives $2.10\times$ training and $1.97\times$ generation speedups over full attention.

It should be emphasized that on FFHQ, SFA achieves nearly constant token throughput at a fixed training token batch, demonstrating that attention is no longer the computational bottleneck even at 262,144 tokens per sequence. Tables 4, 5, 7, and 12 provide the full kernel, FFHQ, PixelFlow, and Stable Diffusion 3.5 Medium measurements, respectively. Across these model-level comparisons, SFA's memory consumption remains close to the FlashAttention baseline.

5.4 GENERATION FROM SCRATCH

To assess generation quality alongside these efficiency gains, we train all five methods on ImageNet for 40 epochs. Figure 2(C) summarizes their final FID and IS, while Table 1 also reports the 10-epoch results and model throughput. We additionally continue full attention and SFA to 80 epochs to examine whether the quality advantage persists with longer training.

At 40 epochs, SFA achieves the lowest FID of 12.51 and the highest IS of 101.20 among the five attention mechanisms. Its advantage over full attention persists through 80 epochs (Figure 3). SFA achieves these quality gains while improving both training and generation throughput. SLA also

Table 1: PixelFlow on ImageNet-256, with runtime measured for the entire model using image batch sizes 4 for training and 16 for generation. Both FID and IS use 50,000 generated samples. Quality is reported at epochs 10 and 40. Bold marks the best value in each column.

	Train	Generate	$\text{FID}_{50k} \downarrow$		IS ↑	
Method	(ms/step)	(images/s)	10 epochs	40 epochs	10 epochs	40 epochs
Full	308.6	0.178	41.88	13.25	30.58	94.23
LLSA	300.5	0.179	41.66	13.22	31.81	94.72
SLA	316.7	0.171	41.10	12.68	31.38	96.13
VSA	313.0	0.175	42.62	13.89	30.40	91.33
SFA	**286.8**	**0.194**	**40.66**	**12.51**	**32.90**	**101.20**

lowers FID relative to full attention, but with lower throughput in both training and generation. Appendix C.4 (Tables 7–10) provides the complete runtime and training results.

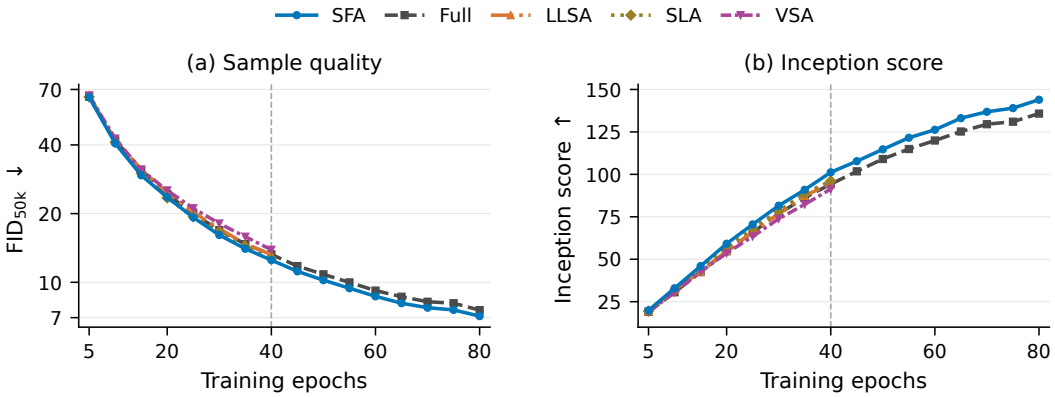


Figure 3: Training PixelFlow from scratch on ImageNet-256. SFA achieves the best FID and IS at the 40-epoch endpoint, marked by the vertical line, and improves both metrics over full attention at every recorded checkpoint through epoch 80. Each point uses 50,000 generated samples; LLSA, SLA, and VSA end at epoch 40.

On FFHQ, SFA achieves the best generation quality among the four efficient-attention variants at resolutions 128 and 256 using 1×1 pixel patches. After the 20-epoch stage at each resolution, its FID_{50k} reaches 22.91 and 30.39, respectively (Figure 2(D); Table 6). At resolution 512, we compare SFA and LLSA after two epochs of continuation from their respective resolution-256 checkpoints: SFA achieves FID_{50k} of 35.59, compared with 38.90 for LLSA. Appendix C.3 gives the detailed results.

5.5 Adapting pretrained generators

We fine-tune the original PixelFlow model and a model with SFA in its highest-resolution stage from the same checkpoint under an identical schedule. After 3.5 epochs, full attention reaches FID_{50k} of 2.28 and IS of 273.25, compared with 2.29 and 284.71 for SFA, using the same 50,000-sample evaluation protocol (Table 11). SFA thus retains comparable FID and improves IS under the matched budget, preserving generation quality with the original projections and parameter count.

For Stable Diffusion 3.5 Medium (Stability AI, 2024; Esser et al., 2024), we retain the original effective patch size of 16×16 pixels and replace image self-attention alone (SA), which the first 13 of its 24 blocks carry, or together with image-to-image interactions in the joint attention of all 24 blocks (ALL). After one epoch of matched fine-tuning, both scopes recover the quality of the equally fine-tuned original model. ALL achieves a Dense Prompt Graph Benchmark (DPG-Bench; Hu et al., 2024) score of 85.51, matching full attention, and a human preference score (HPS) v2.1 (Wu et al., 2023) of 30.262, compared with 30.261 for full attention.

The joint-attention replacement is specified in Appendix C.6, with the complete direct-replacement and fine-tuning comparisons in Tables 13–15.

5.6 Ablations

We evaluate four variants to examine count correction, normalization across scales, far-field context, and aggregation. Under the FFHQ-128 setup above, removing the $\log m_B$ correction raises FID_{50k} from 22.91 to 25.91, and normalizing each scale separately, as FMA does, raises it to 27.99. Retaining only the exact 12×12 local neighborhood raises it to 45.76, supporting the contribution of far-field summaries.

The first two changes also reduce quality under direct image self-attention replacement in Stable Diffusion 3.5 Medium without fine-tuning. In particular, separate softmax lowers overall HPS v2.1 from 29.596 for SFA to 11.176 and produces grid-like artifacts. On ImageNet-256, replacing mean pooling with learnable aggregation and removing count correction raises FID_{50k} from 40.66 to 41.40 and lowers IS from 32.90 to 31.50 after ten epochs. In our RTX 5090 implementation, this variant also takes 1.96–$2.64\times$ SFA's forward attention time and 2.89–$4.74\times$ its backward attention time across the tested sequence lengths of 4096–262,144 tokens. These results support the combination of mean pooling, count correction, and shared normalization. Appendix C.8 gives the four variant definitions, detailed quality results, and efficiency analysis.

6 Discussion and Limitations

The spatial hierarchy in SFA suggests a natural extension to video generation through spatiotemporal partitioning. Such a hierarchy could preserve fine interactions within nearby frames while summarizing increasingly distant regions across both space and time. Mean pooling, token-count correction, and shared normalization apply to these spatiotemporal blocks, with motion and the different rates of spatial and temporal variation guiding their shapes and resolutions. Adapting temporal coarsening to these correlations throughout denoising would be a promising direction for balancing generation quality and computational cost.

Beyond video generation, this principle could support action-conditioned visual world models, which predict how an environment evolves in response to actions (Bruce et al., 2024). A hierarchy over past observations could retain detailed interactions with recent frames and conditioning actions while representing older visual context through coarser summaries, potentially making longer histories more affordable during prediction. In a causal formulation, each pooled summary and its token count would need to include only the history visible to its query. The resulting partition would thus need to respect both spatiotemporal structure and causal ordering.

Our study focuses on regular two-dimensional image grids, leaving irregular grids unexplored. Additionally, further acceleration is constrained by the remaining transformer components: our FFHQ experiments show that the computational bottleneck has shifted from attention to projections, feed-forward networks, and other operations. Optimizing the transformer block as a whole therefore remains a direction for future work.

7 Conclusion

By combining a scale-free spatial partition with mean aggregation, token-count correction, and a single softmax across scales, SFA gives an $O(N \log N)$ attention mechanism with a precise approximation relationship to full attention and a spatial inductive bias suited to image generation. Our analysis links approximation quality to within-block logit variation and explains why relative spatial resolution is appropriate for power-law interactions. Empirically, SFA achieves the best quality in the five-method PixelFlow comparison at 40 epochs, improves FFHQ generation, and preserves full-attention quality under matched fine-tuning budgets in PixelFlow and Stable Diffusion 3.5 Medium. Together with substantial attention and whole-model speedups, these results support its use in both training new generators and adapting pretrained ones.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and identifying relevant literature. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. We have reviewed all AI-assisted work. We have reviewed the relevant literature found by the large language model. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work introduces efficient attention mechanisms for diffusion transformers. Our experiments use existing public image datasets, including FFHQ, which contains photographs of real people. We acknowledge the privacy concerns and demographic biases associated with such datasets, and the importance of respecting their licenses and usage restrictions. No real images from the FFHQ dataset are displayed in this manuscript. All datasets and models utilized in this study are strictly used for non-commercial, academic research purposes in full compliance with their respective licenses. Additionally, reducing computational costs may improve access to image-generation models and reduce resource use, but may also lower the cost of producing deceptive or harmful synthetic content. We therefore encourage responsible use and consideration of privacy, bias, and misuse risks in downstream applications.

REPRODUCIBILITY STATEMENT

Section 3 specifies SFA's multiscale partition, mean pooling, and count-corrected shared softmax. Appendix A states the assumptions and provides proofs of the theoretical results, while Appendix B details the forward and backward Triton computations. To support reproduction of the experiments, Appendix C documents the hardware and software environments, baseline configurations, training and fine-tuning schedules, and sampling and evaluation protocols, with training hyperparameters in Table 2. We will open-source our code, including the Triton implementation of SFA, in the Supplementary Material.

REFERENCES

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Velickovic. Round and round we go! What makes rotary positional encodings useful? In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=GtvuNrk58a.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal M. P. Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott E. Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. PixelFlow: Pixel-space generative models with flow. *CoRR*, abs/2504.07963, 2025. doi: 10.48550/ARXIV.2504.07963. URL https://doi.org/10.48550/arXiv.2504.07963.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z. Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 9550–9575. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/crowson24a.html.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=mZn2Xyh9Ec.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part X*, volume 15068 of *Lecture Notes in Computer*

Science, pp. 289–305. Springer, 2024. doi: 10.1007/978-3-031-72684-2_17. URL https://doi.org/10.1007/978-3-031-72684-2_17.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6626–6637, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 01621459, 1537274X. URL http://www.jstor.org/stable/2282952.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment. *CoRR*, abs/2403.05135, 2024. doi: 10.48550/ARXIV.2403.05135. URL https://doi.org/10.48550/arXiv.2403.05135.

Zahra Kadkhodaie, Florentin Guth, Eero P. Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ANvmVS2Yr0.

Yanming Kang, Giang Tran, and Hans De Sterck. Fast multipole attention: A scalable multilevel attention mechanism for text and images, 2025. URL https://arxiv.org/abs/2310.11960.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 4401–4410. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00453. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Karras_A_Style-Based_Generator_Architecture_for_Generative_Adversarial_Networks_CVPR_2019_paper.html.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020. URL http://proceedings.mlr.press/v119/katharopoulos20a.html.

Haopeng Li, Shitong Shao, Wenliang Zhong, Zikai Zhou, Lichen Bai, Hui Xiong, and Zeke Xie. PISA: Piecewise sparse attention is wiser for efficient diffusion transformers. *CoRR*, abs/2602.01077, 2026. doi: 10.48550/ARXIV.2602.01077. URL https://doi.org/10.48550/arXiv.2602.01077.

Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, Maneesh Agrawala, Ion Stoica, Kurt Keutzer, and Song Han. Radial attention: $\mathcal{O}(n \log n)$ sparse attention with energy decay for long video generation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/18a367688479c1b08b001584218a4443-Abstract-Conference.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 9992–10002. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00986. URL https://doi.org/10.1109/ICCV48922.2021.00986.

Tan M. Nguyen, Vai Suliafu, Stanley J. Osher, Long Chen, and Bao Wang. FMMformer: Efficient and flexible transformer via decomposed near-field and far-field attention. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 29449–29463, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f621585df244e9596dc70a39b579efb1-Abstract.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.org/10.1109/ICCV51070.2023.00387.

Hongyu Ren, Hanjun Dai, Zihang Dai, Mengjiao Yang, Jure Leskovec, Dale Schuurmans, and Bo Dai. Combiner: Full attention transformer with sparse computation cost. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 22470–22482, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/bd4a6d0563e0604510989eb8f9ff71f5-Abstract.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.org/10.1109/CVPR52688.2022.01042.

Daniel L. Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. *Phys. Rev. Lett.*, 73:814–817, Aug 1994. doi: 10.1103/PhysRevLett.73.814. URL https://link.aps.org/doi/10.1103/PhysRevLett.73.814.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(Volume 24, 2001):1193–1216, 2001. ISSN 1545-4126. doi: https://doi.org/10.1146/annurev.neuro.24.1.1193. URL https://www.annualreviews.org/content/journals/10.1146/annurev.neuro.24.1.1193.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Stability AI. Stable Diffusion 3.5 Medium. Model card, 2024. URL https://huggingface.co/stabilityai/stable-diffusion-3.5-medium.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=fR-EnKWL_Zb.

Philippe Tillet, Hsiang-Tsung Kung, and David D. Cox. Triton: an intermediate language and compiler for tiled neural network computations. In Tim Mattson, Abdullah Muzahid, and Armando Solar-Lezama (eds.), *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages, MAPL@PLDI 2019, Phoenix, AZ, USA, June 22, 2019*, pp. 10–19. ACM, 2019. doi: 10.1145/3315508.3329973. URL https://doi.org/10.1145/3315508.3329973.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

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. *CoRR*, abs/2306.09341, 2023. doi: 10.48550/ARXIV.2306.09341. URL https://doi.org/10.48550/arXiv.2306.09341.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=N8Oj1XhtYZ.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 30008–30022, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/fc1a36821b02abbd2503fd949bfc9131-Abstract.html.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 23078–23097. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.1126. URL https://doi.org/10.18653/v1/2025.acl-long.1126.

Zhanpeng Zeng, Sourav Pal, Jeffery Kline, Glenn Moo Fung, and Vikas Singh. Multi resolution analysis (MRA) for approximate self-attention. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25955–25972. PMLR, 2022. URL https://proceedings.mlr.press/v162/zeng22a.html.

Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, Joseph E Gonzalez, Jianfei Chen, and Jun Zhu. SLA: Beyond sparsity in diffusion transformers via fine-tunable sparse–linear attention. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 138288–138305, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/df77cca99e2785eafb5b2416d688ae24-Paper-Conference.pdf.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric P. Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/dfc310e81992d2e4cedc09ac47eff13e-Abstract-Conference.html.

Yifan Zhou, Zeqi Xiao, Tianyi Wei, Shuai Yang, and Xingang Pan. Trainable log-linear sparse attention for efficient diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9424–9433, 2026.

Zhenhai Zhu and Radu Soricut. H-Transformer-1D: Fast one-dimensional hierarchical attention for sequences. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 3801–3815. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.294. URL https://doi.org/10.18653/v1/2021.acl-long.294.

A Theoretical Details

Organization of the analysis. We develop the approximation results by following how pooling changes an attention distribution. Appendix A.1 first connects inverse-square image spectra to the logarithmic spatial model. Appendix A.2 then proves SFA's reverse-KL optimality for arbitrary finite logits. The quantitative error analysis proceeds through two probability inequalities (Appendix A.3), an exact separation of block-mass and within-block errors (Appendix A.4), and the scale-consistent bounds of Theorem 4.2 (Appendix A.5). Appendix A.6 expresses the same error through query-projected feature statistics. Appendices A.7–A.10 then explain the associated spatial bias: mass across distance shells, noise reduction by averaging, the frequency response after rotary encoding, and the special geometry of the two-dimensional logarithmic model.

Notation and scope. We fix one attention head and one query i, using the token indices, head dimensions, queries, keys, and values from Section 3. Positional encoding is included in the queries and keys. The partition $\mathcal{P}_i$ consists of disjoint nonempty blocks covering the effective key set; $m_B = |B|$, and each exact near-field interaction is a singleton block. Unless specified otherwise, sums over B range over $\mathcal{P}_i$, and the queries, keys, and values are held fixed. Logarithms are natural. For probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ with coordinates p_j and q_j, we use $D_{\mathrm{KL}}(\boldsymbol{p}\|\boldsymbol{q}) = \sum_j p_j \ln(p_j/q_j)$ and $\mathrm{TV}(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2}\sum_j |p_j - q_j|$ for Kullback–Leibler (KL) divergence and total variation (TV), respectively.

The operator $\mathbb{E}_B$ denotes uniform finite averaging over tokens in B, while $\mathbb{E}_{\boldsymbol{p}}$ denotes averaging with probabilities $\boldsymbol{p}$. For a scalar function f and a vector function $\boldsymbol{g}$ of the same sampled index, define $\mathrm{Var}_{\boldsymbol{p}}(f) = \mathbb{E}_{\boldsymbol{p}}(f - \mathbb{E}_{\boldsymbol{p}} f)^2$ and $\mathrm{Cov}_{\boldsymbol{p}}(f, \boldsymbol{g}) = \mathbb{E}_{\boldsymbol{p}}[(f - \mathbb{E}_{\boldsymbol{p}} f)(\boldsymbol{g} - \mathbb{E}_{\boldsymbol{p}} \boldsymbol{g})]$. Subscript B denotes the corresponding uniform block statistics. An unsubscripted $\mathbb{E}$ refers to the distribution specified for the random variable. These conventions also apply to block-indexed quantities when $\boldsymbol{p}$ is a distribution over blocks.

A.1 From inverse-square spectra to logarithmic correlation

We begin with the spatial model that motivates distance-dependent pooling. Consider a centered, stationary, isotropic scalar random field $\mathrm{x}(\boldsymbol{x})$ at positions $\boldsymbol{x} \in \mathbb{R}^2$. Suppose its power spectral density is $S(\boldsymbol{\omega}) = A\|\boldsymbol{\omega}\|_2^{-2}$ for angular frequencies satisfying $0 < k_{\min} \le \|\boldsymbol{\omega}\|_2 \le k_{\max} < \infty$, and zero elsewhere, where $A > 0$. The finite cutoffs make the covariance well defined while retaining the approximate inverse-square scaling of natural images (Ruderman & Bialek, 1994).

Let $\mathbf{0}$ denote the origin and use the Fourier convention $C(\boldsymbol{x}) = \mathbb{E}[\mathrm{x}(\mathbf{0})\mathrm{x}(\boldsymbol{x})] = (2\pi)^{-2}\int e^{\mathrm{i}\boldsymbol{\omega}^\top \boldsymbol{x}} S(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}$, where $\mathrm{i}^2 = -1$. Isotropy makes this covariance a function of distance $r = \|\boldsymbol{x}\|_2$. For $r > 0$, rotate the coordinates so that $\boldsymbol{x} = (r, 0)^\top$ and write the frequency in polar form as $\boldsymbol{\omega} = k(\cos\theta, \sin\theta)^\top$, where $k > 0$ and $\theta \in [0, 2\pi)$. Then $\boldsymbol{\omega}^\top \boldsymbol{x} = kr\cos\theta$ and the area element is $\mathrm{d}\boldsymbol{\omega} = k\,\mathrm{d}k\,\mathrm{d}\theta$. Multiplying this Jacobian by the spectral factor A/k^2 leaves the radial measure $A\,\mathrm{d}k/k$.

The angular integral is given by the integral representation of the order-zero Bessel function of the first kind,

$$J_0(t) = \frac{1}{2\pi}\int_0^{2\pi} e^{\mathrm{i}t\cos\theta}\,\mathrm{d}\theta, \qquad t \in \mathbb{R}.$$

Substituting the polar coordinates and then using this identity at $t = kr$ gives

$$\begin{aligned} C(r) &= \frac{A}{(2\pi)^2}\int_{k_{\min}}^{k_{\max}} \left[\int_0^{2\pi} e^{\mathrm{i}kr\cos\theta}\,\mathrm{d}\theta\right]\frac{\mathrm{d}k}{k} \\ &= \frac{A}{2\pi}\int_{k_{\min}}^{k_{\max}} J_0(kr)\,\frac{\mathrm{d}k}{k}. \end{aligned} \tag{7a}$$

To differentiate the radial covariance, set $t = kr$. Since $\mathrm{d}k/k = \mathrm{d}t/t$, Eq. (7a) becomes

$$C(r) = \frac{A}{2\pi}\int_{k_{\min}r}^{k_{\max}r} \frac{J_0(t)}{t}\,\mathrm{d}t.$$

The integrand now has no dependence on r, and r enters through both integration limits. The Leibniz rule gives the upper-limit contribution minus the lower-limit contribution, each multiplied by the derivative of its limit:

$$\begin{aligned} C'(r) &= \frac{A}{2\pi}\left[k_{\max}\frac{J_0(k_{\max}r)}{k_{\max}r} - k_{\min}\frac{J_0(k_{\min}r)}{k_{\min}r}\right] \\ &= -\frac{A}{2\pi r}\big[J_0(k_{\min}r) - J_0(k_{\max}r)\big]. \end{aligned} \tag{7b}$$

In the intermediate range $k_{\min}r \ll 1 \ll k_{\max}r$, the low-frequency term approaches one and the high-frequency term decays in amplitude. Thus Eq. (7b) gives $C'(r) \simeq -A/(2\pi r)$, which integrates to $C(r) \simeq C_0 - A\ln(r/r_*)/(2\pi)$, with reference distance $r_* > 0$ and additive constant C_0.

This calculation motivates the logarithmic spatial component of Eq. (2). Its coefficient in a learned attention logit also depends on feature projections, positional encoding, and dot-product scaling, so α_i remains a learned decay exponent. We next establish the approximation identity for arbitrary logits; the radial model will enter when we bound their variation within each block.

A.2 Projection and reverse-KL optimality

We use the same attention definitions as in Sections 3 and 4, restated here for the fixed query i. The full-attention logit is $z_{ij} = \boldsymbol{q}_i^\top \boldsymbol{k}_j/\sqrt{d}$. For a block B, let $\bar{\boldsymbol{k}}_B = \mathbb{E}_B \boldsymbol{k}_j$ and $\bar{\boldsymbol{v}}_B = \mathbb{E}_B \boldsymbol{v}_j$ be its mean key and value, and let $\bar{z}_{iB} = \mathbb{E}_B z_{ij} = \boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_B/\sqrt{d}$. Full attention assigns a weight a_{ij} to each token, while SFA assigns a probability mass $\widehat{p}_{iB}$ to each block:

$$\begin{aligned} Z_i &= \sum_{j=1}^{N} e^{z_{ij}}, & a_{ij} &= \frac{e^{z_{ij}}}{Z_i}, \\ \widehat{Z}_i &= \sum_{B\in\mathcal{P}_i} m_B e^{\bar{z}_{iB}}, & \widehat{p}_{iB} &= \frac{m_B e^{\bar{z}_{iB}}}{\widehat{Z}_i}. \end{aligned} \tag{8}$$

The corresponding outputs are $\boldsymbol{y}_i = \sum_j a_{ij}\boldsymbol{v}_j$ and $\widehat{\boldsymbol{y}}_i = \sum_B \widehat{p}_{iB}\bar{\boldsymbol{v}}_B$.

To compare the distributions on a common token set, expand each SFA block mass uniformly as $\widehat{a}_{ij} = \widehat{p}_{iB}/m_B$ for $j \in B$. Conversely, summing full-attention token weights gives the block mass $p_{iB} = \sum_{j\in B} a_{ij}$. Thus $\boldsymbol{a}_i = (a_{ij})_{j=1}^N$ and $\widehat{\boldsymbol{a}}_i = (\widehat{a}_{ij})_{j=1}^N$ are token probability vectors, whereas $\boldsymbol{p}_i = (p_{iB})_{B\in\mathcal{P}_i}$ and $\widehat{\boldsymbol{p}}_i = (\widehat{p}_{iB})_{B\in\mathcal{P}_i}$ are block probability vectors. As in Section 4, $\mathcal{U}_i$ denotes the family of token distributions uniform within each block of $\mathcal{P}_i$.

Define the block-averaging matrix $\boldsymbol{\Pi}_i \in \mathbb{R}^{N\times N}$ by $(\boldsymbol{\Pi}_i\boldsymbol{g})_j = m_B^{-1}\sum_{t\in B} g_t$ for $j \in B$ and $\boldsymbol{g} \in \mathbb{R}^N$. For the logit vector $\boldsymbol{z}_i = (z_{ij})_{j=1}^N$, $\boldsymbol{\Pi}_i\boldsymbol{z}_i$ repeats $\bar{z}_{iB}$ at every token in B. This representation expresses SFA directly through averaged logits on the original token set.

Proof of Theorem 4.1. Substituting the SFA block mass from Eq. (8) into $\widehat{a}_{ij} = \widehat{p}_{iB}/m_B$ gives

$$\widehat{a}_{ij} = \frac{e^{\bar{z}_{iB}}}{\widehat{Z}_i} = \mathrm{softmax}(\boldsymbol{\Pi}_i\boldsymbol{z}_i)_j, \qquad j \in B. \tag{9}$$

Mean pooling of the values preserves this expansion at the output: $\sum_B \widehat{p}_{iB}\bar{\boldsymbol{v}}_B = \sum_j \widehat{a}_{ij}\boldsymbol{v}_j$.

Any distribution $\boldsymbol{b} \in \mathcal{U}_i$ can likewise be written as $b_j = \pi_B/m_B$ for $j \in B$, where $\boldsymbol{\pi} = (\pi_B)_{B\in\mathcal{P}_i}$ is a probability vector over blocks, so $\sum_B \pi_B = 1$. Because the blocks partition the token set, we can group the token-level KL sum by block. Substituting $b_j = \pi_B/m_B$ and $a_{ij} = e^{z_{ij}}/Z_i$, then expanding the logarithm, gives

$$\begin{aligned}
D_{\mathrm{KL}}(\boldsymbol{b}\|\boldsymbol{a}_i) &= \sum_B \sum_{j\in B} \frac{\pi_B}{m_B} \ln \frac{\pi_B/m_B}{e^{z_{ij}}/Z_i} \\
&= \sum_B \sum_{j\in B} \frac{\pi_B}{m_B} \left(\ln \frac{\pi_B}{m_B} - z_{ij} + \ln Z_i \right) \\
&= \sum_B \left[\pi_B \ln \frac{\pi_B}{m_B} - \frac{\pi_B}{m_B} \sum_{j\in B} z_{ij} + \pi_B \ln Z_i \right] \\
&= \sum_B \pi_B \ln \frac{\pi_B}{m_B} - \sum_B \pi_B \bar{z}_{iB} + \ln Z_i.
\end{aligned} \tag{10}$$

Here the terms constant within B are repeated m_B times, the averaged logit is $m_B^{-1}\sum_{j\in B} z_{ij} = \bar{z}_{iB}$, and $\sum_B \pi_B \ln Z_i = \ln Z_i$. Terms from blocks with $\pi_B = 0$ are interpreted by continuity.

To identify the block-level divergence in Eq. (10), take logarithms of the SFA mass in Eq. (8): $\ln \widehat{p}_{iB} = \ln m_B + \bar{z}_{iB} - \ln \widehat{Z}_i$. Expanding the KL between the two block distributions therefore yields

$$\begin{aligned}
D_{\mathrm{KL}}(\boldsymbol{\pi}\|\widehat{\boldsymbol{p}}_i) &= \sum_B \pi_B \left[\ln \pi_B - \ln m_B - \bar{z}_{iB} + \ln \widehat{Z}_i \right] \\
&= \sum_B \pi_B \ln \frac{\pi_B}{m_B} - \sum_B \pi_B \bar{z}_{iB} + \ln \widehat{Z}_i,
\end{aligned}$$

where the last term again uses $\sum_B \pi_B = 1$. Comparing this expression with Eq. (10) leaves the difference between the two log normalizers:

$$D_{\mathrm{KL}}(\boldsymbol{b}\|\boldsymbol{a}_i) = D_{\mathrm{KL}}(\boldsymbol{\pi}\|\widehat{\boldsymbol{p}}_i) + \ln Z_i - \ln \widehat{Z}_i. \tag{11}$$

Only the first term on the right-hand side of Eq. (11) depends on $\boldsymbol{\pi}$. It is nonnegative and vanishes precisely at $\boldsymbol{\pi} = \widehat{\boldsymbol{p}}_i$, proving uniqueness of SFA's expanded distribution and giving

$$D_{\mathrm{KL}}(\widehat{\boldsymbol{a}}_i\|\boldsymbol{a}_i) = \ln \frac{Z_i}{\widehat{Z}_i}. \tag{12}$$

Since $\boldsymbol{b}$ and $\widehat{\boldsymbol{a}}_i$ share uniform conditionals within each block, $D_{\mathrm{KL}}(\boldsymbol{\pi}\|\widehat{\boldsymbol{p}}_i) = D_{\mathrm{KL}}(\boldsymbol{b}\|\widehat{\boldsymbol{a}}_i)$. Therefore the approximation also satisfies

$$D_{\mathrm{KL}}(\boldsymbol{b}\|\boldsymbol{a}_i) = D_{\mathrm{KL}}(\boldsymbol{b}\|\widehat{\boldsymbol{a}}_i) + D_{\mathrm{KL}}(\widehat{\boldsymbol{a}}_i\|\boldsymbol{a}_i). \tag{13}$$

□

Two consequences explain how the approximation responds to the partition. Refinement enlarges the family of blockwise uniform token distributions, so the minimum reverse KL cannot increase. If the logits are constant within each block, then $\mathbf{\Pi}_i \boldsymbol{z}_i = \boldsymbol{z}_i$ and Eq. (9) gives exact recovery of full attention, including its output for arbitrary values.

A.3 Two probability inequalities

The identity in Eq. (12) reduces token-level KL error to a ratio of normalizers. Bounding this ratio requires control of centered exponential moments; bounding attention outputs additionally requires control of changes in normalized probabilities. The following two lemmas provide these steps.

Lemma A.1 (Hoeffding's lemma; Hoeffding, 1963)**.** *If a scalar random variable* x *satisfies* $\mathbb{E}\mathrm{x} = 0$ *and lies in an interval of width* $\Delta \geq 0$*, then*

$$0 \leq \ln \mathbb{E} e^{\mathrm{x}} \leq \frac{\Delta^2}{8}. \tag{14}$$

Proof. Define $g(t) = \ln \mathbb{E} e^{t\mathrm{x}}$ for $t \in [0, 1]$. Centering gives $g(0) = g'(0) = 0$, and differentiating twice expresses $g''(t)$ as the variance of x under weights proportional to $e^{t\mathrm{x}}$. Reweighting preserves the support interval. Its midpoint is at most $\Delta/2$ from every supported value, so the second moment about that midpoint is at most $\Delta^2/4$. Variance is no larger than the second moment about any chosen center; hence $g''(t) \leq \Delta^2/4$. Integrating twice gives $g(1) = \int_0^1 (1-t) g''(t)\, \mathrm{d}t \leq \Delta^2/8$. Jensen's inequality gives $g(1) \geq \ln e^{\mathbb{E}\mathrm{x}} = 0$. □

Lemma A.2. *For finite vectors* $\boldsymbol{z}, \boldsymbol{u} \in \mathbb{R}^n$*, let* $\boldsymbol{q} = \mathrm{softmax}(\boldsymbol{z})$ *and* $\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{z} + \boldsymbol{u})$*. With* $\mathrm{osc}(\boldsymbol{u}) = \max_j u_j - \min_j u_j$*, where* $u_j = (\boldsymbol{u})_j$*,*

$$\mathrm{TV}(\boldsymbol{p}, \boldsymbol{q}) \leq \tanh\left(\frac{\mathrm{osc}(\boldsymbol{u})}{4}\right). \tag{15}$$

Proof. Draw an index j with probabilities $\boldsymbol{q}$ and define $\mathrm{x} = e^{u_j}$, $l = \min_j e^{u_j}$, $h = \max_j e^{u_j}$, and $\mu_{\mathrm{x}} = \mathbb{E}_{\boldsymbol{q}}\mathrm{x}$. The relation $p_j/q_j = e^{u_j}/\mu_{\mathrm{x}}$ gives $\mathrm{TV}(\boldsymbol{p}, \boldsymbol{q}) = \mathbb{E}_{\boldsymbol{q}}|\mathrm{x} - \mu_{\mathrm{x}}|/(2\mu_{\mathrm{x}})$. When $h > l$, convexity bounds $|x - \mu_{\mathrm{x}}|$ on $[l, h]$ by the line segment joining its endpoint values. Taking the expectation of this bound for a variable with mean μ_{x} gives

$$\mathrm{TV}(\boldsymbol{p}, \boldsymbol{q}) \leq \frac{(h - \mu_{\mathrm{x}})(\mu_{\mathrm{x}} - l)}{\mu_{\mathrm{x}}(h - l)} \leq \frac{\sqrt{h} - \sqrt{l}}{\sqrt{h} + \sqrt{l}}. \tag{16}$$

The first expression on the right is maximal at $\mu_{\mathrm{x}} = \sqrt{lh}$. The final ratio equals $\tanh(\ln(h/l)/4)$, and $\ln(h/l) = \mathrm{osc}(\boldsymbol{u})$. If $h = l$, the perturbation is constant and the two probability vectors coincide. □

We will apply TV to values through their coordinate ranges. For scalar values v_j of range $R = \max_j v_j - \min_j v_j$, subtracting the midpoint of this interval from every value leaves $\sum_j (p_j - q_j) v_j$ unchanged. The triangle inequality then gives $|\sum_j (p_j - q_j) v_j| \leq (R/2) \sum_j |p_j - q_j| = R\, \mathrm{TV}(\boldsymbol{p}, \boldsymbol{q})$. Applying this argument to each coordinate yields the vector bounds below.

A.4 Exact decomposition of approximation error

Pooling can change both the total probability assigned to a block and the weighting of values inside it. To separate these effects, write the logit as its block mean plus a centered residual. For $j \in B$, define

$$\epsilon_{ij} = z_{ij} - \bar{z}_{iB}, \qquad J_{iB} = \ln \mathbb{E}_B e^{\epsilon_{ij}}. \tag{17}$$

By construction, $\mathbb{E}_B \epsilon_{ij} = 0$. The original unnormalized block mass is $W_{iB} = \sum_{j \in B} e^{z_{ij}} = m_B e^{\bar{z}_{iB}} \mathbb{E}_B e^{\epsilon_{ij}}$, whereas SFA uses $\widehat{W}_{iB} = m_B e^{\bar{z}_{iB}}$. Thus $e^{J_{iB}} = W_{iB}/\widehat{W}_{iB}$ is exactly the multiplicative mass correction omitted by pooling. Jensen's inequality gives $J_{iB} \geq 0$.

Let $F_i = \sum_B \widehat{p}_{iB} e^{J_{iB}}$ be the average of these corrections under the SFA block distribution. Restoring the full mass of every block and renormalizing gives

$$p_{iB} = \frac{\widehat{p}_{iB} e^{J_{iB}}}{F_i}, \qquad F_i = \frac{Z_i}{\widehat{Z}_i} \geq 1. \tag{18}$$

Hence the full block distribution is an exponential reweighting of SFA's block distribution by J_{iB}. A correction identical across all blocks cancels in this normalization.

Within a block, full attention has conditional probabilities $s_{ij|B} = a_{ij}/p_{iB} = e^{\epsilon_{ij}}/(m_B e^{J_{iB}})$. Let $\boldsymbol{s}_{i|B} = (s_{ij|B})_{j\in B}$ and let $\boldsymbol{u}_B$ be the uniform distribution with entries $1/m_B$. The difference between their conditional value averages is

$$\boldsymbol{c}_{iB} = \sum_{j\in B} s_{ij|B} \boldsymbol{v}_j - \bar{\boldsymbol{v}}_B = \frac{\mathbb{E}_B[e^{\epsilon_{ij}}(\boldsymbol{v}_j - \bar{\boldsymbol{v}}_B)]}{e^{J_{iB}}} \in \mathbb{R}^{d_v}. \tag{19}$$

Full attention therefore outputs $\sum_B p_{iB}(\bar{\boldsymbol{v}}_B + \boldsymbol{c}_{iB})$. Subtracting SFA's output gives the exact decomposition

$$\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i = \sum_B (p_{iB} - \widehat{p}_{iB})\bar{\boldsymbol{v}}_B + \sum_B p_{iB}\boldsymbol{c}_{iB}. \tag{20}$$

The first term changes the mixture of block means; the second changes the value average inside each block. Singleton blocks have $\epsilon_{ij} = J_{iB} = 0$ and $\boldsymbol{c}_{iB} = \boldsymbol{0}$, so their within-block approximation is exact.

The divergence admits the same separation. Since $D_{\mathrm{KL}}(\boldsymbol{u}_B \| \boldsymbol{s}_{i|B}) = J_{iB}$, the KL chain rule and Eq. (18) yield

$$D_{\mathrm{KL}}(\widehat{\boldsymbol{a}}_i \| \boldsymbol{a}_i) = D_{\mathrm{KL}}(\widehat{\boldsymbol{p}}_i \| \boldsymbol{p}_i) + \sum_B \widehat{p}_{iB} J_{iB}, \tag{21}$$

$$D_{\mathrm{KL}}(\widehat{\boldsymbol{p}}_i \| \boldsymbol{p}_i) = \ln \mathbb{E}_{\widehat{\boldsymbol{p}}_i} e^{J_{iB}} - \mathbb{E}_{\widehat{\boldsymbol{p}}_i} J_{iB}. \tag{22}$$

The token-level divergence includes the average error of replacing each conditional distribution by a uniform one. The block-level divergence depends only on how the corrections J_{iB} vary across blocks. This distinction explains the different orders of the bounds in the next section.

A.5 Proof of scale-consistent accuracy

We now connect the preceding identities to relative spatial resolution. Recall $p_{iB} = \sum_{j\in B} a_{ij}$ and the value ranges from Section 4. Equivalently, writing $v_{jc} = (\boldsymbol{v}_j)_c$ for $c \in \{1, \ldots, d_v\}$, these ranges are $R_V = \max_c(\max_j v_{jc} - \min_j v_{jc})$ and $R_{V,B} = \max_c(\max_{j\in B} v_{jc} - \min_{j\in B} v_{jc})$. They measure the largest coordinate variation globally and within a block. The norm $\|\boldsymbol{g}\|_\infty = \max_c |g_c|$ uses the same coordinates. A common shift of all values changes neither these ranges nor the attention-output error.

Proof of Theorem 4.2. First, relative geometry controls the range of the logits. Define $r^+_{iB} = \max_{j\in B} r_{ij}$ for a pooled block. The triangle inequality gives $r^+_{iB} \leq r^-_{iB} + \ell_B \leq (1+\eta) r^-_{iB}$. With $u_{ij} = 0$ in the logarithmic model,

$$\Delta_{iB} := \max_{j\in B} z_{ij} - \min_{j\in B} z_{ij} = \alpha_i \ln \frac{r^+_{iB}}{r^-_{iB}} \leq \delta_i. \tag{23}$$

Subtracting the mean preserves this range. Singleton blocks have $\Delta_{iB} = 0$, including the query's own position, and require no radial assumption.

Second, the logit range controls the correction to each block mass. Lemma A.1 gives $0 \leq J_{iB} \leq \delta_i^2/8$. The projection identity and the definition of F_i then imply

$$D_{\mathrm{KL}}(\widehat{\boldsymbol{a}}_i \| \boldsymbol{a}_i) = \ln F_i = \ln \mathbb{E}_{\widehat{\boldsymbol{p}}_i} e^{J_{iB}} \leq \frac{\delta_i^2}{8}. \tag{24}$$

At the block level, Eq. (18) is a softmax perturbation with logits $(\ln \widehat{p}_{iB})_B$ and perturbations $(J_{iB})_B$. The range of the perturbations J_{iB} is at most $\delta_i^2/8$, so Lemma A.2 yields

$$\mathrm{TV}(\widehat{\boldsymbol{p}}_i, \boldsymbol{p}_i) \le \tanh\left(\frac{\delta_i^2}{32}\right). \tag{25}$$

Third, within each block, $\boldsymbol{s}_{i|B}$ is the softmax of $(\epsilon_{ij})_{j\in B}$, while $\boldsymbol{u}_B$ is the softmax of a zero vector. Applying the same lemma and the coordinate-range inequality gives

$$\|\boldsymbol{c}_{iB}\|_\infty \le R_{V,B}\,\mathrm{TV}(\boldsymbol{s}_{i|B}, \boldsymbol{u}_B) \le R_{V,B}\tanh\left(\frac{\delta_i}{4}\right). \tag{26}$$

The block means have coordinate ranges at most R_V. Applying the coordinate-range inequality to the first term of Eq. (20), and the triangle inequality to its second term, now gives

$$\begin{aligned}\|\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i\|_\infty &\le R_V\,\mathrm{TV}(\boldsymbol{p}_i, \widehat{\boldsymbol{p}}_i) + \sum_B p_{iB}\|\boldsymbol{c}_{iB}\|_\infty \\ &\le R_V\tanh\left(\frac{\delta_i^2}{32}\right) + \tanh\left(\frac{\delta_i}{4}\right)\sum_B p_{iB}R_{V,B}.\end{aligned} \tag{27}$$

Together with Eq. (24), this proves Eq. (6). □

The proof also controls unnormalized block masses and reveals a tighter KL order after summing probabilities within blocks.

Proposition A.3. *Under the assumptions of Theorem 4.2, the masses W_{iB} and $\widehat{W}_{iB}$ defined in Appendix A.4 satisfy*

$$0 \le \ln\frac{W_{iB}}{\widehat{W}_{iB}} = J_{iB} \le \frac{\delta_i^2}{8}, \qquad D_{\mathrm{KL}}(\widehat{\boldsymbol{p}}_i\|\boldsymbol{p}_i) \le \frac{\delta_i^4}{512}. \tag{28}$$

Proof. The first bound is the exponential-moment estimate used above. For the second, center the block correction by setting $\mu_J = \mathbb{E}_{\widehat{\boldsymbol{p}}_i} J_{iB}$. Equation (22) becomes $D_{\mathrm{KL}}(\widehat{\boldsymbol{p}}_i\|\boldsymbol{p}_i) = \ln \mathbb{E}_{\widehat{\boldsymbol{p}}_i} e^{J_{iB}-\mu_J}$. This centered variable has range at most $\delta_i^2/8$, so Lemma A.1 bounds its log moment by $(\delta_i^2/8)^2/8 = \delta_i^4/512$. □

Thus the expanded token distribution has a second-order KL bound in δ_i, while the block distribution has a fourth-order KL bound. Within-block value variation determines how this difference carries over to the output. With only a global range bound, Pinsker's inequality and the coordinate-range inequality give $\|\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i\|_\infty \le R_V \min\{1, \delta_i/4\}$. If, in addition, $R_{V,B} \le L_V\eta$ for a constant $L_V \ge 0$ uniform over blocks, Eq. (27) gives

$$\|\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i\|_\infty \le R_V\tanh\left(\frac{\delta_i^2}{32}\right) + L_V\eta\tanh\left(\frac{\delta_i}{4}\right) = O(\eta^2). \tag{29}$$

Here the limit is $\eta \to 0$: $\delta_i = \alpha_i\ln(1+\eta) = O(\eta)$, and each displayed term is quadratic. The constants are independent of token count when α_i, R_V, and L_V are uniformly bounded.

Content-dependent logits enter through the same range calculation. For a pooled block with residual u_{ij} in Eq. (2), define $\mathrm{osc}_{j\in B}\,u_{ij} = \max_{j\in B}u_{ij} - \min_{j\in B}u_{ij}$. Then $\Delta_{iB} \le \alpha_i\ln(1+\eta) + \mathrm{osc}_{j\in B}\,u_{ij}$. Replacing δ_i by a common upper bound on these ranges gives the same KL, TV, and output inequalities. This extension identifies within-block content variation as the additional quantity needed beyond relative geometry. The $O(\eta^2)$ output conclusion also holds when $\mathrm{osc}_{j\in B}\,u_{ij} = O(\eta)$ uniformly, together with the preceding within-block value condition.

A.6 Projected feature statistics and output error

The preceding section bounds residual logits through a spatial model. We now express their effect directly in terms of the learned keys and values, using the exact decomposition from Appendix A.4. This analysis applies to arbitrary finite logits.

For each block, define $\rho_{iB} = \max_{j\in B} |\epsilon_{ij}|$, $\sigma^2_{z,iB} = \mathbb{E}_B \epsilon^2_{ij}$, and $\sigma_{V,B} = \max_c \sqrt{\mathbb{E}_B (v_{jc} - \bar{v}_{Bc})^2}$, where $\bar{v}_{Bc} = (\bar{\boldsymbol{v}}_B)_c$. These are the maximum absolute logit residual, its variance, and the largest coordinate standard deviation of the values. Let $\boldsymbol{\Sigma}_{K,B} = \mathbb{E}_B[(\boldsymbol{k}_j - \bar{\boldsymbol{k}}_B)(\boldsymbol{k}_j - \bar{\boldsymbol{k}}_B)^\top] \in \mathbb{R}^{d\times d}$ and $\boldsymbol{C}_{KV,B} = \mathbb{E}_B[(\boldsymbol{k}_j - \bar{\boldsymbol{k}}_B)(\boldsymbol{v}_j - \bar{\boldsymbol{v}}_B)^\top] \in \mathbb{R}^{d\times d_v}$ be the key covariance and key–value cross-covariance. Substituting $\epsilon_{ij} = \boldsymbol{q}_i^\top (\boldsymbol{k}_j - \bar{\boldsymbol{k}}_B)/\sqrt{d}$ gives

$$\sigma^2_{z,iB} = \frac{\boldsymbol{q}_i^\top \boldsymbol{\Sigma}_{K,B} \boldsymbol{q}_i}{d}, \qquad \mathrm{Cov}_B(\epsilon_{ij}, \boldsymbol{v}_j) = \frac{\boldsymbol{C}_{KV,B}^\top \boldsymbol{q}_i}{\sqrt{d}}. \tag{30}$$

The query therefore selects which directions of key variation affect the approximation, and the cross-covariance determines how those variations couple to the values.

A bound from block statistics. To connect these quantities to the output, write $f_{iB} = e^{J_{iB}} = \mathbb{E}_B e^{\epsilon_{ij}}$ and $\boldsymbol{d}_{iB} = f_{iB}\boldsymbol{c}_{iB} = \mathbb{E}_B[e^{\epsilon_{ij}}(\boldsymbol{v}_j - \bar{\boldsymbol{v}}_B)]$. Centering removes the linear term from $f_{iB} - 1$ and the constant term from $\boldsymbol{d}_{iB}$. Taylor's theorem and coordinatewise Cauchy–Schwarz then give

$$0 \le f_{iB} - 1 \le \frac{1}{2} e^{\rho_{iB}} \sigma^2_{z,iB}, \qquad \|\boldsymbol{d}_{iB}\|_\infty \le e^{\rho_{iB}} \sigma_{z,iB}\sigma_{V,B}. \tag{31}$$

For the first inequality, use $f_{iB} - 1 = \mathbb{E}_B(e^{\epsilon_{ij}} - 1 - \epsilon_{ij})$ and $|e^x - 1 - x| \le e^\rho x^2/2$ for $|x| \le \rho$. For the second, write $\boldsymbol{d}_{iB} = \mathbb{E}_B[(e^{\epsilon_{ij}} - 1)(\boldsymbol{v}_j - \bar{\boldsymbol{v}}_B)]$, apply $|e^x - 1| \le e^\rho |x|$, and bound each coordinate by Cauchy–Schwarz.

The full output can be written as $\boldsymbol{y}_i = F_i^{-1} \sum_B \widehat{p}_{iB}(f_{iB}\bar{\boldsymbol{v}}_B + \boldsymbol{d}_{iB})$. Since $\sum_B \widehat{p}_{iB}(f_{iB} - 1) = F_i - 1$, subtracting $\widehat{\boldsymbol{y}}_i$ yields the useful identity

$$\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i = \frac{1}{F_i} \sum_B \widehat{p}_{iB} \big[(f_{iB} - 1)(\bar{\boldsymbol{v}}_B - \widehat{\boldsymbol{y}}_i) + \boldsymbol{d}_{iB}\big]. \tag{32}$$

Here $F_i \ge 1$, and $\widehat{\boldsymbol{y}}_i$ is a convex combination of block means, so $\|\bar{\boldsymbol{v}}_B - \widehat{\boldsymbol{y}}_i\|_\infty \le R_V$. Substituting Eq. (31) gives

$$\|\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i\|_\infty \le \sum_B \widehat{p}_{iB} e^{\rho_{iB}} \left(\frac{R_V}{2} \sigma^2_{z,iB} + \sigma_{z,iB}\sigma_{V,B} \right). \tag{33}$$

Each block contributes according to its SFA probability mass, its projected logit variation, and its value variation. Singleton contributions vanish. This gives a content-dependent counterpart to the geometric bound without requiring the radial logit model.

Leading terms for small residuals. Consider a family of attention problems as $\eta \to 0$. Assume $\rho_{iB} = O(\eta)$, $\max_{j\in B} \|\boldsymbol{v}_j - \bar{\boldsymbol{v}}_B\|_\infty = O(\eta)$, and $R_V = O(1)$, with constants uniform over blocks and tokens. All vector remainder estimates below use the infinity norm. These assumptions imply $\sigma_{z,iB} = O(\eta)$ and $\sigma_{V,B} = O(\eta)$, and Taylor expansion gives

$$f_{iB} - 1 = \frac{1}{2}\sigma^2_{z,iB} + O(\eta^3), \qquad \boldsymbol{d}_{iB} = \mathrm{Cov}_B(\epsilon_{ij}, \boldsymbol{v}_j) + O(\eta^3). \tag{34}$$

The scalar remainder is bounded by $e^{\rho_{iB}}\rho_{iB}\sigma^2_{z,iB}/6$, and the vector remainder by $e^{\rho_{iB}}\rho_{iB}\sigma_{z,iB}\sigma_{V,B}/2$. Their orders are therefore uniform. Since $\sum_B \widehat{p}_{iB} = 1$ and $\|\bar{\boldsymbol{v}}_B - \widehat{\boldsymbol{y}}_i\|_\infty \le R_V = O(1)$, probability weighting preserves the $O(\eta^3)$ remainders in Eq. (32). Its numerator is $O(\eta^2)$, while averaging the scalar expansion gives $F_i = 1 + O(\eta^2)$ and hence $F_i^{-1} = 1 + O(\eta^2)$. Replacing F_i^{-1} by one therefore contributes only an $O(\eta^4)$ term. Finally, $\widehat{\boldsymbol{y}}_i = \mathbb{E}_{\widehat{\boldsymbol{p}}_i} \bar{\boldsymbol{v}}_B$ identifies the variance-weighted term as a covariance:

$$\sum_B \widehat{p}_{iB}\sigma^2_{z,iB}(\bar{\boldsymbol{v}}_B - \widehat{\boldsymbol{y}}_i) = \mathrm{Cov}_{\widehat{\boldsymbol{p}}_i}(\sigma^2_{z,iB}, \bar{\boldsymbol{v}}_B).$$

Substituting these relations into Eq. (32) yields

$$\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i = \sum_B \widehat{p}_{iB} \mathrm{Cov}_B(\epsilon_{ij}, \boldsymbol{v}_j) + \frac{1}{2}\mathrm{Cov}_{\widehat{\boldsymbol{p}}_i}(\sigma^2_{z,iB}, \bar{\boldsymbol{v}}_B) + O(\eta^3). \tag{35}$$

The first term measures within-block coupling between logit and value residuals. The second measures whether blocks with larger logit variance also have systematically different mean values. Both are quadratic under the stated assumptions.

A second-order output bound also holds under a weaker condition on the values. Suppose $R_V = O(1)$ and $\rho_{iB} = O(\eta)$, and assume $\|\mathrm{Cov}_B(\epsilon_{ij}, \boldsymbol{v}_j)\|_\infty = O(\eta^2)$ uniformly. The remainder after the linear term in $\boldsymbol{d}_{iB}$ is at most $R_V e^{\rho_{iB}} \sigma_{z,iB}^2/2 = O(\eta^2)$. Hence $\boldsymbol{d}_{iB} = O(\eta^2)$ and $f_{iB} - 1 = O(\eta^2)$, so Eq. (32) again gives $\|\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i\|_\infty = O(\eta^2)$. This conclusion controls the output order; the more detailed expansion in Eq. (35) uses the additional within-block value regularity stated above.

A.7 Mass in logarithmic distance shells

The approximation bounds describe the error introduced by pooling. To understand how much attention is represented at each spatial scale, fix a query in Eq. (2) and set the content term to zero. We use a continuous approximation to the token grid, with constant density $\rho_0 > 0$, and consider complete annuli contained in the spatial domain at positive radius r. Let M_i denote the unnormalized attention mass on radial intervals. The radial logit assigns unnormalized weight $e^{b_i}(r/r_*)^{-\alpha_i}$ to a location at distance r. Multiplying by the annulus area $2\pi r\,\mathrm{d}r$ and the token density gives its unnormalized mass:

$$\mathrm{d}M_i(r) = 2\pi\rho_0 e^{b_i} r_*^{\alpha_i} r^{1-\alpha_i}\,\mathrm{d}r \propto r^{2-\alpha_i}\,\mathrm{d}\log r. \tag{36}$$

The last expression uses $\mathrm{d}r = r\,\mathrm{d}\log r$ and describes mass per logarithmic distance interval. In particular, let $M_i([r, 2r])$ be the mass in the annulus between radii r and $2r$. Integrating the radial density yields

$$M_i([r, 2r]) = 2\pi\rho_0 e^{b_i} r_*^{\alpha_i} \begin{cases} r^{2-\alpha_i}\dfrac{2^{2-\alpha_i} - 1}{2 - \alpha_i}, & \alpha_i \neq 2, \\ \log 2, & \alpha_i = 2. \end{cases} \tag{37}$$

At $\alpha_i = 2$, the increase in spatial population exactly balances the decay of individual weights, giving equal mass to successive distance doublings. Larger shells carry more mass when $\alpha_i < 2$ and less when $\alpha_i > 2$. This continuous calculation explains the population factor that SFA retains when it represents a region by one pooled key and value. On a finite token grid, that factor is the actual discrete count m_B. The $\log m_B$ correction restores this count after pooling, leaving the block's mean logit to determine its content-dependent relevance.

A.8 Mean pooling and conditional noise variance

Once the probability mass assigned to each block is fixed, its allocation among the constituent values determines how much noise reaches the output. For one value coordinate, model the observation at token j as $\mathrm{z}_j = s_j + \mathrm{n}_j$, where s_j is a deterministic signal and the noises n_j are independent, centered, and have common variance $\sigma^2 > 0$. Take a probability vector $\boldsymbol{w} = (w_j)_{j=1}^N$ that is fixed or independent of these noises; in the latter case, all calculations below condition on the weights. Prescribe block masses $\pi_B \geq 0$ with $\sum_B \pi_B = 1$ and require $\sum_{j\in B} w_j = \pi_B$.

Write the deviation from uniform weighting as $d_j = w_j - \pi_B/m_B$. Since $\sum_{j\in B} d_j = 0$, expanding the square gives $\sum_{j\in B} w_j^2 = \pi_B^2/m_B + \sum_{j\in B} d_j^2$. Independence of the noises gives $\mathrm{Var}(\sum_j w_j \mathrm{n}_j) = \sigma^2 \sum_j w_j^2$, so summing the blockwise identity yields

$$\mathrm{Var}\left(\sum_j w_j \mathrm{n}_j\right) = \sigma^2 \sum_B \frac{\pi_B^2}{m_B} + \sigma^2 \sum_B \sum_{j\in B} d_j^2. \tag{38}$$

For the prescribed masses, the first term is fixed and the second vanishes exactly at $w_j = \pi_B/m_B$. Uniform weights therefore uniquely minimize the conditional noise variance. The same calculation applies to independent centered vector noises $\mathbf{n}_j \in \mathbb{R}^{d_v}$ with common covariance matrix $\boldsymbol{\Sigma} = \mathbb{E}[\mathbf{n}_j \mathbf{n}_j^\top]$. Their output covariance is $\boldsymbol{\Sigma} \sum_j w_j^2$, which uniform weighting minimizes in the positive-semidefinite order.

Uniformization also changes the signal estimate. To quantify this tradeoff at the same block masses, define the target $y^* = \sum_j w_j s_j$ and uniformized weights $w_j^{\mathrm{U}} = \pi_B/m_B$ for $j \in B$. The estimators $\mathrm{z}_w = \sum_j w_j \mathrm{z}_j$ and $\mathrm{z}_{\mathrm{U}} = \sum_j w_j^{\mathrm{U}} \mathrm{z}_j$ have mean squared errors (MSE) $\mathrm{MSE}_w = \mathbb{E}(\mathrm{z}_w - y^*)^2$ and $\mathrm{MSE}_{\mathrm{U}} = \mathbb{E}(\mathrm{z}_{\mathrm{U}} - y^*)^2$. The first estimator is unbiased for this target, while the second has bias

$b_{\mathrm{U}} = \sum_j (w_j^{\mathrm{U}} - w_j) s_j$. Decomposing MSE into squared bias and variance gives

$$\mathrm{MSE}_{\mathrm{U}} - \mathrm{MSE}_w = b_{\mathrm{U}}^2 - \sigma^2 \sum_B \sum_{j \in B} (w_j - \pi_B / m_B)^2. \tag{39}$$

The variance reduction improves this signal estimate when it exceeds the squared bias. Spatially coherent signals make uniformization useful because their small within-block variation keeps that bias small.

Full attention and SFA can assign different masses to a block, so their comparison also includes the mass adjustment. Substituting the full-attention weights a_{ij} and the block-uniform SFA weights $\widehat{a}_{ij}$ into the variance decomposition gives the conditional noise variance of SFA minus that of full attention:

$$\sigma^2 \left[\sum_B \frac{\widehat{p}_{iB}^2 - p_{iB}^2}{m_B} - \sum_B \sum_{j \in B} \left(a_{ij} - \frac{p_{iB}}{m_B} \right)^2 \right]. \tag{40}$$

The first term depends on the block-probability changes controlled by Eq. (25); the second is the variance reduction obtained by making the weights uniform within each block. Under the stated conditional noise model, these identities isolate how signal coherence, within-block averaging, and mass changes interact. The generation experiments assess their net effect after the attention weights and features are learned. This provides a multiscale averaging mechanism alongside the geometry-adaptive harmonic shrinkage observed in learned diffusion denoisers (Kadkhodaie et al., 2024).

A.9 Rotary positional encoding and the frequency response of pooling

On the key side, SFA averages features after positional encoding. Rotary position embeddings (RoPE) rotate queries and keys according to position (Su et al., 2024), and linearity of the dot product ensures that averaging the rotated keys preserves the mean of the actual attention logits. To describe the spatial filtering induced by this operation, represent a pair of real key channels at position $\boldsymbol{x} \in \mathbb{R}^2$ by the complex scalar $\chi(\boldsymbol{x}) = e^{\mathrm{i}\boldsymbol{\omega}^\top \boldsymbol{x}} \chi^0(\boldsymbol{x})$. Here $\boldsymbol{\omega} = (\omega_x, \omega_y)^\top$ is the rotary frequency vector, χ^0 is the unrotated content, and $\mathrm{i}^2 = -1$.

Consider an axis-aligned rectangular block centered at $\boldsymbol{c}_B \in \mathbb{R}^2$, with n_x, n_y unit-spaced positions along its axes. For constant content $\chi^0(\boldsymbol{x}) = \chi_0 \in \mathbb{C}$, its mean $\bar{\chi}_B$ is determined entirely by the average rotary phase:

$$\bar{\chi}_B = e^{\mathrm{i}\boldsymbol{\omega}^\top \boldsymbol{c}_B} H_B(\boldsymbol{\omega}) \chi_0, \qquad H_B(\boldsymbol{\omega}) = \frac{\sin(n_x \omega_x / 2)}{n_x \sin(\omega_x / 2)} \frac{\sin(n_y \omega_y / 2)}{n_y \sin(\omega_y / 2)}. \tag{41}$$

To obtain this expression, consider one axis with n positions $t = 0, \ldots, n-1$, center $c = (n-1)/2$, and scalar frequency ω. The geometric-series identity gives $n^{-1} \sum_{t=0}^{n-1} e^{\mathrm{i}\omega t} = e^{\mathrm{i}\omega c} \sin(n\omega/2) / (n \sin(\omega/2))$. Multiplying the responses along the two axes and translating to the block center gives the displayed product and its center phase. At denominator zeros, the expressions are defined by continuity.

For spatially varying content, apply the same calculation to a Fourier component $\chi^0(\boldsymbol{x}) = \chi_0 e^{\mathrm{i}\boldsymbol{\nu}^\top \boldsymbol{x}}$, where $\boldsymbol{\nu} \in \mathbb{R}^2$ is its content frequency and χ_0 its complex amplitude. Multiplication by the rotary phase adds the frequency vectors before averaging, giving

$$\bar{\chi}_B = \chi_0 e^{\mathrm{i}(\boldsymbol{\nu}+\boldsymbol{\omega})^\top \boldsymbol{c}_B} H_B(\boldsymbol{\nu} + \boldsymbol{\omega}). \tag{42}$$

Superposition extends this response to a general key field. The window's magnitude response has its main lobe at zero frequency in the rotated representation, so its main lobe for the unrotated content is centered at $-\boldsymbol{\omega}$, with frequencies understood modulo 2π along each unit-grid axis. Its widths decrease in proportion to $1/n_x$ and $1/n_y$. Thus larger blocks apply narrower spatial filtering to the position-modulated keys. Values undergo the same averaging window without rotary modulation.

Distinct uses of high and low rotary frequencies have also been observed in trained language models (Barbero et al., 2025). In SFA, the additional averaging window makes the retained frequency resolution depend on block size and hence on distance. The approximation bounds account for this operation through the residuals of the rotated keys, while the exact local neighborhood retains individual token interactions.

A.10 TWO-DIMENSIONAL LOGARITHMIC GEOMETRY

We next examine the effect of region shape using continuous spatial averages of the logarithmic radial model. Place the query at the origin and suppress its index, writing $\alpha = \alpha_i$ and $b = b_i$. At position $\boldsymbol{x}$ with radius $r = \|\boldsymbol{x}\|_2 > 0$, the logit is $z(\boldsymbol{x}) = b - \alpha \ln(r/r_*)$. Since $\nabla^2 \ln r = -r^{-2} + r^{-2} = 0$ away from the query, z is harmonic. Its mean over a disk excluding the query therefore equals its center value. For a square, isotropic second moments make the average quadratic Taylor term proportional to the Laplacian, so that term also vanishes; symmetry leaves the fourth-order term as the first possible discrepancy.

To compute this term, take a continuous square B of side $s > 0$ centered at $\boldsymbol{c}_B$. Represent its center by the complex number $R_B = re^{\mathrm{i}\theta}$, where $r = \|\boldsymbol{c}_B\|_2$ and θ is the polar angle. Independent offsets $\mathrm{x}, \mathrm{y} \sim \mathrm{Uniform}[-s/2, s/2]$ define $\mathrm{u} = \mathrm{x} + \mathrm{iy}$, and the block mean is $\bar{z}_B = \mathbb{E}[z(\boldsymbol{c}_B + (\mathrm{x}, \mathrm{y})^\top)]$. The condition $s/(\sqrt{2}r) < 1$ places every offset within the convergence radius of

$$\ln |R_B + \mathrm{u}| = \ln r + \mathrm{Re} \sum_{n \geq 1} \frac{(-1)^{n+1}}{n} \left(\frac{\mathrm{u}}{R_B} \right)^n , \tag{43}$$

where Re takes the real part and n is a positive integer. Fourfold rotational symmetry gives $\mathbb{E}\mathrm{u}^n = 0$ unless n is divisible by four, and direct evaluation gives $\mathbb{E}\mathrm{u}^4 = 2s^4/80 - 6(s^2/12)^2 = -s^4/60$. Substituting this moment into the logarithm series and multiplying by $-\alpha$ yields, as $s/r \to 0$,

$$\bar{z}_B - z(\boldsymbol{c}_B) = -\frac{\alpha s^4}{240 r^4} \cos(4\theta) + O_\alpha\big((s/r)^8\big) . \tag{44}$$

The remainder begins at the next power allowed by the same rotational symmetry, and O_α allows its constant to depend on the fixed exponent α. For a centered regular square grid, interpret the mean as a uniform average over its discrete points. The same cancellation holds, with the coefficient of $\cos(4\theta)$ replaced by $\alpha(2\mu_4 - 6\mu_2^2)/(4r^4)$, where μ_2 and μ_4 are the second and fourth moments of a coordinate offset from the block center.

Attention mass also depends on exponentiating the variation around the mean logit. Returning to the continuous square, define the exponential correction $J_B = \ln \mathbb{E}[\exp(z(\boldsymbol{c}_B + (\mathrm{x}, \mathrm{y})^\top) - \bar{z}_B)]$. Its leading term is

$$J_B = \frac{\alpha^2 s^2}{24 r^2} + O_\alpha\big((s/r)^4\big) . \tag{45}$$

To see the source of this term, let ∇z denote the spatial gradient. The linear Taylor term has variance $(s^2/12)\|\nabla z(\boldsymbol{c}_B)\|_2^2 = \alpha^2 s^2/(12r^2)$. With the block center and α fixed, rescaling the offsets to a fixed unit square makes the centered logit residual analytic in $t = s/r$ near zero. Expanding its exponential and logarithm gives a quadratic term in J_B equal to half the variance of the linear Taylor term. Simultaneous reflection of both offsets preserves their joint distribution and is equivalent to replacing t by $-t$, so J_B is analytic and even in t. Its cubic term therefore vanishes, giving the $O_\alpha(t^4)$ remainder in Eq. (45). Thus Eq. (44) describes a fourth-order mean-logit discrepancy from the center, while Eq. (45) describes the second-order mass correction generated by within-block variation.

These local calculations are consistent across geometrically scaled blocks. To state the scaling law, let D be the spatial dimension, with $D = 2$ for images, and use the same radial logit at unit spatial density. For a bounded block $B \subset \mathbb{R}^D$ of positive volume at positive distance from the query, let $|B|$ denote its volume. Define $\bar{z}_B = |B|^{-1} \int_B z(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$, the exact mass $W_B = \int_B e^{z(\boldsymbol{x})}\, \mathrm{d}\boldsymbol{x}$, and the pooled mass $\widehat{W}_B = |B| e^{\bar{z}_B}$, so $J_B = \ln(W_B/\widehat{W}_B)$. For a dilation $B \mapsto \lambda B$ about the query with $\lambda > 0$,

$$W_{\lambda B} = \lambda^{D-\alpha} W_B, \qquad \widehat{W}_{\lambda B} = \lambda^{D-\alpha} \widehat{W}_B, \qquad J_{\lambda B} = J_B. \tag{46}$$

The change of variables in the integral contributes a volume factor λ^D, while the radial weight contributes $\lambda^{-\alpha}$, proving the first equality. The second follows from $|\lambda B| = \lambda^D |B|$ and $\bar{z}_{\lambda B} = \bar{z}_B - \alpha \ln \lambda$. Blocks related by these query-centered dilations therefore share a common relative mass correction, which cancels when normalization is restricted to such a family. For a fixed discrete collection of token positions, dilation changes distances while leaving the count unchanged, so the exact and pooled sums both scale as $\lambda^{-\alpha}$. Their relative correction is again invariant. The continuous and discrete statements describe the same radial homogeneity with their respective spatial measures.

B Details of the Triton Kernel Implementation

This section follows the computation of SFA and its gradients. After introducing the notation in Appendix B.1, we describe the forward pass in Appendix B.2: constructing block means, sharing their loads across queries, and accumulating one softmax across scales. Appendix B.3 follows the backward pass from probability recomputation to the final query, key, and value gradients. At each step, we explain how the hierarchy enables data reuse and reduces memory traffic. The description covers the default Triton implementation (Tillet et al., 2019), with fixed mean pooling and token-count correction.

B.1 Notation

Let n_b and n_h denote batch size and head count. The query, key, value, and output tensors $\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{Y}$ have shape $[n_b, n_h, N, d]$, with $N = L^2$ and $d_v = d$. At a fixed batch and head, their token-i slices are $\boldsymbol{q}_i$, $\boldsymbol{k}_i$, $\boldsymbol{v}_i$, and $\widehat{\boldsymbol{y}}_i$, respectively; batch and head indices are suppressed below. The supplied kernels support power-of-two grid sides $8 \le L \le 512$ and head dimensions $d \in \{64, 72\}$.

A level-0 base block contains $4 \times 4 = 16$ tokens. There are $G = N/16$ base blocks and $n_{\mathrm{lev}} = \log_2 L - 3$ pooled levels, indexed by $\ell = 0, \dots, n_{\mathrm{lev}} - 1$. A block at level ℓ contains $m_\ell = 16 \cdot 4^\ell$ tokens. The full quadtree extends to the root at level $n_{\mathrm{lev}}+1$; let $B_\ell(i)$ be its level-ℓ block containing query i. Let $\mathcal{N}(B)$ be the set of same-level blocks in the 3×3 neighborhood of B, clipped to the image boundary. The exact tokens form the union of the blocks in $\mathcal{N}(B_0(i))$; the pooled interaction list is

$$\mathcal{I}_\ell(i) = \mathrm{children}(\mathcal{N}(B_{\ell+1}(i))) \setminus \mathcal{N}(B_\ell(i)), \tag{47}$$

where children maps a block to its immediate quadtree children and a set of blocks to the union of their child sets. This gives at most 144 exact tokens and 27 pooled representatives per level, with the complete partition described in Section 3.2. The geometry is shared across batches and heads and cached for each grid size. At $L = 8$, the exact neighborhood already covers the image.

The partition follows from the nesting of the neighborhood regions. For $\ell = 0, \dots, n_{\mathrm{lev}}$, define the token set $\mathcal{R}_\ell(i) = \bigcup_{B \in \mathcal{N}(B_\ell(i))} B$. For $\ell < n_{\mathrm{lev}}$, the parent of every block in $\mathcal{N}(B_\ell(i))$ lies in $\mathcal{N}(B_{\ell+1}(i))$, so $\mathcal{R}_\ell(i) \subseteq \mathcal{R}_{\ell+1}(i)$. Since the children partition each parent, Eq. (47) gives

$$\bigcup_{B \in \mathcal{I}_\ell(i)} B = \mathcal{R}_{\ell+1}(i) \setminus \mathcal{R}_\ell(i).$$

These differences are disjoint across levels, and blocks within each level are disjoint. Together with the exact region $\mathcal{R}_0(i)$, their union is $\mathcal{R}_{n_{\mathrm{lev}}}(i)$. At level n_{lev}, the grid consists of 2×2 blocks, so every clipped 3×3 neighborhood covers all N tokens. The exact singleton tokens and pooled blocks therefore form a complete, disjoint partition, including at image boundaries.

B.2 Forward pass

Figure 4 shows how the mean hierarchy and shared candidate loads feed one online softmax across spatial scales.

Construct shared block means. Morton order interleaves the bits of the two spatial coordinates, placing every aligned quadtree block in a contiguous token interval. After positional encoding, the first pooling step averages each group of 16 keys and values. Every coarser block then averages its four child summaries:

$$\bar{\boldsymbol{k}}_B = \frac{1}{4} \sum_{C \in \mathrm{children}(B)} \bar{\boldsymbol{k}}_C, \qquad \bar{\boldsymbol{v}}_B = \frac{1}{4} \sum_{C \in \mathrm{children}(B)} \bar{\boldsymbol{v}}_C. \tag{48}$$

The implementation uses contiguous reshaping and mean reductions, retaining the input data type. The number of summaries decreases geometrically, with $\sum_{\ell=0}^{n_{\mathrm{lev}}-1} N/(16 \cdot 4^\ell) < N/12$ summary vectors per key or value hierarchy. Construction therefore costs $O(Nd)$ per head. Each summary is computed once and reused by all queries that access it.

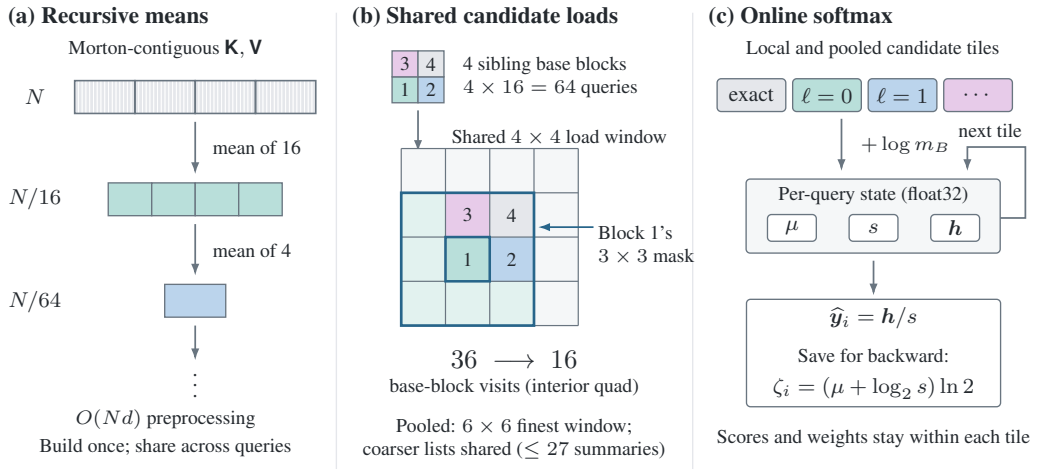


Figure 4: Forward computation of SFA, with batch and head indices suppressed. **(a)** Morton-contiguous keys and values form a shared hierarchy: the first reduction averages 16 tokens, and each subsequent reduction averages four child summaries. **(b)** One program processes four sibling base blocks, or 64 queries. For an interior quad, their exact neighborhoods cover 16 distinct base blocks, compared with 36 block visits when processed separately; per-query masks preserve the original neighborhoods. **(c)** Each query maintains its own running maximum μ, exponential sum s, and unnormalized output $\boldsymbol{h}$, shared across exact tokens and all pooled scales. The final state gives $\widehat{\boldsymbol{y}}_i = \boldsymbol{h}/s$ and the log-sum-exp statistic ζ_i saved for backward.

Share candidate loads across nearby queries. Each forward GPU program loads 64 queries from four sibling base blocks, called a quad. Their exact neighborhoods overlap: the union is a 4×4 window of base blocks. The program reads this window in 32-token tiles and masks each query back to its own 3×3 neighborhood. For an interior quad, this loads 16 base blocks instead of the $4 \cdot 9 = 36$ visits made by processing the four query blocks separately. The shared tiles retain regular matrix products while reducing repeated key and value loads.

The pooled traversal uses the same principle. At $\ell = 0$, the quad shares a 6×6 candidate window, visited in two 32-column tiles, and each query masks out its exact neighborhood. At $\ell \geq 1$, all four siblings have the same interaction list, so one list of at most 27 summaries supplies all 64 queries. The exact and finest pooled windows are generated from coordinates; coarser lists are read from the cached indices. These masks preserve each query's partition while avoiding a separately gathered candidate tensor for every query.

Accumulate one softmax across all scales. For a representative B, including singletons, define $\lambda_{iB} = \boldsymbol{q}_i^\top \bar{\boldsymbol{k}}_B/\sqrt{d} + \log m_B$ and its base-two equivalent $\beta_{iB} = \lambda_{iB} \log_2 e$. At pooled level ℓ, the count correction is simply $\log_2 m_B = 4 + 2\ell$. Following online tiled attention (Dao et al., 2022; Dao, 2024), each query maintains a running maximum μ, exponential sum s, and unnormalized output vector $\boldsymbol{h} \in \mathbb{R}^d$. A candidate tile $\mathcal{T}$ updates them as

$$\mu' = \max\left(\mu, \max_{B \in \mathcal{T}} \beta_{iB}\right), \tag{49}$$

$$s' = 2^{\mu - \mu'} s + \sum_{B \in \mathcal{T}} 2^{\beta_{iB} - \mu'}, \tag{50}$$

$$\boldsymbol{h}' = 2^{\mu - \mu'} \boldsymbol{h} + \sum_{B \in \mathcal{T}} 2^{\beta_{iB} - \mu'} \bar{\boldsymbol{v}}_B. \tag{51}$$

The initial state is $(\mu, s, \boldsymbol{h}) = (-\infty, 0, \boldsymbol{0})$, and empty masked tiles leave it unchanged. The state persists through the exact region and every pooled level, producing $\widehat{\boldsymbol{y}}_i = \boldsymbol{h}/s$. For training, the kernel also saves the natural-log log-sum-exp (LSE) statistic $\zeta_i = (\mu + \log_2 s) \ln 2 = \ln \widehat{Z}_i$. Scores

and probabilities are consumed within each tile, so the shared normalization requires no stored attention matrix.

Matrix operands use the input bfloat16 or float16 format, with float32 accumulation and normalization. In particular, probabilities are converted to the input format for the probability–value product. For $d = 72$, matrix tiles use 128 feature lanes with zero padding, while stored tensors and dot-product scaling retain dimension 72. These choices combine tensor-core matrix operations with compact storage. The training wrapper retains the mean hierarchy and LSE values, together with references to the input and output tensors.

B.3 Backward pass

The backward pass recomputes candidate scores using the saved inputs and means. Let $\mathcal{L}$ be the loss, $\boldsymbol{g}_i = \partial\mathcal{L}/\partial\widehat{\boldsymbol{y}}_i$, and $D_i = \boldsymbol{g}_i^\top \widehat{\boldsymbol{y}}_i$. The saved normalizer recovers each probability, which gives the loss derivative with respect to the natural-log score:

$$\widehat{p}_{iB} = \exp(\lambda_{iB} - \zeta_i), \qquad \gamma_{iB} := \frac{\partial \mathcal{L}}{\partial \lambda_{iB}} = \widehat{p}_{iB}\big(\boldsymbol{g}_i^\top \bar{\boldsymbol{v}}_B - D_i\big). \tag{52}$$

Treating the stored representatives as independent inputs to the attention computation, let $\boldsymbol{d}_B^K, \boldsymbol{d}_B^V \in \mathbb{R}^d$ denote their direct key and value gradients. The required accumulations are

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{q}_i} &= \frac{1}{\sqrt{d}} \sum_{B \in \mathcal{P}_i} \gamma_{iB} \bar{\boldsymbol{k}}_B, \\ \boldsymbol{d}_B^K = \frac{1}{\sqrt{d}} \sum_{i: B \in \mathcal{P}_i} \gamma_{iB} \boldsymbol{q}_i, &\qquad \boldsymbol{d}_B^V = \sum_{i: B \in \mathcal{P}_i} \widehat{p}_{iB} \boldsymbol{g}_i. \end{aligned} \tag{53}$$

The pooled gradients are subsequently propagated through the means. The fixed count bias requires no gradient; its effect is already included in $\widehat{p}_{iB}$. Figure 5 organizes the following four steps and shows how shared summaries reduce gradient traffic.

Compute query gradients (B1). One program owns the 16 queries of a base block. It computes and saves D_i, revisits the exact and pooled candidates, and accumulates each query gradient in float32 before storing it. Each query has a unique writer, while the saved D_i values are reused by both key and value gradient computations.

Combine gradients of shared summaries (B3). Several query programs can update the same pooled representative. Each program first combines the contributions of a group of query blocks. At $\ell = 0$, one program processes four sibling query blocks against a shared set of at most 36 candidates, retaining the per-query exclusion masks. At $\ell \geq 1$, identical interaction lists allow one program to combine $c_\ell = \min(4^\ell, 32)$ base query blocks. It loads the pooled keys and values once, processes 32 query rows at a time, and accumulates their contributions in registers. Atomic additions then write the combined gradients to initially zero float32 buffers. Relative to separate base-block programs, grouping reduces shared-candidate loads and gradient updates by a factor of c_ℓ at these coarser levels.

Propagate through the means (T4). All 16 tokens in a base block receive the same gradient share from every pooled ancestor. Let $g \in \{0, \ldots, G-1\}$ be its Morton index, and let $\boldsymbol{d}_{\ell,t}^K, \boldsymbol{d}_{\ell,t}^V$ denote the direct gradients of level-ℓ block t. Its ancestor index is $\lfloor g/4^\ell \rfloor$, so the per-token pooled contributions are

$$\boldsymbol{s}_g^K = \sum_{\ell=0}^{n_{\mathrm{lev}}-1} \frac{\boldsymbol{d}_{\ell,\lfloor g/4^\ell \rfloor}^K}{16 \cdot 4^\ell}, \tag{54}$$

$$\boldsymbol{s}_g^V = \sum_{\ell=0}^{n_{\mathrm{lev}}-1} \frac{\boldsymbol{d}_{\ell,\lfloor g/4^\ell \rfloor}^V}{16 \cdot 4^\ell}. \tag{55}$$

The denominator is the number of original tokens in the corresponding mean. A single kernel sums these ancestor contributions for both keys and values, accumulating in float32 and storing one vector per base block in the input format. This completes the chain rule through the hierarchy in compact arrays of shape $[n_b, n_h, G, d]$.

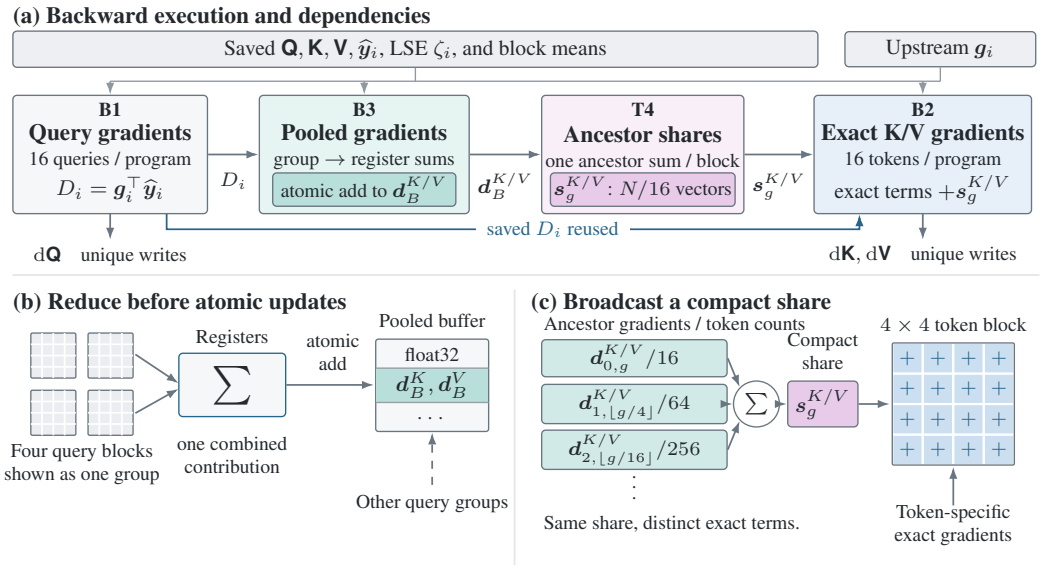


Figure 5: Backward computation of SFA, with batch and head indices suppressed; $\mathrm{d}\mathbf{Q}, \mathrm{d}\mathbf{K}, \mathrm{d}\mathbf{V}$ denote input gradients, and superscript K/V denotes either key or value quantities. The quantity $\boldsymbol{d}_{\ell,t}^{K/V}$ is the pooled gradient of block t at level ℓ. **(a)** B1 computes query gradients and the statistics D_i; B3 accumulates pooled gradients; T4 combines ancestor contributions into base-block shares; and B2 adds these shares to the exact gradients during the final write. **(b)** Query-block contributions are summed in registers before atomic updates to shared pooled-gradient buffers. **(c)** For base block g, ancestor gradients weighted by reciprocal token counts give per-token shares $\boldsymbol{s}_g^K, \boldsymbol{s}_g^V$. These shares occupy $G = N/16$ rows per batch and head and are broadcast to the 16 tokens of each base block.

Write the original key and value gradients (B2). Finally, one program owns the 16 original keys and values in each base block. Symmetry of the exact neighborhood identifies all queries that attend to those tokens directly. The program accumulates these exact contributions, adds $\boldsymbol{s}_g^K$ and $\boldsymbol{s}_g^V$ to each token's corresponding gradients, and performs the final stores. Thus the full-resolution gradients have unique writers, and the pooled contribution needs neither a full-resolution intermediate buffer nor a separate addition pass.

The saved mean hierarchy, pooled-gradient buffers, and base-block shares all contain $O(n_b n_h N d)$ floating-point entries. Integer geometry is cached separately and shared across batches and heads.

C Experimental Details and Additional Results

C.1 Hardware and evaluation settings

All throughput measurements use one NVIDIA RTX 5090 GPU, with the attention-kernel benchmark allocated 32 CPU cores and 64 GB of host memory. The software stack comprises PyTorch 2.12.0+cu132, CUDA 13.2 (`nvcc` 13.2.78), and NVIDIA driver 595.71.05. Kernel inputs use bfloat16 precision, and training-throughput measurements for PixelFlow and FFHQ enable gradient checkpointing. Every reported training time covers one complete optimizer step, and every generation rate measures the sampling pipeline end to end. ImageNet training, Stable Diffusion 3.5 Medium fine-tuning, and every generation-quality evaluation run on eight NVIDIA A800 GPUs with PyTorch 2.13.0+cu129 and NVIDIA driver 535.261.03, whereas the FFHQ models are trained on the same RTX 5090.

Alongside the timing measurements, we evaluate PixelFlow quality with the evaluation suite released with the ablated diffusion model (ADM; Dhariwal & Nichol, 2021), using 50,000 generated samples and Euler sampling to compute FID (Heusel et al., 2017) and IS (Salimans et al., 2016). In

this comparison, efficient-attention operators replace the 4096-token stage, leaving full attention in the other three stages. Table 2 collects the PixelFlow and FFHQ training and sampling settings, and Appendix C.7 specifies the PixelFlow-XL architecture used for the separate radial-attention study. The five-method ImageNet comparison uses 40 training epochs; full attention and SFA then continue through epoch 80 to examine whether their quality difference persists beyond this common budget.

Table 2: Training and sampling settings for PixelFlow and FFHQ, including exponential moving average (EMA) decay. PixelFlow expands its guidance scale across the four stages, so the listed value applies to the highest-resolution stage. The FFHQ models are unconditional and therefore use no guidance. The Stable Diffusion 3.5 Medium protocol is described in Appendix C.6.

Setting	PixelFlow from scratch	PixelFlow fine-tuning	FFHQ
Dataset	ImageNet	ImageNet	FFHQ
Image resolution	256×256	256×256	128, 256, 512
Model variant	PixelFlow-XL, 677M	PixelFlow-XL, 677M	DiT-S, 35M
Replaced stage	4096 tokens	4096 tokens	All attention layers
Initialization	Random	Released checkpoint	Table 3
Training hardware	8 NVIDIA A800	8 NVIDIA A800	1 NVIDIA RTX 5090
Global batch size	64	64	16, 4, 1
Training epochs	40; 80 for Full/SFA	5 scheduled	20 at 128/256; 2 at 512
Evaluated epochs	Every 5	3.5	Final
Optimizer	AdamW, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, gradient-norm clipping 1.0		
Learning rate	10^{-4}	$10^{-5} \rightarrow 10^{-6}$	10^{-4}
Learning-rate schedule	Constant	2 epochs constant, 3 epochs cosine	Constant
Weight decay	0	0	10^{-2}
EMA decay	0.9999	0.9999, scaled with the learning rate	0.9999 with warmup
Training precision	bfloat16	bfloat16	bfloat16
Sampling method	Euler	Euler	Euler
Sampling steps	30 per stage	30 per stage	20
Guidance scale	2.4	2.4	None
Evaluation sample count	50,000	50,000	50,000

The FFHQ models follow the resolution cascade of LLSA (Zhou et al., 2026). A full-attention DiT-S is trained at 32×32 from random initialization, and each higher-resolution stage continues from the preceding stage's checkpoint, with the training noise rescaled as the token grid grows. The four efficient-attention operators replace full attention from 128×128 onward, starting from the same 32×32 model and training for 20 epochs at each of the resolutions 128 and 256. At 512×512, we compare SFA and LLSA, each continuing for two epochs from its own 256×256 checkpoint. Table 3 lists this schedule.

C.2 Baseline configurations and kernel performance

We use LLSA as the reference for the selected-block budget, with its two-level configuration, block size 16, and selection counts $k_1 = k_2 = 8$ at the two hierarchy levels. SLA uses query and key/value blocks of 64 tokens and the softmax feature map in its linear branch, and VSA uses block shape $(1, 8, 8)$, whose unit temporal dimension corresponds to image inputs. Accounting for these different block sizes, the budget-matched SLA and VSA kernel settings select 4, 5, 8, and 20 key/value blocks for spatial grid sides $L = 64, 128, 256, 512$, respectively, matching LLSA up to block rounding.

At $L = 64$, we round the selected-block budget down to four blocks in the isolated kernel benchmark and up to five blocks in generation-quality experiments. The higher sparsity in the kernel benchmark favors SLA and VSA's throughput, and the denser attention in generation experiments favors their quality, making both comparisons more demanding for SFA. Model-level throughput uses the same

Table 3: Resolution cascade used for the FFHQ models, where one epoch covers the 70,000 FFHQ images. Each stage continues from the checkpoint of the preceding stage, and the compared attention operator replaces full attention from 128×128 onward. The rescaling factor multiplies the noise sampled during training at that resolution.

Stage resolution	32×32	128×128	256×256	512×512
Attention	Full	SFA, LLSA, SLA, VSA		SFA, LLSA
Initialized from	Random	32×32	128×128	256×256
Training epochs	40	20	20	2
Image batch size	64	16	4	1
Optimizer steps	43,760	87,500	350,000	140,000
Noise rescaling	1	2	4	8

five-block configuration as generation quality. At $L = 128, 256, 512$, rounding up and down yields identical counts of 5, 8, and 20 blocks, respectively, so the throughput and generation-quality settings coincide.

In addition to the budget-matched configurations, we evaluate fixed-rate alternatives that keep 5% of the key blocks for SLA and 10% for VSA at every grid size. We reuse the Triton implementations released by the authors of each baseline wherever possible. Those kernels require a power-of-two head dimension, so at the head dimension 72 of PixelFlow we zero-pad their features to 128 and keep the original softmax scale, either by rescaling the query (LLSA, VSA) or by passing the scale to the kernel (SLA), which leaves the attention they compute unchanged. SFA handles head dimension 72 inside its kernel, using similar methods. SFA and LLSA add no parameters to the backbone, whereas SLA and VSA each introduce a small module in every replaced layer. FMA has not yet been open-sourced, so we examine several key design differences between SFA and FMA primarily through the ablation experiments in Appendix C.8. With these configurations, Table 4 provides the numerical results underlying Figure 2(A,B) and adds the fixed-rate SLA and VSA comparisons at all four grid sizes.

Table 4: Complete kernel benchmark, expressed as speedup over the FlashAttention backend of PyTorch scaled dot-product attention (SDPA) for forward and backward computation separately. Inputs have shape $[n_b, n_h, N, d]$, with batch size $n_b = 4$, head count $n_h = 6$, query/key dimension $d = 64$, value dimension $d_v = d$, and $N = L^2$ tokens on an $L \times L$ grid, using bfloat16 precision. Full attention is the unit reference, and bold marks the best speedup.

	Forward speedup (×)				Backward speedup (×)			
Grid side L	64	128	256	512	64	128	256	512
Full	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SFA	**2.99**	**25.2**	**95.6**	**357.3**	**2.50**	**22.7**	**75.3**	**265.6**
LLSA	1.21	11.9	35.8	76.0	1.95	10.0	29.7	59.7
SLA, budget-matched	1.20	7.72	24.1	65.2	2.20	5.23	9.08	11.5
SLA, fixed 0.05 sparsity	1.22	6.38	11.7	15.7	2.23	4.82	7.39	8.79
VSA, budget-matched	0.75	12.6	39.2	60.6	1.72	17.7	43.7	80.9
VSA, fixed 0.10 sparsity	0.76	6.59	7.60	7.98	1.92	5.89	6.48	6.81

C.3 FFHQ: Training and inference throughput and generation quality

To examine how the kernel gains carry through to a complete generator, Table 5 reports runtime and memory use at all three FFHQ resolutions. Training image batch sizes of 16, 4, and 1 keep the token count per step fixed at 262,144, and generation uses image batch sizes of 20, 4, and 1. This fixed training token batch makes the per-token scaling directly comparable: SFA takes 3.88, 3.91, and 3.94 μs per token, whereas full-attention training time per token rises from 9.57 to 101.31 μs over the same resolution range. Figure 6 summarizes this difference alongside the generation speedup.

Across this fixed token batch, step time grows by only 1.47%, while $\log_2 N$ rises from 14 to 18, or 28.6%. SFA retains $O(N \log N)$ attention work, but its logarithmic contribution is small in the measured runtime. The remaining operations scale linearly with token count at fixed model size and dominate this regime, limiting further end-to-end gains from attention-only acceleration.

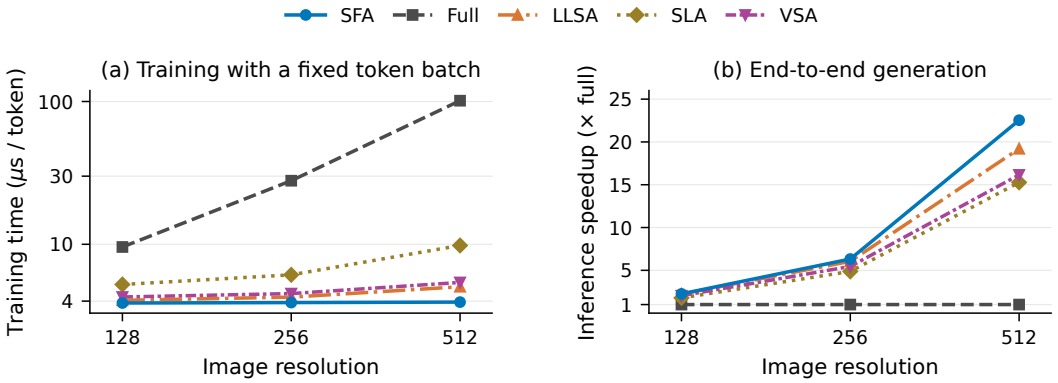


Figure 6: End-to-end scaling of pixel-space DiT-S on FFHQ. The left panel shows training time per token at a fixed batch of 262,144 tokens, using image batch sizes 16, 4, and 1 at resolutions 128, 256, and 512. The right panel shows generation speedup over full attention with image batch sizes 20, 4, and 1. SFA maintains nearly constant training time per token as resolution increases.

Table 5: Complete FFHQ end-to-end runtime measurements on one RTX 5090, with gradient checkpointing enabled for training. Speedups are relative to full attention at the same resolution, and generation rates and speedups retain the precision of the measurements. Bold marks the lowest time and memory or the highest throughput and speedup, including ties.

		Training			Generation		
Resolution	Method	ms/step	Speedup	GiB	images/s	Speedup	GiB
128	Full	2510	1.00	**10.57**	1.38	1.00	**3.44**
	SFA	**1017**	**2.47**	10.79	**3.10**	**2.25**	**3.44**
	LLSA	1062	2.36	10.60	3.06	2.22	**3.44**
	SLA	1371	1.83	12.27	2.45	1.78	4.15
	VSA	1121	2.24	11.41	2.81	2.04	3.75
256	Full	7308	1.00	**10.58**	0.12	1.00	**2.84**
	SFA	**1024**	**7.13**	10.80	**0.77**	**6.32**	**2.84**
	LLSA	1118	6.53	10.60	0.74	6.05	**2.84**
	SLA	1602	4.56	12.30	0.59	4.87	3.43
	VSA	1184	6.17	11.60	0.67	5.45	3.21
512	Full	26557	1.00	**10.60**	0.0085	1.00	**2.86**
	SFA	**1032**	**25.75**	10.83	**0.19**	**22.54**	2.87
	LLSA	1322	20.08	10.63	0.16	19.20	**2.86**
	SLA	2572	10.32	12.39	0.13	15.27	3.52
	VSA	1417	18.75	12.32	0.14	16.15	3.74

Generation quality. Table 6 and Figure 2(D) compare four efficient-attention variants in pixel-space DiT-S models trained on FFHQ at resolutions 128 and 256, with 20 epochs at each resolution in the cascade of Table 3. SFA achieves the lowest FID_{50k} at both resolutions, reaching 22.91 at resolution 128 and 30.39 at resolution 256. At resolution 256, the corresponding values for LLSA, SLA, and VSA are 32.03, 33.83, and 33.41, respectively.

At 512×512, SFA and LLSA continue from their respective 256×256 checkpoints for two epochs. SFA achieves FID_{50k} of 35.59, compared with 38.90 for LLSA, an improvement of 3.31. To-

Table 6: FID_{50k} for pixel-space DiT-S trained on FFHQ. The 128×128 and 256×256 results follow the 20-epoch stage at each resolution in Table 3 and use 50,000 generated samples. Lower values are better, and bold marks the lowest FID among the reported methods at each resolution.

Method	128×128	256×256
SFA	**22.91**	**30.39**
LLSA	23.24	32.03
SLA	24.99	33.83
VSA	23.17	33.41

gether with the four-method comparisons at resolutions 128 and 256, this result shows that SFA's generation-quality advantage accompanies the throughput gains in Table 5.

C.4 PixelFlow: Training and inference throughput and generation quality

Table 7 reports training and generation throughput and memory use for the complete PixelFlow model, with attention replaced in its 4096-token stage. These whole-model measurements include the cost of the stages that retain full attention.

Table 7: Complete PixelFlow end-to-end measurements using an image batch size of 4 and gradient checkpointing for training, and an image batch size of 16 for generation. Each efficient operator replaces the 4096-token stage. Speedups are relative to Full, computed from the listed step times and generation rates. Bold marks the lowest time and memory or the highest throughput and speedup, including ties.

	Training			Generation		
Method	ms/step	Speedup ($\times$)	GiB	images/s	Speedup ($\times$)	GiB
Full	308.6	1.00	**13.41**	0.178	1.00	**7.24**
SFA	**286.8**	**1.08**	13.42	**0.194**	**1.09**	**7.24**
LLSA	300.5	1.03	13.45	0.179	1.01	8.22
SLA	316.7	0.97	13.53	0.171	0.96	9.81
VSA	313.0	0.99	14.18	0.175	0.98	12.67

Table 8: FID_{50k} during PixelFlow training from scratch on ImageNet-256 through epoch 40 of the five-method comparison. Lower values indicate better sample quality, and bold marks the lowest FID among the five methods at each epoch.

Epoch	Full	SFA	LLSA	SLA	VSA
5	65.06	**64.77**	65.92	64.95	65.90
10	41.88	**40.66**	41.66	41.10	42.62
15	31.14	**29.45**	31.19	30.06	31.13
20	24.35	23.58	24.73	**23.43**	25.27
25	19.95	**19.25**	20.25	19.38	21.11
30	16.93	**16.12**	17.19	16.59	18.10
35	14.70	**14.03**	14.72	14.31	15.84
40	13.25	**12.51**	13.22	12.68	13.89

Generation quality. The corresponding generation-quality trajectories appear in Tables 8 and 9, which trace FID and IS throughout the five-method ImageNet-256 comparison through epoch 40. SFA achieves the lowest FID at seven of the eight checkpoints and the highest IS at every checkpoint. SLA briefly leads in FID, reaching 23.43 at epoch 20 compared with 23.58 for SFA, and remains a strong quality baseline at epoch 40 with FID 12.68 versus 12.51. SLA's quality gains over full

attention come with lower model throughput: Table 7 reports 316.7 versus 308.6 ms per training step and 0.171 versus 0.178 images/s during generation.

Table 9: Inception Score during PixelFlow training from scratch on ImageNet-256 through epoch 40 of the five-method comparison, evaluated on 50,000 generated samples. Higher values are better, and bold marks the highest IS among the five methods at each epoch.

Epoch	Full	SFA	LLSA	SLA	VSA
5	19.02	**19.88**	19.45	19.43	19.07
10	30.58	**32.90**	31.81	31.38	30.40
15	42.43	**45.94**	42.74	44.04	42.63
20	54.67	**59.15**	54.97	56.72	53.60
25	65.87	**70.53**	65.89	68.25	63.38
30	76.78	**81.53**	76.63	78.45	73.82
35	86.24	**90.93**	87.33	88.83	82.29
40	94.23	**101.20**	94.72	96.13	91.33

Beyond the common five-method budget, Table 10 follows full attention and SFA at epochs 45–80 to examine the effect of longer training. SFA maintains lower FID and higher IS at every recorded checkpoint, showing that its advantage over full attention persists through this extended period.

Table 10: Extended ImageNet-256 training of full attention and SFA beyond the 40-epoch five-method comparison, with both metrics evaluated on 50,000 generated samples. Bold marks the lower FID and higher IS within each two-method comparison.

	Full attention		SFA	
Epoch	$\text{FID}_{50k} \downarrow$	IS $\uparrow$	$\text{FID}_{50k} \downarrow$	IS $\uparrow$
45	11.78	101.76	**11.17**	**107.74**
50	10.84	109.05	**10.22**	**114.73**
55	9.99	114.92	**9.42**	**121.56**
60	9.20	119.94	**8.68**	**126.25**
65	8.65	125.24	**8.10**	**133.08**
70	8.22	129.53	**7.74**	**136.86**
75	8.10	131.03	**7.57**	**139.00**
80	7.55	135.80	**7.11**	**143.97**

C.5 Fine-tuning a pretrained PixelFlow model

Table 11: Pretrained and adapted PixelFlow checkpoints, evaluated with 50,000 generated samples. The adapted models share the same initialization, learning-rate schedule, and 3.5-epoch budget. Among the fine-tuned models, bold marks the lower FID and higher IS.

Model	Fine-tuning epochs	$\text{FID}_{50k} \downarrow$	IS $\uparrow$
Pretrained full attention	0	2.15	277.84
Fine-tuned full attention	3.5	**2.28**	273.25
Fine-tuned SFA	3.5	2.29	**284.71**

To evaluate adaptation of pretrained weights, we start from the original checkpoint with $\text{FID}_{50k} = 2.15$ and $\text{IS} = 277.84$ under our Euler sampling protocol. Evaluating the same checkpoint with the original work's dopri5 solver gives FID 2.00; its higher sampling cost leads us to use Euler sampling throughout the paper unless stated otherwise. Both full attention and SFA are fine-tuned from these weights at a learning rate of 10^{-5} for two epochs, followed by a three-epoch cosine decay to 10^{-6}. Table 11 compares the resulting models at epoch 3.5 under this shared budget, using

50,000 generated samples for each evaluation. SFA reaches FID 2.29 and IS 284.71, retaining FID comparable to full attention at 2.28 and improving on its IS of 273.25.

C.6 Fine-tuning Stable Diffusion 3.5 Medium

We extend the evaluation to pretrained text-to-image generation with Stable Diffusion 3.5 Medium (Stability AI, 2024; Esser et al., 2024), retaining its original image tokenization. Its variational autoencoder (VAE) compresses each spatial dimension by a factor of 8, after which the transformer embeds 2×2 latent patches, producing an effective patch size of 16×16 pixels. Image resolutions of 1024×1024 and 2048×2048 consequently give 64×64 and 128×128 image-token grids with 4096 and 16,384 image tokens, respectively. Joint attention combines these image tokens with the text tokens.

To distinguish the effect of attention replacement from subsequent adaptation, we compare six attention and fine-tuning settings. Of the 24 transformer blocks, the first 13 contain a separate image self-attention module alongside the joint attention. Full retains the original attention, and the self-attention-only setting (SA) replaces those 13 modules with SFA. ALL extends the replacement to the image-to-image interactions inside the joint attention of all 24 blocks, preserving the original computation for text-to-text, text-to-image, and image-to-text interactions. Image and text contributions share one softmax normalizer for each query. Because ALL covers more of the quadratic attention work, it reaches a larger speedup than SA at both resolutions (Table 12). The suffix +ft identifies a fine-tuned checkpoint. The quality tables group Full, SA, and ALL together, followed by Full+ft, SA+ft, and ALL+ft, so that each group compares methods within a shared adaptation budget.

Fine-tuning protocol. For the adaptation comparison, we fine-tune Full, SA, and ALL for one epoch on the first 30 shards of the Flux-generated dataset (about 12,000 steps at a global batch size of 16), using eight NVIDIA A800 GPUs and a learning rate of 10^{-5}. Training uses AdamW with weight decay 10^{-4}, gradient-norm clipping 1.0, bfloat16 parameters and optimizer states, and gradient checkpointing. All models start from the same pretrained checkpoint, and we evaluate their raw checkpoint weights.

Runtime. For these same replacement scopes, Table 12 compares training time, generation throughput, and memory use on one NVIDIA RTX 5090 GPU at image resolutions 1024×1024 and 2048×2048. Training uses image batch sizes 2 and 1, respectively, with gradient checkpointing enabled at both resolutions; generation uses batch sizes 4 and 1. At 2048×2048, ALL reduces the training step from 2214.7 to 1054.7 ms and increases generation throughput from 0.02414 to 0.04767 images/s, with inference memory unchanged at 24.75 GiB. Generation follows the released sampling defaults of 40 Euler steps at guidance scale 4.5, and each rate covers one complete pipeline call.

Table 12: Stable Diffusion 3.5 Medium runtime on one RTX 5090. Image resolutions 1024×1024 and 2048×2048 give 4096 and 16,384 image tokens with effective 16×16 pixel patches. Training batch sizes are 2 and 1, and generation batch sizes are 4 and 1, respectively. Speedups are relative to Full at the same resolution and can be reproduced from the listed times and rates. Bold marks the lowest time and memory or the highest throughput and speedup at each resolution, including ties.

		Training			Generation		
Image resolution	Setting	ms/step	Speedup (×)	GiB	images/s	Speedup (×)	GiB
1024×1024	Full	686.5	1.00	**26.69**	0.1549	1.00	**23.27**
	SA	644.4	1.07	26.72	0.1629	1.05	**23.27**
	ALL	**567.7**	**1.21**	26.75	**0.1852**	**1.20**	**23.27**
2048×2048	Full	2214.7	1.00	**27.80**	0.02414	1.00	**24.75**
	SA	1797.4	1.23	27.85	0.02913	1.21	**24.75**
	ALL	**1054.7**	**2.10**	27.93	**0.04767**	**1.97**	**24.75**

DPG-Bench. To assess generation quality, we evaluate prompt following with the Dense Prompt Graph Benchmark (DPG-Bench; Hu et al., 2024), which scores alignment with the entities, attributes, and relationships in detailed prompts. Every benchmark image below is generated at 1024×1024 with the same 40 Euler steps and guidance scale 4.5. Following the DPG-Bench protocol, we generate four images for each of 1,065 prompts and score them as a 2×2 grid. Table 13 presents category and overall scores, and Table 14 provides the subcategory breakdown under Attribute, Entity, Relation, and Other, with higher scores indicating better alignment. Direct SA replacement gives an overall score of 84.35, slightly below Full at 84.62, while ALL gives a comparable score of 84.74. Under the shared fine-tuning schedule, SA+ft reaches 85.68 and ALL+ft reaches 85.51, matching or exceeding the equally fine-tuned Full+ft at 85.51.

Table 13: DPG-Bench category and overall scores for Stable Diffusion 3.5 Medium. The upper group compares Full, SA, and ALL without fine-tuning; the lower group compares them after the same one-epoch fine-tuning schedule. Bold marks the highest value within each group and metric, including ties. The subcategory breakdown follows in Table 14.

Setting	Attribute	Global	Relation	Entity	Other	**Overall**
Full	**88.73**	84.50	93.17	89.94	82.40	84.62
SA	88.53	84.50	93.14	89.77	**84.40**	84.35
ALL	88.51	**86.32**	**93.60**	**90.23**	79.60	**84.74**
Full+ft	**89.34**	86.02	92.98	90.87	**86.00**	85.51
SA+ft	89.06	84.50	93.33	90.85	84.40	**85.68**
ALL+ft	89.16	**86.32**	**93.41**	**90.98**	82.80	85.51

Table 14: DPG-Bench subcategory scores for the same six settings as Table 13. Bold marks the highest value in each row separately among the three settings without fine-tuning and among the three fine-tuned settings, including ties.

		No fine-tuning			After fine-tuning		
Category	Subcategory	Full	SA	ALL	Full+ft	SA+ft	ALL+ft
Attribute	Color	**92.82**	92.72	92.17	**94.18**	93.63	93.18
	Shape	81.66	**82.53**	80.79	81.66	**82.53**	78.60
	Size	72.73	70.66	**73.55**	68.18	68.60	**71.49**
	Texture	88.14	**88.69**	87.78	88.69	88.69	**88.81**
	Other	86.82	85.27	**87.71**	87.49	86.82	**88.37**
Entity	Part	85.16	84.38	**86.13**	83.59	**83.98**	82.62
	State	83.84	83.32	**84.37**	81.94	82.37	**83.21**
	Whole	91.67	91.65	**91.84**	**93.44**	93.29	**93.44**
Relation	Non-spatial	86.79	86.79	**89.31**	88.05	87.42	**89.94**
	Spatial	93.59	93.55	**93.88**	93.30	**93.71**	93.63
Other	Count	79.50	**81.00**	76.50	**83.00**	81.50	80.00
	Text	94.00	**98.00**	92.00	**98.00**	96.00	94.00

Human preference. Complementing the prompt-alignment results, Table 15 reports predicted human preference using human preference score (HPS) v2.1 (Wu et al., 2023) across anime, concept art, paintings, and photographs. Each style contributes 800 prompts with one image each, and the overall score averages the four style means. Direct SA replacement slightly lowers the overall score from 29.967 to 29.596, and ALL produces a larger decrease to 27.927 despite its comparable DPG-Bench score. After fine-tuning, SA+ft reaches 30.242 and ALL+ft reaches 30.262, closely matching Full+ft at 30.261. The two benchmarks therefore show that both replacement scopes recover the quality of the identically fine-tuned original model.

Table 15: HPS v2.1 scores for Stable Diffusion 3.5 Medium, with higher values indicating stronger predicted human preference. Within each adaptation group, either without fine-tuning or after the shared one-epoch fine-tuning schedule, bold marks the highest value, including ties.

Setting	Anime	Concept art	Paintings	Photo	**Overall**
Full	**31.399**	**30.526**	**30.478**	**27.466**	**29.967**
SA	31.162	30.128	29.954	27.141	29.596
ALL	29.426	27.976	27.070	27.237	27.927
Full+ft	31.066	**30.499**	30.512	28.968	30.261
SA+ft	**31.072**	30.450	30.489	28.955	30.242
ALL+ft	30.975	30.438	**30.590**	**29.044**	**30.262**

C.7 Radial attention statistics

To relate the generation experiments to learned spatial structure, we measure attention in a pretrained PixelFlow-XL with 28 layers, 16 heads, and head dimension 72. Its final stage operates at 256×256 resolution, where 4×4 patches form a 64×64 token grid. We select one image from each of the 1000 ImageNet training classes, then resize each image by its shorter side, center-crop it to 256×256, and normalize it to $[-1, 1]$.

To follow the final stage's training interpolation, let $\boldsymbol{f}$ denote the three-channel image flattened into a vector and $\bar{\boldsymbol{f}}$ the vector obtained by twofold bilinear downsampling followed by nearest-neighbor upsampling. We introduce a noise vector $\boldsymbol{\epsilon}$ with the same dimension as $\boldsymbol{f}$ and independent standard normal entries. At normalized within-stage time $\tau = 0.95$, the resulting noisy image vector supplied to the model is

$$\mathbf{f}_\tau = \tau \boldsymbol{f} + (1 - \tau)(0.6\bar{\boldsymbol{f}} + 0.4\boldsymbol{\epsilon}). \tag{56}$$

Using this input with its corresponding timestep and class label, we retain the original scaled dot-product attention computation for model outputs and additionally compute the attention probabilities from the queries and keys. These probabilities are collected for every head in each of the 28 layers.

For a fixed image, layer, and head, the softmax attention matrix $\boldsymbol{A} = (a_{ij})_{i,j=1}^{N}$ contains probabilities a_{ij} assigned by query i to key j, with $N = 4096$. To isolate interactions between distinct spatial positions, we exclude the diagonal and renormalize each row, obtaining

$$\widetilde{a}_{ij} = \frac{a_{ij}}{1 - a_{ii}}, \qquad j \neq i, \qquad \sum_{j \neq i} \widetilde{a}_{ij} = 1. \tag{57}$$

We convert these query-wise probabilities into radial statistics by averaging over images and token pairs at equal Euclidean distance. The pairs are grouped by the 1576 exact squared-distance values of the 64×64 lattice, with the diagonal kept separate, and their attention mass is accumulated. For plotting, we combine positive distances by quantiles of token-pair counts, requesting 100 bins. The plotted value for each bin is its attention mass divided by the number of token pairs, and the horizontal coordinate is the pair-count-weighted mean distance. This weighting accounts for the discrete multiplicities of lattice distances, with a uniform reference of $1/(4096 - 1)$ for the renormalized rows.

This aggregation produces a curve $s_{\ell h}(r)$ for each layer–head pair, where $\ell \in \{1, \ldots, 28\}$ indexes the layer, $h \in \{1, \ldots, 16\}$ indexes the head, and r is spatial distance. Let $\mathcal{H}(r)$ collect the pairs with a positive value in the corresponding distance bin. Their geometric mean summarizes the spatial pattern across layers and heads and is plotted in Figure 1(C) against $\log_2 r$, with r measured in pixels:

$$s_{\mathrm{geo}}(r) = \exp\left[\frac{1}{|\mathcal{H}(r)|} \sum_{(\ell,h) \in \mathcal{H}(r)} \log s_{\ell h}(r)\right]. \tag{58}$$

For an individual or aggregate radial curve $s(r)$, we fit $\log s = b_{\mathrm{fit}} - \alpha \log r$, with intercept b_{fit} and decay exponent α, over 8–160 pixels. The exponent follows the convention $s(r) \propto r^{-\alpha}$. To compare the radial decay forms, we also fit $s = a_{\mathrm{fit}} \log r + c_{\mathrm{fit}}$, where a_{fit} and c_{fit} are fitted coefficients, and evaluate both coefficients of determination R^2 from residuals on the original s scale. Table 16 reports

this comparison together with a random-initialization measurement using the same architecture, images, and interpolation. The randomly initialized model's geometric-mean curve stays close to the uniform reference, in contrast to the pronounced spatial decay learned by the pretrained model.

Table 16: Power-law and logarithmic fits to radial attention over 2–40 patches, with decay exponent defined by $s(r) \propto r^{-\alpha}$. The final row summarizes the individual fitted curves through their median parameters and fit scores.

Weights	Aggregation	α	Power-law R^2	Logarithmic R^2
Pretrained	Geometric mean	1.414	0.974	0.589
Pretrained	Pointwise median	1.306	0.986	0.635
Random	Geometric mean	-0.009	0.325	0.328
Pretrained	Median over 448 individual fits	1.437	0.929	0.673

The aggregate trend is also reflected in individual layer–head curves: 321 of 448 favor the power-law fit over logarithmic decay, and 261 have power-law $R^2 > 0.9$. For the 16 heads, the median decay exponents across layers range from 0.993 to 1.692. To assess sensitivity to the distance grouping, we also use logarithmically spaced and linearly spaced bins, obtaining geometric-mean exponents of 1.205 and 1.324, respectively. Their power-law R^2 values are 0.829 and 0.977, compared with 0.622 and 0.748 for logarithmic decay, so the spatial trend persists across these binning choices.

These measurements of the radial structure of normalized attention support SFA's scale-free prior and connect the learned attention pattern to the image-statistics motivation. The relationship between image covariance and learned logits provides the corresponding theoretical interpretation.

C.8 Ablations

We examine count correction, normalization across scales, far-field context, and block aggregation through four variants of SFA. We first report their generation quality and then analyze how these design choices affect computational efficiency.

Variant 1 removes the $\log|B|$ correction while retaining mean pooling and shared softmax. **Variant 2** retains mean pooling but applies softmax separately at each scale and sums the resulting outputs. **Variant 3** retains only SFA's exact local neighborhood: the 3×3 neighborhood of 4×4 base blocks, covering 12×12 tokens away from image boundaries. It applies softmax over these local tokens and omits all pooled far-field representatives, with the neighborhood clipped at image boundaries. **Variant 4** replaces mean pooling with learnable aggregation and omits the $\log|B|$ correction, as FMA does (Kang et al., 2025), while retaining shared softmax.

FFHQ training. We compare SFA with Variants 1–3 at FFHQ-128, following the cascade in Table 3 and the settings in Table 2. The four operators therefore start from the same 32×32 model and share the same 20-epoch budget. Figure 7 traces FID_{10k} every two epochs through epoch 20, and Table 17 reports the complete trajectory together with the final FID_{50k}. SFA achieves the lowest FID_{10k} at every recorded checkpoint. At epoch 20, removing count correction increases FID_{50k} from 22.91 to 25.91, while separate softmax increases it to 27.99. Retaining only the exact local neighborhood raises it to 45.76, showing the substantial contribution of pooled far-field information beyond the local attention branch.

Direct replacement in Stable Diffusion 3.5 Medium. To assess pretrained-weight reuse, we also compare SFA with Variants 1 and 2 by directly replacing only image self-attention in Stable Diffusion 3.5 Medium, preserving joint attention and applying no fine-tuning. This follows the SA replacement setting in Appendix C.6. Table 18 reports HPS v2.1 for anime, concept art, paintings, and photographs, together with the overall score, under the same evaluation protocol as Table 15.

SFA achieves the highest score in all four categories. Removing count correction lowers overall HPS from 29.596 to 28.428, while separate softmax reduces it to 11.176. The qualitative comparison in Figure 8 illustrates these differences under a common prompt: SFA produces a detailed character illustration, Variant 1 loses definition in the clothing and background, and Variant 2 is dominated by

Table 17: Ablations of count correction, shared softmax, and far-field summaries on FFHQ-128. The first ten data rows report FID_{10k}; the final row reports FID_{50k} at epoch 20. Lower values indicate better generation quality, and bold marks the lowest FID in each row.

Epochs	SFA	Variant 1	Variant 2	Variant 3
2	**40.64**	53.41	61.43	98.73
4	**35.65**	42.47	48.63	76.14
6	**31.74**	38.19	43.23	67.54
8	**29.84**	35.70	39.10	61.84
10	**28.17**	33.01	36.91	58.79
12	**27.24**	31.10	34.69	54.61
14	**26.37**	30.73	34.48	52.69
16	**25.49**	29.12	32.19	50.83
18	**25.04**	28.28	31.36	49.55
20	**24.16**	27.17	29.43	47.08
20 (FID_{50k})	**22.91**	25.91	27.99	45.76

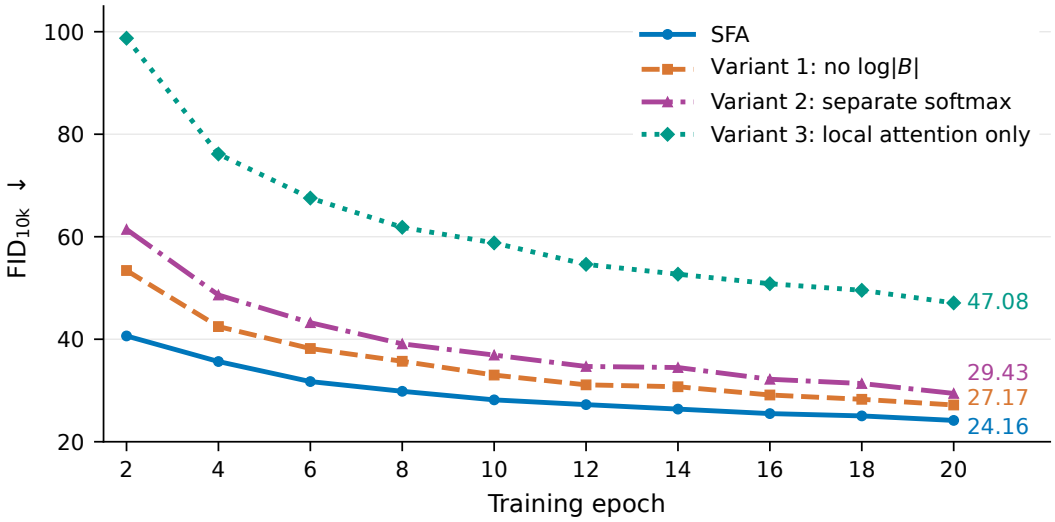


Figure 7: FID_{10k} during the FFHQ-128 stage. Each point evaluates 10,000 generated samples at a two-epoch interval. SFA achieves lower FID than all three ablations at every recorded checkpoint; Variant 3 retains only the exact local neighborhood.

Table 18: HPS v2.1 ablation for direct self-attention replacement in Stable Diffusion 3.5 Medium without fine-tuning. Higher values indicate stronger predicted human preference; bold marks the highest score in each column.

Attention	Anime	Concept art	Paintings	Photo	Overall
SFA	**31.162**	**30.128**	**29.954**	**27.141**	**29.596**
Variant 1	30.144	29.001	28.534	26.032	28.428
Variant 2	11.484	11.401	11.104	10.715	11.176

grid-like artifacts that obscure the subject. These results support the importance of count correction and shared normalization for preserving generation quality when reusing pretrained weights.



(a) SFA (b) Variant 1: no $\log|B|$ (c) Variant 2: separate softmax

Figure 8: Qualitative ablation of direct self-attention replacement in Stable Diffusion 3.5 Medium. All three 1024×1024 images are generated without fine-tuning using the prompt "Male character illustration by Gaston Bussiere." Panels compare (a) SFA, (b) removal of count correction, and (c) separate softmax at each scale.

Learnable aggregation in PixelFlow. Variant 4 introduces aggregation parameters that must be learned, so we evaluate it through training from scratch on ImageNet-256 with PixelFlow. After ten epochs under the setup in Appendix C.4, SFA achieves FID_{50k} of 40.66 and IS of 32.90, compared with 41.40 and 31.50, respectively, for Variant 4.

Interpretation. Variant 1 supports weighting pooled blocks by their original token counts, and Variant 2 shows the benefit of normalizing all scales together. Their Stable Diffusion 3.5 Medium results also connect these choices to pretrained-weight reuse. Variant 3 demonstrates that pooled far-field information improves generation beyond the exact local branch. Finally, the comparison with Variant 4 favors SFA's mean pooling with count correction over learnable aggregation without that correction at the same ten-epoch budget.

Efficiency. Variants 1 and 3 run at approximately the same speed as SFA in our experiments, despite substantially lower generation quality. Variant 2 introduces a separate softmax denominator and output-dependent gradient correction at every scale. Each query therefore requires scale-specific normalization and backward statistics, giving up the shared normalization and gradient correction used by SFA across all interactions. Variant 4 loses two useful properties of mean pooling. Independently learned aggregation weights at different scales generally prevent a parent representative from being constructed from its four existing child summaries, removing the inexpensive recursive construction available to SFA. During backward, spatially varying weights require distinct gradient contributions for individual token positions, replacing SFA's uniform gradient sharing within each base block; learning those weights adds parameter-gradient computations. These structural differences increase aggregation and differentiation work and remove the computational reuse described in Appendix B. Our current implementation of Variant 4 shows a substantial runtime penalty on an RTX 5090 with batch size 4, six heads, head dimension 64, and bfloat16 inputs. For $N = 4096, 16384, 65536, 262144$ tokens, its forward runtimes are $2.64\times$, $2.27\times$, $1.96\times$, and $1.98\times$ those of SFA; its backward runtimes are $4.74\times$, $2.89\times$, $3.10\times$, and $3.14\times$, respectively. This slowdown substantially reduces or even eliminates the speed advantage over other efficient-attention mechanisms. Timings are medians after warmup. Forward is measured without gradient recording and includes aggregation; backward is measured separately on a retained graph and includes gradients through aggregation and its learned weights.

C.9 Qualitative samples

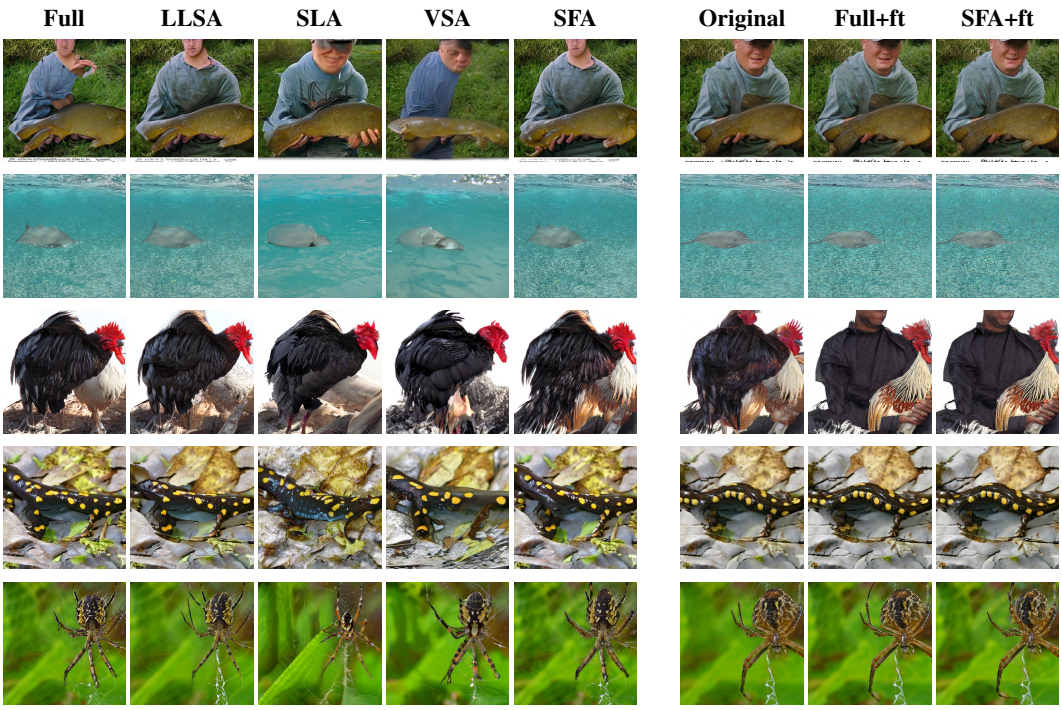
We provide additional generated images below for FFHQ, ImageNet, and Stable Diffusion 3.5 Medium, complementing the examples in Figure 2(E).

FFHQ. Figure 9 presents additional samples from pixel-space face generation on FFHQ at 128×128, 256×256, and 512×512 resolution, extending the examples in Figure 2(E). These samples are generated by a pixel-space DiT-S with SFA in all attention layers. The three columns display the resolutions from highest to lowest, with two, four, and eight samples, respectively.



Figure 9: Pixel-space face samples on FFHQ at 512×512 (left), 256×256 (middle), and 128×128 (right). The columns contain two, four, and eight images, respectively, with displayed side lengths proportional to image resolution.

ImageNet. Figure 10 presents five groups of class-conditional ImageNet samples generated with PixelFlow. The left panel compares Full, LLSA, SLA, VSA, and SFA after 40 epochs of training from scratch; the right panel compares the original pretrained weights with the full-attention and SFA models after fine-tuning. This organization complements the training comparison in Table 1 and the adaptation comparison in Table 11.



(a) From scratch (epoch 40). (b) Fine-tuning.

Figure 10: PixelFlow samples on ImageNet at 256×256 resolution. (a) Five attention methods after 40 epochs of training from scratch. (b) The original pretrained full-attention model (Original), the full-attention model after fine-tuning (Full+ft), and the model fine-tuned with SFA in its highest-resolution stage (SFA+ft). The five sample groups are aligned by row across both panels. In panel (a), SFA produces the most coherent human anatomy in the scene with a person holding a fish (first row). In the ray scene (second row), SFA renders the finest ripple patterns, while SLA and VSA produce less realistic water textures.

Stable Diffusion 3.5 Medium. We present text-to-image samples generated with Stable Diffusion 3.5 Medium. The 1024×1024 example in Figure 2(E) uses the prompt "Black and white portrait of Thabo Mbeki with highly detailed ink lines and a cyberpunk flair, created for the Inktober challenge as part of the Cyberpunk 2020 manual coloring pages".

Figures 11 and 12 compare ten groups of samples before fine-tuning. From left to right, each row shows samples from the original full-attention model (Full), the model with SFA replacing image self-attention (SA), and the model with SFA replacing both image self-attention and image-to-image interactions in joint attention (ALL). The three images in each row share the same prompt and are generated at 1024×1024 resolution. This comparison illustrates the immediate effect of attention replacement under the settings in Appendix C.6.

Figures 13 and 14 extend the comparison to ten further examples after the shared one-epoch fine-tuning schedule, with columns for Full+ft, SA+ft, and ALL+ft. These samples are generated with the raw checkpoints evaluated in Tables 15 and 13, with prompts matched across models and the same 1024×1024 image resolution.

Prompt	Full	SA	ALL
Close-up portrait of a goblin emperor with a bone headdress, depicted in vibrant colors with intricate details.			
An astronaut in white futuristic cybernetic armor running on the surface of the moon, featured in an artwork illustration on Artstation.			
A birch forest in autumn with falling leaves that resemble flying butterflies and dancing elves.			
a guy in the desert sitting on his motorcycle			
A colorful kite is ready for launch on a blue sky day			

Figure 11: Samples from Stable Diffusion 3.5 Medium before fine-tuning. Each row compares Full, SA, and ALL using the same prompt.

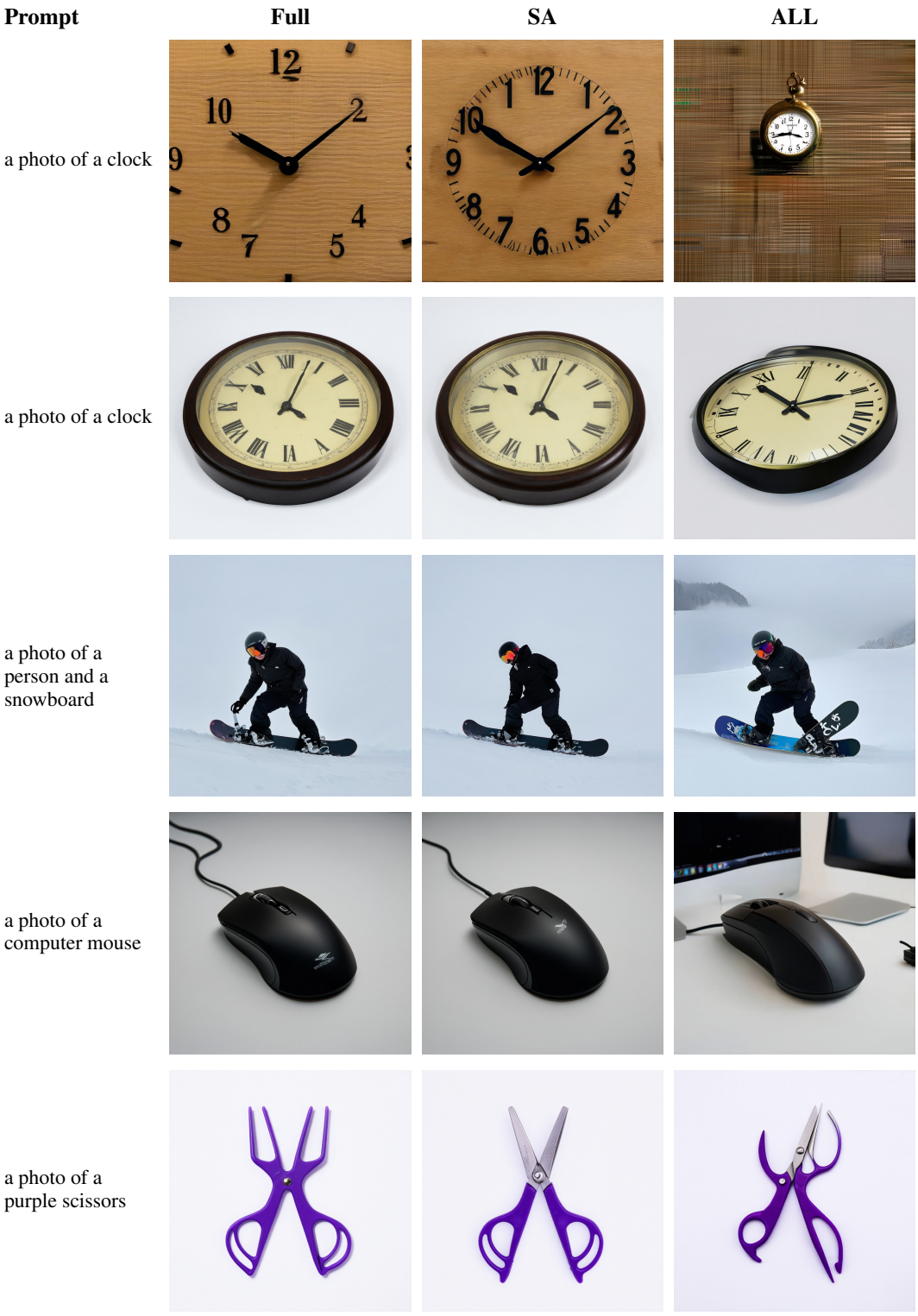


Figure 12: Additional samples from Stable Diffusion 3.5 Medium before fine-tuning. Columns show Full, SA, and ALL, with matching sampling settings within each row. The first two rows are distinct samples from the same clock prompt. SA depicts the clock numerals more accurately and renders the snowboarder's goggle lenses with more realistic colors (third row). It also renders the computer mouse and scissors more realistically than Full (last two rows).

Prompt	Full+ft	SA+ft	ALL+ft
a bunch of people on skiing on a hill			
a couple of men that are next to some boxes			
Swedish lake at night with heavy snowfall depicted in hyper-realistic and detailed art.			
A green landscape by Makoto Shinkai.			
A stylized digital art image of a cherry tree overlooking a valley with a waterfall during sunset.			

Figure 13: Samples from Stable Diffusion 3.5 Medium after one epoch of fine-tuning. Each row compares Full+ft, SA+ft, and ALL+ft using the same prompt. ALL+ft avoids the unnatural overlap between skiers seen in Full+ft (first row) and renders the men handling boxes with more plausible anatomy (second row).

Prompt	Full+ft	SA+ft	ALL+ft
a simple white background highlighting a hand-drawn black circle that encompasses the playful, cursive text ‘infinity makes me happy’. The words are crafted to resemble a quick, personal brush script, giving it a casual and intimate feel. Just outside the circle, faint sketch marks are visible, implying a human touch in the creation of this design.			
A vivid and eclectic depiction of a fisherman’s village drawn by Ralph Steadman, composed of dark velvet hues, contrasting with splashes of bright yellow and teal. In the foreground, a tall coconut tree sways, its silhouette framed against the backdrop of a small, bustling island. A sturdy jetty extends into the water, where several fishing boats rock gently with the waves.			
A chilled, transparent bottle of light beer with condensation beads forming on its surface, placed on a wooden table. A bright yellow slice of lemon is wedged onto the rim of the bottle, adding a citrus accent to the beverage. The bottle label is partially obscured by the moisture, but hints of green and gold are visible on the design.			
A man in a casual gray shirt and faded blue jeans is captured mid-air above a sleek black skateboard, executing a skillful jump. To the side of him lies a large black skateboard ramp with visible scuff marks from frequent use. In the expansive blue sky overhead, a few wispy clouds drift lazily by, adding a sense of height to his aerial feat.			
A striking black bird with glossy feathers sits atop the vibrant orange petals of a Bird of Paradise flower. The unique flower is positioned in the midst of an arid desert landscape, with various cacti and sparse vegetation dotting the sandy ground. In the background, the sun casts a warm glow on the distant rolling dunes.			

Figure 14: Additional samples from Stable Diffusion 3.5 Medium after one epoch of fine-tuning. Each row compares Full+ft, SA+ft, and ALL+ft using the same prompt.

D Extended Related Work

The attention comparisons in Section 2 draw on broader work in computational efficiency, spatial hierarchies, and image statistics.

D.1 Linear and hardware-efficient attention

Attention efficiency depends on both the representation of interactions and their implementation. FlashAttention (Dao et al., 2022) reduces memory traffic through tiled exact attention, and FlashAttention-2 (Dao, 2024) improves work partitioning and parallelism. However, their dense arithmetic remains quadratic in token count. Linear attention addresses this arithmetic cost through kernel feature maps and associative multiplication (Katharopoulos et al., 2020), with Performer approximating the softmax kernel through positive random features (Choromanski et al., 2021). Kernel approximations and memory-efficient implementations therefore address distinct components of attention cost. SFA combines an $O(N \log N)$ attention architecture that delivers strong generative performance with a computationally efficient Triton implementation.

In diffusion models, Sana demonstrates the use of linear attention in a high-resolution diffusion architecture (Xie et al., 2025). Native Sparse Attention (NSA) combines compressed representations, selected blocks, and a local window in language models (Yuan et al., 2025). These approaches allocate computation through a kernel representation or several attention branches; SFA instead assigns spatial resolution through a complete, query-specific partition of the token grid.

D.2 Spatial hierarchies

Spatial hierarchies provide a structured allocation of computation by representing interactions at different resolutions. For example, Swin Transformer (Liu et al., 2021) uses shifted local windows, and H-Transformer-1D (Zhu & Soricut, 2021) constructs a one-dimensional hierarchical matrix approximation. For long-range interactions, Focal attention (Yang et al., 2021) combines fine local tokens and coarse distant learnable summaries in one softmax, retaining overlapping representations of regions across scales. QuadTree attention (Tang et al., 2022) instead makes refinement content dependent, building pooled query and key pyramids and selecting relevant coarse regions before evaluating finer interactions. This content-dependent refinement contrasts with SFA's fixed distance-dependent allocation, and the methods also differ in how they aggregate contributions across levels.

The methods mentioned above and in Section 2 all differ from SFA. SFA introduces no additional parameters and supports both effective reuse of pretrained weights and high-quality generation when trained from scratch. Our comparison with LLSA shows that SFA's fixed spatial hierarchy achieves higher kernel throughput and better generation quality than hierarchical dynamic selection. The four ablations in Section 5.6 and Appendix C.8 further demonstrate the necessity of various components of our approach. Variants 1 and 2 remove count correction and normalize scales separately, respectively; both reduce quality in FFHQ-128 training and in direct self-attention replacement for Stable Diffusion 3.5 Medium without fine-tuning. Variant 3 retains only the exact local neighborhood, causing a substantial loss of FFHQ generation quality and demonstrating the contribution of pooled far-field context. Variant 4 replaces mean pooling with learnable aggregation and removes count correction, yielding worse FID and IS than SFA under the same ten-epoch PixelFlow training budget. Together, these comparisons support the combination of global coverage, mean pooling, count correction, and shared normalization.

These choices also support the kernel efficiencies detailed in Appendix B. Shared softmax reuses normalization and gradient statistics across scales, while mean pooling enables recursive preprocessing and uniform gradient propagation within each base block. Variants 2 and 4 lose these respective forms of computational reuse, with their efficiency consequences analyzed in Appendix C.8. In the FFHQ measurements, SFA reduces attention's share of runtime sufficiently for other transformer components to dominate, yielding nearly constant token throughput at a fixed token batch across resolutions. Further end-to-end acceleration therefore depends on optimizing those components.

D.3 NATURAL-IMAGE STATISTICS AND POSITIONAL FREQUENCIES

The spatial prior underlying SFA has roots in classical studies that connect natural-image statistics to efficient spatial representations (Field, 1987; Simoncelli & Olshausen, 2001) and document approximate scale invariance (Ruderman & Bialek, 1994). In two dimensions, an inverse-square spectrum distributes comparable energy across spatial octaves, motivating attention resolution relative to distance. A related connection between frequency and image structure appears in diffusion models, where Kadkhodaie et al. (2024) relate generalization to denoising in harmonic bases adapted to image geometry. Within attention, positional encoding adds another source of frequency dependence by determining how spatial phase enters the computation. Rotary position embeddings (RoPE) express relative positions through rotations (Su et al., 2024) and have effective two-dimensional constructions for vision transformers (Heo et al., 2024). Studies of trained language models further reveal distinct uses of high and low rotary frequencies (Barbero et al., 2025). These observations are relevant to SFA because it averages keys after positional encoding, so its coarse blocks aggregate both content and positional phase. The resulting connection between spatial averaging and frequency response is analyzed in Appendix A.9.