BEYOND QUADRATIC ATTENTION: SEPARABLE GAUSSIAN OPERATORS FOR ODE VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

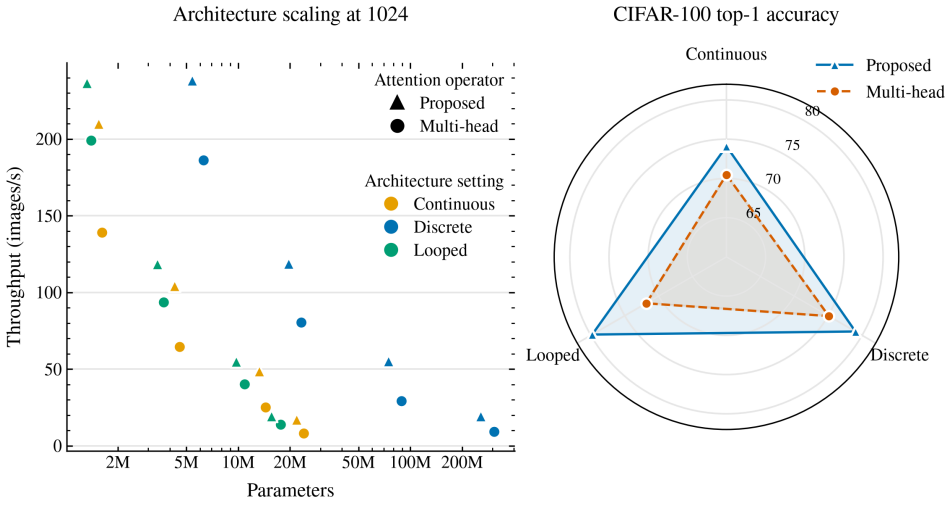


Figure 1: *Left:* throughput–memory trade-off at resolution 1024 as the width of the model increases. Points farther upward provide more throughput and points farther left allocate less memory. Our proposed method dominates multi-head attention, achieving higher throughput at lower memory. *Right:* comparison of the three model families with our proposed method.

ABSTRACT

Self-attention carries a quadratic cost in token count, which limits how far Vision Transformers can scale with input resolution. We introduce SSoG, a separable sum-of-Gaussians spatial operator that replaces dense self-attention with a learned mixture of axis-factorized Gaussian kernels, reducing the per-layer cost from $\mathcal{O}(N^2)$ to subquadratic in N. Unlike dot-product self-attention, whose Lipschitz constant is unbounded outside a compact input domain, SSoG with fixed kernels is globally Lipschitz continuous, which, via the Picard–Lindelöf theorem, guarantees that the continuous-depth variant has a unique solution on the whole integration interval, a global guarantee that dense-attention counterparts lack in general. On CIFAR-100 at 224×224, in a matched discrete-Transformer setting and in the continuous setting, SSoG improves Top-1 accuracy by 2.2–4.1 points across three model widths while using 8.8–16.6% fewer parameters, with depth and training recipe held fixed. In a looped-Transformer setting, SSoG improves Top-1 accuracy by 5.4–8.1 points. We further probe inference-time scaling by evaluating on inputs up to 1024×1024 at matched batch size: in the continuous-depth (Neural ODE) architecture, SSoG reaches up to 154% higher throughput and over 60% lower peak memory than dense attention, with the gap widening as resolution grows, consistent with the difference in asymptotic cost. In the discrete and looped settings, SSoG also achieves over 30% higher throughput than dense attention at this resolution. Together, these results establish SSoG as a practical, parameter-efficient, and theoretically grounded alternative to dense attention across discrete, weight-shared, and continuous-depth Transformer variants.

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.

1 INTRODUCTION

The self-attention mechanism, implemented through scaled dot-product attention (SDPA) (Vaswani et al., 2017), has marked a paradigm shift in AI. ViT (Dosovitskiy et al., 2021) showed that this SDPA mechanism could scale to images, and this led to remarkably impressive results across a wide range of vision tasks. Foundation models such as DINO (Caron et al., 2021), SAM (Kirillov et al., 2023) for segmentation and DiT (Peebles & Xie, 2023) for diffusion, as well as VLMs and LLMs, demonstrated that this attention mechanism could scale to results beyond what seemed possible. Alongside these strengths, however, come several problems: the complexity of self-attention itself, a limited understanding of how a model learns to *attend*, difficulty training these models due to theoretical issues inherent to SDPA, and the emergence of artifacts (Darcet et al., 2024; Qiu et al., 2025), among others. These issues have mostly been studied in isolation. In this paper, we show that reducing the complexity of self-attention and satisfying the theoretical property required for Lipschitz continuity can be achieved by the same operator, and we study the behaviour of this operator when instantiated across discrete, looped, and continuous-depth Transformers.

The complexity of SDPA arises from the token sequence length. For N tokens of dimension d, SDPA needs $\mathcal{O}(N^2)$ memory and $\mathcal{O}(N^2 d)$ computation, and in vision N grows with the square of the image side, so the cost increases with the resolution. Research on context compression (Zhang et al., 2025; Liu et al., 2026) reduces the memory needed to store the matrix, reducing the effective sequence length, but the operation is still quadratic. State Space Models (Gu & Dao, 2024; Zhu et al., 2024) drop pairwise interaction altogether and are linear in N; however, they tend to be harder to train and imply a complete change of architecture. Looped Transformers (Byra et al., 2026) and Neural ODEs (Zhong et al., 2022) work on a different axis: they share parameters across depth, which reduces model size but not the cost of a single evaluation. Moreover, these models are sensitive to hyperparameters, and noise in the parameter space is carried through all the optimization steps and cannot be isolated.

Beyond the cost of recurrent and continuous models, they inherit a second, more fundamental problem from dense attention itself: attention scores are computed over an unbounded domain. Since the query-key interaction is an inner product between unconstrained vectors, SDPA is not globally Lipschitz continuous (Kim et al., 2021). This is part of why the literature argues that Transformers are sensitive to hyperparameters, initialization, and learning rate (Qi et al., 2023). These theoretical implications matter most for continuous-depth models. Dense attention is smooth, and hence Lipschitz on bounded sets, so the Picard–Lindelöf theorem still guarantees a unique solution locally in time. What global Lipschitz continuity adds is existence of the solution on the whole integration interval for every initial condition, without finite-time blow-up, together with a uniform bound on how much two trajectories can drift apart when their initial conditions differ. Recurrent and continuous architectures, which apply the same block many times, are therefore the ones that benefit most from a property that dense attention lacks.

Pisoni (2026) recently outlined SSoG, a *Sum of Separable Gaussians* operator that replaces dot-product attention in standard ViTs with a small mixture of anisotropic, axis-factorized Gaussian kernels, in a *fixed* and a *content-steered* variant, and reported preliminary results on image classification. Because the kernel factorizes along the image axes, SSoG never builds the $N \times N$ attention matrix and is subquadratic in N. We study it in a regime for which it was neither designed nor analyzed: as the token mixer of looped and continuous-depth Transformers. In this paper, we show that SSoG with fixed kernels is globally Lipschitz continuous by construction, precisely the property that looped and continuous-depth models require, while its content-steered variant is Lipschitz on sets of bounded token norms; we are the first to study it inside these architectures. Unlike MHA, where tokens compete for a fixed normalization budget through content-dependent scores, fixed SSoG assigns weights from a spatial kernel of the relative offsets, so no token can take mass away from another, leaving no mechanism for an attention sink to form. We confirm this empirically on the trained models: dense attention develops a persistent sink on ImageNet, whereas SSoG does not (Appendix A.2). We benchmark this instantiation of SSoG against dense attention across discrete, looped, and continuous-depth settings, providing the first empirical and theoretical characterization of its behavior as a weight-shared, continuous-depth attention mechanism. We also find that content steering improves the discrete and looped models but not the continuous-depth one, and that the efficiency gains of SSoG grow with input resolution.

Contributions.

- We prove that fixed SSoG is globally Lipschitz by construction, which makes continuous-depth models built on it well posed on the whole integration interval, and that the content-steered variant is Lipschitz on bounded sets (Appendix A.6).
- We use SSoG, in fixed and content-steered variants, as a drop-in replacement for dense attention in discrete, looped and continuous-depth vision Transformers, and benchmark it extensively against dense attention in each setting.
- We study the accuracy-compute trade-off of ODE-SSoG in integration depth and number of atoms, and show its sub-quadratic operator gains over dense attention in throughput and memory as resolution grows.

2 METHOD

We follow the formulation of Pisoni (2026) and restate it with the notation used in our analysis. In contrast to conventional self-attention, SSoG does not construct query-key similarities or explicitly materialize an $N \times N$ attention matrix. Instead, each head mixes value features through a small collection of learned anisotropic Gaussian spatial kernels. These kernels factorize along the x and y axes, so two-dimensional mixing reduces to successive one-dimensional contractions.

Each Gaussian atom, r, provides a compact geometric description of a spatial interaction pattern. Its center determines *where* information is gathered from, its spread determines *how local or global* the interaction is, and its mixture weight determines the contribution of that pattern to the head output. Let $X \in \mathbb{R}^{B\times N\times D}$ denote the sequence of input tokens, where the $N-1$ tokens are patch tokens and one is a class token and let H be the number of heads with head dimension $D^h = D/H$. Unlike dot-product self-attention, SSoG contains no query or key projections. For each head $h \in \{1, \dots, H\}$, a learned projection $W_V^h \in \mathbb{R}^{D\times D^h}$ produces:

$$V^h = XW_V^h, \tag{1}$$

Stacking all heads gives $V \in \mathbb{R}^{B\times N\times H\times D^h}$. The class token is separated before spatial filtering, while the remaining $N-1 = n_y n_x$ patch tokens are reshaped onto a $n_y \times n_x$ spatial grid.

Gaussian spatial kernels. Each head h contains R Gaussian atoms. Atom r is parameterized by a center $\mu_r = (\mu_{r,x}, \mu_{r,y})$, an anisotropic spread $\sigma_r = (\sigma_{r,x}, \sigma_{r,y})$, and a mixture logit ℓ_r, from which the mixture weights λ are obtained by softmax over atoms. For axis $a \in \{x, y\}$ and offset $\delta_a = q_a - s_a$, the unnormalized Gaussian log-kernel is:

$$\log G^a(\delta_a) = -\tfrac{1}{2}\log(2\pi) - \log\sigma_a - \frac{(\delta_a - \mu_a)^2}{2\sigma_a^2}. \tag{2}$$

The kernel is normalized over source coordinates using a learned positive temperature τ,

$$A_a(\delta) = \mathrm{softmax}\left(\frac{\log G^a(\delta)}{\tau}\right). \tag{3}$$

To ensure valid, non-degenerate kernels, σ_a and τ are obtained from unconstrained learned counterparts $\tilde{\sigma}_a$ and $\tilde{\tau}$ via a positive reparameterization, with $\sigma_{\min} > 0$ bounding each axis-wise spread from below:

$$\sigma_a = \mathrm{softplus}(\tilde{\sigma}_a) + \sigma_{\min} + \varepsilon \tag{4}$$

$$\tau = \mathrm{softplus}(\tilde{\tau}) + \tau_0, \tag{5}$$

$$\lambda = \mathrm{softmax}_r(\ell) \tag{6}$$

Here $\tau_0 > 0$ and $\varepsilon > 0$ prevent over-sharpening of the axis-wise softmax and ensure numerical stability of the width constraint, respectively. Under the offset convention $\delta = q - s$, a center μ selects source positions near $s = q - \mu$, so the learned centers directly encode the relative spatial locations from which each head gathers information.

We study two variants of this operator: *fixed SSoG* and *content-steered*, described below. In *fixed SSoG* the geometric parameters, centers, spreads, and mixture weights, are shared across all images and query positions. In the *content-steered* variant, termed *LookAt*, bounded input-dependent perturbations adapt the geometry per query patch while preserving the separable structure.

2.1 FIXED SSOG

In fixed SSoG, the atom geometry and mixture weights are shared across images and query positions. For a fixed head h, the value vectors are collected into $V \in \mathbb{R}^{n_y \times n_x \times D^h}$. For each atom r, the axis attention matrices defined entry-wise by equation 3 are $A_{r,y} \in \mathbb{R}^{n_y \times n_y}$ and $A_{r,x} \in \mathbb{R}^{n_x \times n_x}$. The head output is:

$$U_r = A_{r,y}\, V, \tag{7}$$

$$Z = \sum_{r=1}^{R} \lambda_r\, U_r\, A_{r,x}^{\top}. \tag{8}$$

This separable formulation reduces the asymptotic cost of spatial mixing from quadratic to subquadratic in the number of tokens: for a square $n \times n$ token grid with $N-1 = n^2$ and a small, fixed number of atoms R, applying R separable Gaussian atoms through successive horizontal and vertical contractions costs $\mathcal{O}(RN^{3/2}D)$, compared with $\mathcal{O}(N^2D)$ for dense self-attention; Appendix A.1 gives the full derivation, including storage costs and the effect of query-dependent steering.

2.2 CONTENT-STEERED SSOG

Fixed SSoG applies the same learned spatial geometry independently of image content. To introduce content dependence without reintroducing pairwise query-key interactions, we additionally consider the *LookAt* variant. For each query position q, with token $x_q \in \mathbb{R}^D$ (the q-th row of X), the model predicts bounded modifications to the Gaussian centers, spreads, and mixture weights. Three linear predictors generate the following query-dependent perturbations for each head h independently:

$$\mu_{r,a}(x_q) = \mu_{r,a} + \alpha_\mu M \tanh\big([W_\mu x_q + b_\mu]_{r,a}\big), \tag{9}$$

$$\sigma_{r,a}(x_q) = \sigma_{r,a} \exp\Big[\alpha_\sigma \tanh\big([W_\sigma x_q + b_\sigma]_{r,a}\big)\Big], \tag{10}$$

$$\lambda_r(x_q) = \mathrm{softmax}_r\Big[\ell_r + \alpha_\lambda \tanh\big([W_\lambda x_q + b_\lambda]_r\big)\Big]. \tag{11}$$

where W_z and b_z are learned weights and biases, $[v]_{r,a}$ denotes the (r,a)-th entry of v (indexed over atoms and axes), M is the maximum center displacement, and

$$\alpha_z = \mathrm{softplus}(\tilde{\alpha}_z) + \varepsilon, \qquad z \in \{\mu, \sigma, \lambda\}. \tag{12}$$

The $\tanh$ nonlinearities bound the predicted modifications, while the exponential parameterization ensures that the steered spreads remain positive. Query-dependent steering does not remove the separable factorization; the computation continues to use successive one-dimensional contractions. Individual steering paths can be disabled independently for ablation.

The scalar steering parameters are initialized close to zero; the specific values used in each training regime are given in §3. The model therefore starts close to the fixed SSoG geometry and progressively activates content-dependent modulation during training. This initialization is particularly useful in the continuous-depth setting, where the operator is applied repeatedly.

2.3 INSTANTIATING SSOG ACROSS ARCHITECTURES

SSoG is designed as a drop-in replacement for MHA: in every architecture we study, the only change relative to the corresponding baseline is replacing its attention block with SSoG, while the surrounding architecture is left unchanged. We instantiate this substitution in three settings: a standard discrete (non-recurrent) ViT, a looped/recurrent Transformer based on Byra et al. (2026) that shares its attention block across depth, and a continuous-depth architecture in which SSoG defines the vector field of a Neural ODE (Boned Riera et al., 2025).

All three settings share the same tokenization. Let an input image be $I \in \mathbb{R}^{B\times C\times H_0\times W_0}$, where B is the batch size. A convolution with kernel size and stride P produces a grid of $n_y \times n_x$ patch tokens, where $n_y = H_0/P$ and $n_x = W_0/P$. We denote the total sequence length by $N = n_y n_x + 1$ and the embedding dimension by D. With a class token prepended, the initial sequence is

$$X_0 = \mathrm{PatchEmbed}(I) + E_{\mathrm{pos}} \tag{13}$$

Class-token handling. The class token has no natural two-dimensional coordinate and is therefore excluded from Gaussian spatial filtering. Instead, it receives a lightweight global aggregation of the projected patch values $v_p \in \mathbb{R}^{D^h}$, $p = 1, \ldots, N-1$:

$$v_{\mathrm{cls}} = \frac{1}{N-1}\sum_{p=1}^{N-1} v_p. \tag{14}$$

This aggregated class-token representation is concatenated with the filtered patch representations and projected through $W_O \in \mathbb{R}^{D\times D}$, producing the SSoG block output. Patch-to-patch mixing therefore remains entirely Gaussian, while no dense class-to-patch attention scores are introduced. Here v_{cls} denotes the per-block value-space aggregate of the class token; in all three settings below, the class-token state accumulated across blocks is denoted z_{cls}.

Discrete ViT and looped Transformer. In the standard discrete setting, L SSoG blocks are stacked, each with its own independent parameters $\theta_1, \ldots, \theta_L$. In the looped setting, a single block is used instead, with parameters θ shared across all L iterations (L is the depth); both follow the recurrence

$$X_l = X_{l-1} + \mathrm{SSoG}(X_{l-1}|\theta_l), \qquad l = 1, \ldots, L, \tag{15}$$

with θ_l independent across l in the discrete case, and $\theta_l = \theta$ for all l in the looped case.

ODE-SSoG (continuous-depth). Rather than applying SSoG_θ a fixed number of times, ODE-SSoG treats depth as continuous: SSoG serves as the token-mixing operator inside the vector field $F(\cdot,\cdot|\theta)$ of the Continuous ViT, and the token sequence is evolved by the initial-value problem

$$\frac{dX(t)}{dt} = F(t, X(t)|\theta)\,, \qquad X(0) = X_0, \qquad t \in [0,1], \tag{16}$$

integrated numerically from $t = 0$ to $t = 1$. As in the looped setting, the same parameters θ are reused at every function evaluation; ODE-SSoG can be seen as the continuum limit of the looped variant, where the number of iterations define the horizon in the integration path.

In all three settings, the final class-token state z_{cls}, obtained after L iterations of equation 15 or by integrating equation 16 to $t = 1$, is normalized and linearly projected to the class logits:

$$\widehat{y} = W_{\mathrm{cls}}\mathrm{LN}(z_{\mathrm{cls}}) + b_{\mathrm{cls}}. \tag{17}$$

3 Experiments and Results

In this section, we study different questions, each addressed in its own subsection. The Appendix A.3 reports further supporting ablations that isolate individual design choices without bearing directly on these questions. Unless otherwise stated, classification ablations are conducted on CIFAR-100 for 500 epochs, following the same training recipe as Touvron et al. (2021) and patch size 16. For the continuous models, explicit Euler integration is used; NFE denotes the number of vector field evaluations.

3.1 Is SSoG sensitive to different hyperparameters?

Figure 2 summarizes the main comparison among models and capacity levels. In all cases the addition of the SSoG improves the results in all the families of models and in the capacity of each model within the family by at least 2.22 Top-1 points (full results in Table 10).

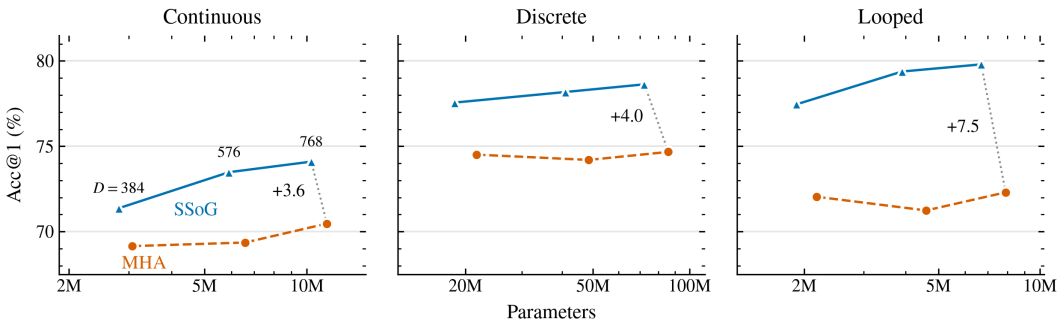


Figure 2: Width scaling on CIFAR-100, dense attention (MHA) versus SSoG, across the three architecture settings. The dotted connector and label give the Acc@1 gain of SSoG at $D = 768$.

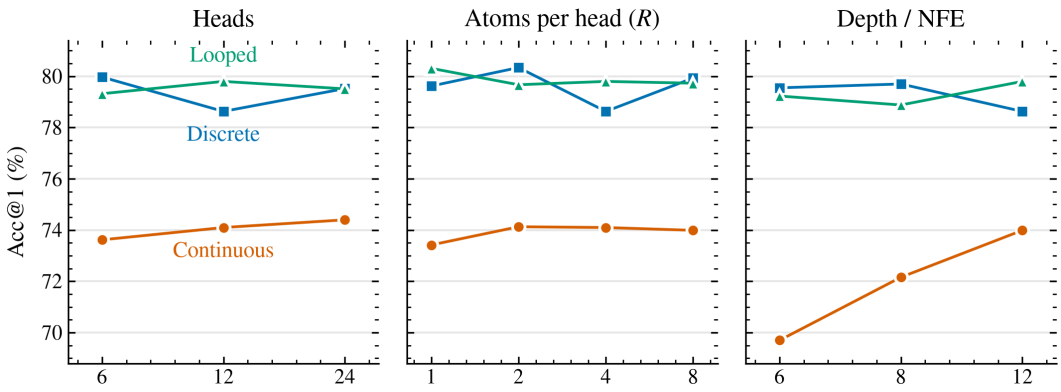


Figure 3: Ablations of SSoG on CIFAR-100 in the three architecture settings.

We report ablations over head count, depth, and atom count. Figure 3 shows the behavior of SSoG under different hyperparameters, with the underlying results reported in Appendix A.3 Table 11. The table shows that, at a fixed embedding dimension of 768 and four atoms per head, increasing the number of heads from 6 to 24 improves top-1 accuracy slightly in the Continuous model while there is almost no effect in the looped and discrete setting. The atom-count sweep saturates rapidly: moving from one to two atoms improves top-1 accuracy from 73.42% to 74.13%, whereas eight atoms reach only 73.99%. To assess whether this saturation reflects redundancy among the learned atoms, we computed the entropy of their mixture weights: on CIFAR-100, the eight-atom model has an entropy-derived effective atom count of only 3.381. Appendix A.4 gives the formulation of the entropy used to compute the effective atoms, and Appendix A.3.1 additionally reports the effect of the sigma floor $\sigma_{\min}$, the remaining capacity constraint on the Gaussian atoms. Two to four atoms per head therefore appear sufficient, while additional atoms are redundant.

We ran the same ablation without SSoG, using dense attention instead; the results are reported in Appendix A.3 Table 12, where the variation across hyperparameters is larger than for SSoG.

3.2 Steering: What Should Be Steered?

We isolated query-dependent changes to the atom center μ, width σ, and mixture weight λ in Table 1. We used full cold steering initialization with 12 NFE as baseline. More runs can be found in the Appendix A.3 Table 7. We focus on the ODE as it yielded the worst results. Width-only steering matches the fixed model in accuracy (74.01% versus 74.22%). Mixture-only steering is also competitive at 73.87%. In contrast, jointly steering centers and widths reduces accuracy to 71.21%, close to full steering at 70.89%. Since center-only and width-only models reach 73.33% and 74.01%, respectively, we conjecture that the main failure mode is their joint query-dependent geometry from the division in equation 2. Appendix A.3.3 reports an ablation showing that a warmer initialization or a larger initialization of the gate $\tilde{\alpha}_\mu$ both cost additional accuracy on top of this in-

Table 1: Steering-component attribution on CIFAR-100 at NFE 12 and $R = 4$, cold initialization throughout. Checkmarks denote query-dependent components.

Variant	μ	σ	λ	Acc@1	Acc@3	Acc@5	Loss	Time (h)
Fixed	-	-	-	74.22	88.59	92.38	1.12	**47.77**
Full, cold	✓	✓	✓	70.89	86.89	91.30	1.20	59.48
Center only	✓	-	-	73.33	87.79	91.61	1.15	58.37
Width only	-	✓	-	**74.01**	**88.39**	**92.16**	**1.11**	57.42
Mixture only	-	-	✓	73.87	88.20	91.73	1.13	57.08
Center + width	✓	✓	-	71.21	86.83	91.13	1.19	58.61

teraction, and Appendix A.3.4 shows that constraining the maximum steering offset partially, but not fully, recovers it.Appendix A.3 Table 7 reports results of steering for the other families, where there is an improvement over the fixed SSoG.

3.3 Does SSoG outperform optimized SDPA kernels as batch size and resolution grow?

We benchmark randomly initialized models in FP16 on an NVIDIA A40 using ten warm-ups and 50 measured iterations. Figure 4 reports a stress test of measured throughput, FLOPs and peak allocated CUDA memory at different resolutions averaged over batch sizes from 8 to 512. Full throughput, memory and FLOPs results are given in Appendix A.1 (Figure 5, Table 4).

On the one hand, in the continuous-depth setting, the throughput of SSoG already exceeds ODE-ViT at every measured resolution, growing from $+11.0\%$ at 224×224 to $+154.5\%$ at 1024×1024. The discrete setting instead starts below parity: SSoG trails ViT-B/16 by up to 22.6% at resolution 224, reaches approximate parity around resolution 512 to 640, and only clearly overtakes it from resolution 640 onward, reaching $+30.1\%$ at 1024. The looped setting sits between the two, with a positive throughput delta at every resolution, from a marginal $+2.9\%$ at 224 to $+33.1\%$ at 1024.

On the other hand, on memory, SSoG uses less memory than ODE-ViT, saving between 27% and over 60%, and less memory than the standard ViT, saving between approximately 6% and 10%, whereas the looped setting is the only one where SSoG costs more than dense attention. A plausible explanation is that the discrete and looped baselines use a highly optimized fused attention kernel with strong absolute memory efficiency, while the separable-Gaussian implementation used by SSoG relies on standard PyTorch ops; at high resolution, this constant-factor gap can outweigh the structural advantage of SSoG, an effect that appears earliest for the looped setting, whose dense baseline also carries less fixed parameter memory due to weight sharing. Appendix A.1.1 and Table 3 report the computation cost of the attention operation in isolation.

An independent sweep over Tiny, Small, Base and Large ViTs (Figure 1, left) shows that SSoG is faster and lighter than MHA at every width.

3.4 Does SSoG generalize beyond scaling?

Finally, Table 2 reports how the operator scales to a larger benchmark, ImageNet-1k in this case. Each model family was trained under the same experimental setup on the same hardware. SSoG matches or exceeds its dense counterpart in the continuous-depth and discrete settings, by a narrow $+0.30$-point margin for continuous-depth and a clearer $+1.42$-point margin for discrete; the looped setting is the only exception, where dense leads SSoG by 0.35 points on top-1 accuracy. The accuracy of SSoG is competitive with dense attention across all three settings rather than specific to any one of them. Appendix A.5 additionally reports results on linear probing and full fine-tuning, as well as the effect of the data augmentation in the transfer setting.

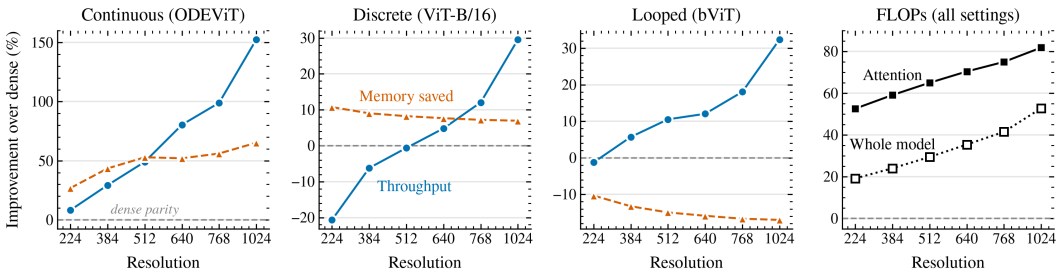


Figure 4: SSoG versus dense attention, throughput and memory improvement (%) by resolution, for each of the three architecture settings (continuous-depth ODE-ViT, discrete ViT-B/16, looped bViT). Each point is the mean relative difference over batch sizes $8 \to 512$ at that resolution. Memory is negated (plotted as % memory *saved*) so both lines share one direction: above the dashed zero line means SSoG wins that metric at that resolution, below means dense wins. The rightmost panel shows the relative improvement in FLOPs for the whole model and for the attention operation alone.

Table 2: ImageNet-1k accuracy at $D = 768$: SSoG versus dense attention (MHA).

	Continuous		Discrete		Looped	
Metric	MHA	SSoG	MHA	SSoG	MHA	SSoG
Acc@1	63.04	**64.02**	76.47	**77.89**	**73.72**	73.37
Acc@3	79.04	**79.62**	89.44	**90.59**	**88.35**	87.93
Acc@5	83.40	**84.45**	92.31	**93.37**	**91.90**	91.65

4 Related Work & Discussion

Approximating attention versus replacing it. Most work on subquadratic attention approximates the softmax kernel instead of abandoning it, using low-rank projections, random features or kernel feature maps (Wang et al., 2020; Choromanski et al., 2021; Katharopoulos et al., 2020). What ties these together is the idea that attention acts as a kernel smoother (Tsai et al., 2019), which vision models have since adopted in several forms (Han et al., 2023; Cai et al., 2023). SSoG goes the other way: it fixes the form of the kernel in advance and learns only the geometry, so the saving comes from structure and not from an approximation. Computationally the closest relatives are separable large-kernel convolutions (Guo et al., 2023; Lau et al., 2024), which split a two-dimensional depthwise kernel into cascaded one-dimensional ones for the same reason we do. What we do differently is constrain each axis kernel to a mixture of R Gaussians instead of learning a free weight at every tap, and that constraint turns out to cost little, since accuracy can be saturated at few atoms per head: for instance, the continuous model with eight atoms uses an effective number of only 3.381.

Gaussians as a bias, and Gaussians as the operator. Gaussians appear often as a spatial prior for attention. The attention maps of a trained ViT tend towards a Gaussian shape (Kim et al., 2023), and several models add a learned Gaussian term to the attention logits (Hajimiri et al., 2026), which keeps the $N \times N$ matrix and makes the Gaussian a bias on top of attention rather than the operator itself. Closer to how we parameterize things, Kundu et al. (2026) drop the query and key projections in favour of an RBF kernel with one bandwidth per head but still compare every pair of tokens, and concurrently Huang & Raza (2026) route tokens through Gaussian mixture components in a latent feature space for language. The operator we study is that of Pisoni (2026), who proposed SSoG for standard ViTs; our contribution lies in its theoretical analysis and in its use as a weight-shared and continuous-depth token mixer. We place the Gaussians over the grid of tokens instead, which is what makes the factorization along axes possible, but it also limits how much content dependence the operator can take on. In the continuous-depth model, steering the width alone matches the fixed model (74.01% against 74.22%) and steering the centers alone costs 0.77 points, while steering both drops to 71.21%; equation 2 suggests why: the center and the spread appear together as the ratio $(\delta_a - \mu_a)^2/2\sigma_a^2$, so when both depend on the query the numerator and the denominator can drift at the same time. We therefore recommend the fixed operator or a limited steering for this model, whereas full steering improves the discrete and looped models (Table 7).

Lipschitz continuity, weight sharing and continuous depth. Kim et al. (2021) showed that dot-product attention is not globally Lipschitz, and the work that followed recovers the property by constraining the operator after the fact, either by substituting Lipschitz counterparts for the unstable components (Qi et al., 2023) or by normalizing the layer and bounding its spectral norm (Dasoulas et al., 2021; Yudin et al., 2025). In fixed SSoG it comes out of the parameterization itself, because the mixing weights depend only on bounded geometric parameters and not on inner products between unconstrained token vectors, so nothing has to be added to the loss to obtain it (Proposition 1); the content-steered variant retains this property only on bounded sets (Proposition 2). This matters most when one block is applied many times, which covers both depth recurrence, from the Universal Transformer (Dehghani et al., 2019) and ALBERT (Lan et al., 2020) to the fully shared single-block ViT of Byra et al. (2026), and continuous depth, from Chen et al. (2018) to recent image classifiers (Zhong et al., 2022; Sander et al., 2022; Boned Riera et al., 2025; Belhasin et al., 2026). This second line has mostly used convolutional vector fields, which, unlike attention, are globally Lipschitz and thus satisfy the Picard–Lindelöf theorem on the whole interval.

We hypothesize that Lipschitz continuity contributes to the accuracy gains: our three settings differ in how much the property should matter, and the results are consistent with that order. The looped model applies an unconstrained dense block repeatedly and gains the most, 7.02 points on average and 79.80% against 72.30% at $D = 768$. The discrete model never repeats a block and gains 3.66 points. ODE-ViT also repeats one block, but it already relaxes the requirement to a local bound, since the JaSMin loss of Boned Riera et al. (2025) penalizes the eigenvalue distribution of the attention map and constrains the spectral radius, which may explain why its gain of 3.32 points is closer to that of the discrete setting than to that of the looped one. The regularizer also has a computational cost. Regularizing the spectrum of the attention map requires materializing that matrix during training, which rules out a fused kernel and keeps the full $N \times N$ map in the computational graph at every function evaluation, so the memory and latency overhead grows with the NFE. What ODE-ViT obtains with a regularizer whose cost scales with the matrix it regularizes, fixed SSoG obtains from the form of the operator at no additional cost. Our efficiency benchmarks (§3.3) are measured at inference, where the regularizer is not evaluated; there, the advantage over ODE-ViT reflects the asymptotic cost of the operator together with the fact that the ODE-ViT baseline does not use a fused attention kernel (Appendix A.1.1).

Limitations. The margins narrow on ImageNet, where SSoG is on par with its counterparts while requiring less compute and achieving higher throughput. Part of this is expected, since a fixed spatial prior is an inductive bias and inductive biases are worth less when data is abundant. It also bounds what the Lipschitz argument can claim, because a benefit that shrinks with dataset size behaves like regularization rather than like well-posedness. We compare only against dense attention, while linear-attention variants and state space models (Gu & Dao, 2024; Zhu et al., 2024) scale subquadratically as well and are the natural competitors in the regime where SSoG wins. SSoG is also $\mathcal{O}(N^{3/2})$ rather than linear (Appendix A.1), so a state space model keeps an asymptotic advantage, whereas SSoG drops into an existing model without changing the rest of the architecture or the training recipe. Finally, we evaluate classification only, leaving open other tasks that could benefit from this operator.

5 Conclusions

We presented the first systematic study of SSoG, a separable sum-of-Gaussians alternative to dense self-attention recently outlined by Pisoni (2026), as the token mixer of discrete, looped and continuous-depth Transformers. We proved that its fixed form is globally Lipschitz continuous, which guarantees a unique solution of the continuous-depth model on the whole integration interval, whereas the content-steered form is Lipschitz only on bounded sets; in both forms the cost is subquadratic in the number of tokens. On CIFAR-100, SSoG outperformed dense attention in all three settings with fewer parameters, most markedly in the looped one; on ImageNet-1k it matched or exceeded dense attention in the discrete and continuous-depth settings and trailed it slightly in the looped one. Its throughput advantage widened with resolution, and it used less memory than dense attention in the continuous-depth and discrete settings but not in the looped one, where the dense baseline benefits from a fused kernel. Future work includes a fused implementation of SSoG, tasks beyond classification, and a comparison with linear attention and state space models.

AI USE STATEMENT

In this work, we used generative AI tools, namely an LLM-based coding and writing assistant (Claude, Anthropic), to aid and polish writing, to support research ideation and execution, and to assist with mathematical claims, as detailed below.

Writing. The assistant polished the grammar and concision of author-written paragraphs, turned results and specifications provided by the authors into LaTeX tables, figure captions and descriptive paragraphs. The authors revised all such text.

Research ideation and execution. The LLM wrote and debugged experiment code: training and evaluation configurations and cluster launch scripts for the ablation and transfer experiments; the throughput and peak-memory benchmark harness; the operator-level attention memory benchmark; the patch-size robustness evaluation; scripts that collect metrics from experiment logs into tables; and the plotting scripts for the figures. While auditing benchmark results, it identified a cross-model measurement interference that led us to re-run every efficiency benchmark with each model in its own process, and all reported efficiency numbers come from these isolated runs. It also proposed an analysis that we adopted: isolating the memory scaling of the attention operator from the rest of the network. The research questions, the method, the choice of experiments to report, the runs and the interpretation of results are the authors'.

Mathematical claims. We used AI to help validate the derivations of the Lipschitz continuity results, providing intuition and suggesting directions that supported the final lemmas and propositions. The LLM also reviewed the notation and its consistency throughout the paper.

Verification. We have reviewed all AI-assisted work. AI-generated code was inspected and tested before any result was reported; for example, models re-loaded by the evaluation code were checked to reproduce their recorded validation accuracy at the training patch size, parameter counts were checked against the saved checkpoints, and efficiency benchmarks were cross-checked by re-running them in isolation. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We release the complete source code as anonymized supplementary material. Every experiment is specified by a modular, hierarchical Hydra configuration: a base file per dataset and architecture setting (continuous, discrete and looped; SSoG and dense-attention baselines) fixes the full training recipe, and each experiment is a short file that inherits from it and overrides only the varied quantity (width, heads, atoms, depth or NFE, or augmentation), so every reported run corresponds to exactly one versioned configuration file. Each model family has its own entry point on a shared PyTorch Lightning training pipeline, and we provide the cluster launch scripts used for every run. Training is seeded through Lightning's `seed_everything` (including data-loader workers) and run with `deterministic=True`; the default seed is 42. Architectures and the training recipes are described in section 2 and section 3, and the per-ablation settings in Appendix A.3–A.5. Efficiency numbers come from a separate benchmark configuration and script, run on an NVIDIA A40 under `torch.inference_mode()`, with each model benchmarked in its own process (section 3.3 and Appendix A.1.1). We use only the public CIFAR-100 and ImageNet-1k datasets with their standard splits; all preprocessing and augmentation is defined in the released configurations and follows the DeiT recipe (Touvron et al., 2021). The table-generation and plotting scripts that produce every table and figure from the logged metrics are also included.

REFERENCES

Omer Belhasin, Shelly Golan, Ran El-Yaniv, and Michael Elad. Advancing image classification with discrete diffusion classification modeling. In *Conference on Computer Vision and Pattern Recognition 2026*, 2026. URL https://openreview.net/forum?id=qxQcVusjI2.

Carlos Boned Riera, David Romero Sanchez, and Oriol Ramos Terrades. Ode-vit: Plug play attention layer from the generalization of the vit as an ordinary differential equation, 2025. URL https://arxiv.org/abs/2511.16501.

Michal Byra, Pawel Olszowiec, Grzegorz Stefanski, Grzegorz Gruszczynski, and Alberto Presta. bvit: Investigating single-block recurrence in vision transformers for image recognition, 2026. URL https://arxiv.org/abs/2605.10661.

Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17302–17313, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, October 2021.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *(NeurIPS)*, 2018. URL https://api.semanticscholar.org/CorpusID:49310446.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking Attention with Performers. In *International Conference on Learning Representations*, 2021. URL https://mlanthology.org/iclr/2021/choromanski2021iclr-rethinking/.

Earl A. Coddington and Norman Levinson. *Theory of Ordinary Differential Equations*. McGraw-Hill, New York, 1955.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *(ICLR)*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

George Dasoulas, Kevin Scaman, and Aladin Virmaux. Lipschitz normalization for self-attention layers with application to graph neural networks. In *International Conference on Machine Learning*, volume 139, pp. 2456–2466, 2021. URL https://proceedings.mlr.press/v139/dasoulas21a.html.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *(ICLR)*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *Computational Visual Media*, 9(4):733–752, 2023. doi: 10.1007/s41095-023-0364-2.

Sina Hajimiri, Farzad Beizaee, Fereshteh Shakeri, Christian Desrosiers, Ismail Ben Ayed, and Jose Dolz. Locality-Attending Vision Transformer. In *International Conference on Learning Representations*, 2026. URL https://mlanthology.org/iclr/2026/hajimiri2026iclr-localityattending/.

Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Yongchao Huang and Hassan Raza. Gaussian mixture attention: Linear-time sequence mixing via probabilistic latent routing, 2026. URL https://arxiv.org/abs/2606.18283.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, volume 119, pp. 5156–5165, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Bum Jun Kim, Hyeyeon Choi, Hyeonah Jang, and Sang Woo Kim. Understanding gaussian attention bias of vision transformers using effective receptive fields, 2023. URL https://arxiv.org/abs/2305.04722.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5562–5571. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kim21i.html.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. URL https://arxiv.org/abs/2304.02643.

Debarshi Kundu, Archisman Ghosh, Swaroop Ghosh, and Vasant Honavar. Projection-free transformers via gaussian kernel attention, 2026. URL https://arxiv.org/abs/2605.02144.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

Kin Wai Lau, Lai-Man Po, and Yasar Abbas Ur Rehman. Large separable kernel attention: Rethinking the large kernel attention design in cnn. *Expert Systems with Applications*, 236:121352, 2024. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2023.121352. URL https://www.sciencedirect.com/science/article/pii/S0957417423018547.

Xin Liu, Runsong Zhao, Pengcheng Huang, Xinyu Liu, Junyi Xiao, Chunyang Xiao, Tong Xiao, Shengxiang Gao, Zhengtao Yu, and Jingbo Zhu. Autoencoding-free context compression for LLMs via contextual semantic anchors. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=8Pi6Du0n7F.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Raphael Pisoni. A few gaussians is all you need: Ssog-attention that steers instead of scores, aug 2026. URL https://pisoni.ai/posts/ssog/. Code available at: https://github.com/4rtemi5/ssog.

Xianbiao Qi, Jianan Wang, Yihao Chen, Yukai Shi, and Lei Zhang. Lipsformer: Introducing lipschitz continuity to vision transformers. In *(ICLR)*, 2023. URL https://openreview.net/forum?id=cHf1DcCwcH3.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated Attention for Large Language Models: Non-Linearity, Sparsity, and Attention-Sink-Free. In *Advances in Neural Information Processing Systems*, 2025. URL https://mlanthology.org/neurips/2025/qiu2025neurips-gated/.

Michael Sander, Pierre Ablin, and Gabriel Peyré. Do residual neural networks discretize neural ordinary differential equations? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 36520–36532. Curran Associates, Inc., 2022. doi: 10.52202/068431-2646. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/ecc38927fe5148c66bee64ee8fed1e76-Paper-Conference.pdf.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & amp; distillation through attention. In Marina Meila and Tong Zhang (eds.), *(ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4344–4353, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1443. URL https://aclanthology.org/D19-1443/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020. URL https://arxiv.org/abs/2006.04768.

Nikolay Yudin, Alexander Gaponov, Sergei Kudriashov, and Maxim Rakhuba. Pay attention to attention distribution: A new local lipschitz bound for transformers, 2025. URL https://arxiv.org/abs/2507.07814.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=1eQT9OzfNQ.

Yaofeng Desmond Zhong, Tongtao Zhang, Amit Chakraborty, and Biswadip Dey. A neural ODE interpretation of transformer layers. In *The Symbiosis of Deep Learning and Differential Equations II*, 2022. URL https://openreview.net/forum?id=nA9hvYMQCy.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62429–62442. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhu24f.html.

A Appendix

A.1 Numerical integration and computational cost

All costs below are stated per sample; multiply by batch size B to obtain the total cost over a mini-batch. We integrate equation 16 over $[0, 1]$ using `torchdiffeq`, with a uniform grid of S evaluation points and the explicit Euler solver:

$$X_{k+1} = X_k + \Delta t\, F(t_k, X_k|\theta), \qquad \Delta t = \frac{1}{S-1}. \tag{18}$$

Euler performs $S - 1$ function evaluations, so 13 grid points correspond to NFE 12.

Cost of one fixed SSoG evaluation. We count multiply-accumulate operations, reading the cost of each contraction off its index set as the product of the ranges of all free and all summed indices. The class token adds one token to the projections and is excluded from spatial filtering; since N already counts it (§2.3), all costs below are written in terms of N.

The value projection of equation 1 costs $\mathcal{O}(ND^2)$, and is the only input projection, since SSoG forms no queries and no keys where dense attention needs three. In the vertical contraction $U_r = A_{r,y}V$ of equation 8, each of the H heads and R atoms multiplies an $n_y \times n_y$ matrix by an $n_y \times (n_x D^h)$ matrix, giving

$$H\,R\,n_y^2\,n_x\,D^h = \mathcal{O}\big(DR\,n_y^2 n_x\big) \tag{19}$$

after recombining $HD^h = D$. In the horizontal contraction $\sum_r \lambda_r^h U_r A_{r,x}^\top$, the weights λ_r^h are applied as the atom index is summed, and each head multiplies, for every atom, an $(n_y D^h) \times n_x$ matrix by an $n_x \times n_x$ matrix, giving

$$H\,R\,n_y\,n_x^2\,D^h = \mathcal{O}\big(DR\,n_y n_x^2\big)\,. \tag{20}$$

The class-token average of equation 14 costs $\mathcal{O}(ND)$ and the output projection $\mathcal{O}(ND^2)$. Adding the two contractions and using $n_y n_x = N - 1$, spatial mixing costs

$$\mathcal{O}\big(DR\,\big(n_y^2 n_x + n_y n_x^2\big)\big) = \mathcal{O}(DR\,N(n_y + n_x))\,, \tag{21}$$

which for a square grid with $n_y = n_x = n$ and $N - 1 = n^2$ becomes

$$\mathcal{O}(DRn^3) = \mathcal{O}\Big(DRN^{3/2}\Big)\,. \tag{22}$$

The exponent $3/2$ rather than 1 is the price of separability on a two-dimensional grid, since one axis of length $\sqrt{N}$ remains inside each contraction.

Spatial mixing overtakes the projections when $DRN^{3/2} > ND^2$, that is when $R\sqrt{N} > D$, while for dense attention the token interaction overtakes them when $DN^2 > ND^2$, that is when $N > D$. At $D = 768$ with $R = 4$ and patch size 16, dense attention becomes interaction-bound above $N = 768$ tokens, an input resolution of about 443, whereas SSoG would only become mixing-bound beyond a resolution of 3000. Across the whole range we benchmark, SSoG stays projection-bound while dense attention crosses into the quadratic regime part way through, which is the mechanism behind the crossover reported in §3.3.

Storage. Dense self-attention stores an attention matrix of size $\mathcal{O}(HN^2)$, and SSoG stores none. Its axis kernels are shared across the batch and cost $\mathcal{O}(HR(n_y^2 + n_x^2))$ independently of B, while the atom-wise intermediates U require $\mathcal{O}(DRN)$. The latter is linear in N and explains the memory behaviour of the looped setting in §3.3, where the dense baseline already carries little parameter memory.

Cost of content-steered SSoG. Steering still avoids the $N \times N$ query-key matrix, but its axis kernels depend on the batch index and on the full query position rather than being shared. Building the query-dependent kernel tables costs $\mathcal{O}(NHR(n_y + n_x))$, and the three linear predictors of equation 9 cost $\mathcal{O}(NDHR)$. Both are dominated by the contraction itself, by a factor D^h for the tables and $\sqrt{N}/H$ for the predictors, the latter holding at every resolution we test although only narrowly at 224. Steering therefore has the same leading-order complexity as fixed SSoG. What changes is arithmetic intensity: in the fixed variant each kernel entry is reused across the batch and along one spatial axis, while under steering each entry is used once.

A.1.1 Isolating the memory scaling of attention operation

Figure 4 shows that the memory of SSoG advantage over dense attention behaves differently across settings: it grows with resolution in the continuous-depth setting but shrinks (discrete) or reverses (looped) as resolution increases. Since the dense baselines of the discrete and looped settings both use the fused SDPA from PyTorch, while the continuous-depth baseline (ODE-ViT) does not. We isolate the attention operator itself stripped of every other confound, including patch embedding, MLP, classifier head, and weight sharing to test whether this implementation difference alone explains the pattern.

Table 3: Peak allocated memory (MiB) of the attention operator alone, at matching token count.

Resolution	Tokens (N)	SSoG	Dense (fused MHA)	Ratio ($\times$)
224	197	47.6	29.2	1.63
384	577	113.0	56.0	2.02
512	1025	184.5	88.3	2.09
640	1601	281.3	128.1	2.20
768	2305	395.8	181.4	2.18
1024	4097	689.0	304.3	2.26

Table 4: Forward FLOPs per image (GFLOPs, 2 per multiply-add) at dim768 with 12 block applications. The counts are identical, to within 0.3 GFLOPs, for the continuous (12 ODE steps), discrete (12 blocks) and looped (12 iterations) settings. Attention includes the projections; element-wise operations are not counted for either operator. Reduction is SSoG relative to dense SDPA attention (MHA).

		Whole model			Attention		
Resolution	Tokens	MHA	SSoG	Reduction (%)	MHA	SSoG	Reduction (%)
224	197	35.1	**28.4**	19.2	12.6	**6.0**	52.6
384	577	111.0	**84.3**	24.0	44.9	**18.4**	59.1
512	1025	214.1	**151.1**	29.4	96.8	**33.9**	65.0
640	1601	368.3	**237.9**	35.4	185.1	**54.9**	70.4
768	2305	590.1	**345.3**	41.5	326.4	**81.7**	75.0
1024	4097	1319.6	**623.6**	52.7	850.8	**154.9**	81.8

We instantiate `SSOGAttention` (fixed, non-steered) and `torch.nn.MultiheadAttention` (matching $D = 768$, $H = 12$, the exact configuration used by the dense baselines) in isolation, feed both an identical $[B, N, D]$ input at token counts N matching each benchmarked resolution (batch 8, patch size 16), reset the peak memory counter, and measure only that forward call of the module.

Table 3 reports that the ratio grows monotonically with resolution: the attention module of SSoG alone uses 63% more memory than the fused dense kernel at N=197, rising to 126% more at N=4097. Fitting a power law (memory $\propto N^k$) to each series gives $k \approx 0.88$ for SSoG and $k \approx 0.78$ for the fused dense kernel. We attribute this to an implementation gap rather than an inherent property of the separable formulation: the kernel construction and mixture application of SSoG are built from generic PyTorch broadcast and reshape operations, whereas SDPA dispatches to a fused, memory-optimized CUDA kernel.

Furthermore, in order to isolate the computation beyond the kernels and the hardware optimization Table 4 reports the numbers of the Flops among all settings.

A.2 Attention sinks

Dense attention can develop *attention sinks*: tokens that receive a disproportionate share of attention from every query, largely independent of the input. Fixed SSoG has no mechanism to form them. Its patch-to-patch weights depend only on the relative offset between two patches, so every key receives the same total attention up to renormalization at the image border, and patches never attend to the class token, which only pools. Content steering can move a kernel, but by at most $\alpha_\mu M$ tokens, so it cannot concentrate attention from the whole image onto one token.

We measure this on the trained models (Table 5). A sink requires three things at once: a key that receives much more than uniform attention (*sink score*), the same key across images (*same key*: the share of images whose most-attended key is the head's most frequent one), and persistence across layers, which the residual attention rollout captures because it compounds only concentration that the heads and layers agree on.

Table 5: Attention-sink diagnostics of trained models on 512 validation images. First 3 settings are on CIFAR100 and the last setting on ImageNet. *Sink score*: attention received by the most-attended key relative to uniform (1 = no sink), averaged over heads, as the mean over layers, for the worst layer, and for the residual attention rollout $\prod_l (A_l + I)/2$ of the head-averaged maps. *Same key*: share of images whose most-attended key is the head's most frequent one (averaged over heads; most consistent layer); for fixed SSoG it is 100% by construction, because its content-independent kernel always peaks at an image corner through border renormalization. *CLS*: attention mass from patch queries to the class token (- where the model has none; SSoG patches never attend to it). *High-norm*: share of final pre-norm patch tokens with norm above $3\times$ the per-image median.

		Sink score					
Setting	Operator	Mean	Worst layer	Rollout	Same key (%)	CLS (%)	High-norm (%)
Continuous	MHA	11.3	22.9	1.7	6	0.3	0.0
	SSoG	2.2	2.2	1.3	100	-	0.2
Discrete	MHA	3.8	8.1	5.0	49	0.9	0.0
	SSoG	2.6	3.3	1.5	100	-	0.3
	SSoG steered	3.3	5.3	1.7	60	-	0.7
Looped	MHA	2.8	3.5	2.8	64	1.0	0.1
	SSoG	2.0	2.0	1.4	100	-	2.6
	SSoG steered	2.3	2.9	1.6	44	-	0.1
Discrete	MHA	20.6	40.6	41.4	91	8.1	3.5
	SSoG	2.0	2.5	1.1	100	-	29.0

Only the ImageNet dense model meets all three. In its worst layer the most-attended key receives $40.6\times$ the uniform mass, it is the same token in 91% of images, and the rollout compounds it to $41.4\times$; 8.1% of patch attention also flows to the class token. The CIFAR-100 dense models each fail at least one condition. The continuous one has strong single-head peaks ($22.9\times$ in the worst layer), but the attended key changes from image to image (same key in 6% of images) and rollout washes it out ($1.7\times$): content-dependent focusing rather than a sink. The discrete one is intermediate ($8.1\times$, same key in 49% of images, $5.0\times$ under rollout). The looped one has a higher same-key rate (64%) but a low worst-layer score ($3.5\times$), so consistency alone does not make a sink: all three conditions must hold together.

Fixed SSoG stays at or below $3.3\times$ in every layer and 1.1-$1.5\times$ under rollout in every setting. Its most-attended key is identical across images (100%) only because its content-independent kernel always peaks at an image corner through border renormalization, a geometric effect that no input can move. Steering raises the worst-layer score moderately ($5.3\times$ discrete, $2.9\times$ looped) and at the same time lowers the same-key rate from 100% to 60% and 44%, while rollout stays near uniform ($1.7\times$ and $1.6\times$). The added concentration therefore follows the content, as in the continuous dense model, rather than forming a fixed sink.

SSoG does not, however, avoid feature-norm artifacts altogether. On ImageNet, 29% of its final patch tokens have a norm above $3\times$ the image median, against 3.5% for the dense model. These tokens are concentrated at the image border, which suggests a positional rather than content-driven origin, unlike the isolated high-norm tokens associated with registers in dense ViTs (Darcet et al., 2024). We therefore claim the absence of attention sinks, not of norm artifacts.

A.3 Ablations

This appendix reports additional CIFAR-100 ablations that support the design choices made in §3. Unless stated otherwise, all rows use $D = 768$, $H = 12$, $R = 4$, NFE 12, and the training recipe of §3

A.3.1 Sigma floor

The lower bound $\sigma_{\min}$ on atom width (equation 4) trades off sharply local kernels against enforced smoothing. Table 6 sweeps this floor for fixed SSoG. The default value 0.25 performs best; lowering

it to 0.10 costs 0.26 points and raising it to 0.50 costs 0.97 points, and the learned minimum sigma tracks the constraint in each case, confirming that the floor is active rather than a slack bound.

Table 6: Effect of the sigma floor $\sigma_{\min}$ on fixed SSoG.

$\sigma_{\min}$	Acc@1	Val. loss	Learned σ min
0.10	73.73	1.14	0.182
0.25	**74.22**	**1.12**	0.366
0.50	73.02	1.14	0.578

A.3.2 Steering baselines

Table 7 reports the results of steered SSoG, which serve as the steering baselines. For the continuous-depth model (ODE-SSoG), steering is evaluated at different NFE; for the discrete and looped models, it is evaluated at 12 blocks and 12 iterations, respectively.

A.3.3 Steering initialization

Table 8 isolates how quickly the content-steered pathway should activate. Warm initialization of full steering underperforms cold initialization by 5.33 points and plateaus early. Within center-only steering, moving the initialization of the gate $\tilde{\alpha}_\mu$ of equation 12 from -8 to -4 costs 1.80 points despite both runs converging to a nearly identical final scale α_μ (0.127 vs. 0.126), indicating that the optimization trajectory, not only the final steering magnitude, determines the outcome.

A.3.4 Maximum steering offset

Table 9 sweeps the maximum center displacement M in the full content-steered model (equation 9). The offset sweep is monotonic over the tested range: constraining M from 7 to 4 gains 1.59 points, and from 4 to 2 gains a further 0.36 points. Smaller M is partially compensated by a larger learned scale α_μ, but the maximum displacement $\alpha_\mu M$ still grows with M (0.87, 0.89, 0.94 for $M = 2, 4, 7$), so the nominal limit remains the binding constraint.

A.3.5 Discrete-setting width sweep

Table 10 gives the full width sweep: a 12-block discrete transformer using SSoG at every block, compared against a matched dense-attention transformer of identical depth, width, and training recipe.

A.3.6 Resolution Robustness

In order to validate the robustness of our method whe test the trained models with different patch size. Table 13 reports the results of models trained at patch 16 (224px, 14 × 14 tokens) evaluated at patch 14 and 8 without retraining. Patch-embedding kernels are PI-resized and position embeddings bicubically interpolated for every model; the rescaled SSoG row also scales μ, σ by the grid ratio so each Gaussian covers the same image.

A.4 Effective atoms

To measure how many of the R Gaussian atoms a head actually uses, we take its learned mixture weights $\lambda_r^h = \mathrm{softmax}_r(\ell_r^h)$ and compute their Shannon entropy and its exponential,

$$R_{\text{eff}}^h = \exp\Big(-\sum_{r=1}^{R} \lambda_r^h \log \lambda_r^h \Big), \qquad \overline{R}_{\text{eff}} = \frac{1}{H} \sum_{h=1}^{H} R_{\text{eff}}^h. \tag{23}$$

R_{eff}^h ranges from 1 (all mass on one atom) to R (uniform mixture), and we report its mean over heads. It is computed on the fixed per-head weights, before content-dependent steering, so it shows how the model spends its atom budget rather than any single query's mixture.

Table 7: Fixed versus steered SSoG on CIFAR-100 across the three architecture settings. Steps are ODE function evaluations (continuous), blocks (discrete) or loop iterations (looped). Time is the W&B training runtime for 500 epochs.

Setting	Variant	Steps	Acc@1	Acc@3	Acc@5	Time (h)
Continuous	Fixed	6	69.70	85.74	90.23	25.5
	Fixed	8	72.16	87.30	91.05	33.2
	Fixed	12	74.22	88.59	92.38	47.8
	Steered	6	70.36	86.04	90.67	31.5
	Steered	8	72.00	87.16	91.42	40.3
	Steered	12	70.89	86.89	91.30	59.5
Discrete	Fixed	12	78.63	90.12	93.38	33.40
	Steered	12	81.17	90.21	91.67	57.81
Looped	Fixed	12	79.80	90.58	93.10	23.90
	Steered	12	81.20	91.12	93.56	49.23

Table 8: Effect of steering initialization at NFE 12, $R = 4$.

Initialization	Final Acc@1	Val. loss	Instabilities
Cold, full steering	70.89	1.2017	None
Warm, full steering	65.56	1.3714	Early plateau
Center-only, $\tilde{\alpha}_\mu = -8$	**73.33**	**1.1546**	None
Center-only, $\tilde{\alpha}_\mu = -4$	71.53	1.2138	None

A.4.1 IMAGENET WIDTH CONFIRMATION

Table 14 reports the ImageNet width sweep for fixed SSoG at constant head dimension $D^h = 64$. Best-epoch Top-1 improves from 55.65% to 62.43% to 64.02% as D increases from 384 to 576 to 768, with diminishing gains from 576 to 768.

A.5 AUGMENTATION AND IMAGENET-TO-CIFAR-100 TRANSFER

Table 15 sweeps training augmentation strength for fixed SSoG trained from scratch on CIFAR-100: stronger augmentation is preferred when learning the full representation from random initialization. Table 16 instead adapts to CIFAR-100, either through a linear probe or full fine-tuning; here weaker augmentation is preferred, reversing the from-scratch trend, consistent with adapting rather than learning a representation from random initialization.

Table 9: Effect of the maximum steering offset M, full steering, NFE 12.

M	Acc@1	Val. loss	Mean $\lvert\mu\rvert$
2	**72.84**	**1.1385**	0.529
4	72.48	1.1643	0.535
7	70.89	1.2017	0.475

Table 10: Width scaling on CIFAR-100, dense attention versus SSoG, across the three architecture settings. Params are per model; bold marks the smaller parameter count and the better Acc@1 within each row.

Setting	D	MHA Params(M)	SSoG Params	MHA Acc@1	SSoG Acc@1
Continuous	384	3.07	**2.8**	69.15	**71.37**
	576	6.59	**5.9**	69.36	**73.48**
	768	11.45	**10.3**	70.47	**74.22**
Discrete	384	21.69	**18.5**	74.51	**77.56**
	576	48.47	**41.0**	74.20	**78.18**
	768	85.86	**71.6**	74.67	**78.63**
Looped	384	2.18	**1.9**	72.04	**77.46**
	576	4.60	**3.9**	71.24	**79.38**
	768	7.91	**6.7**	72.30	**79.80**

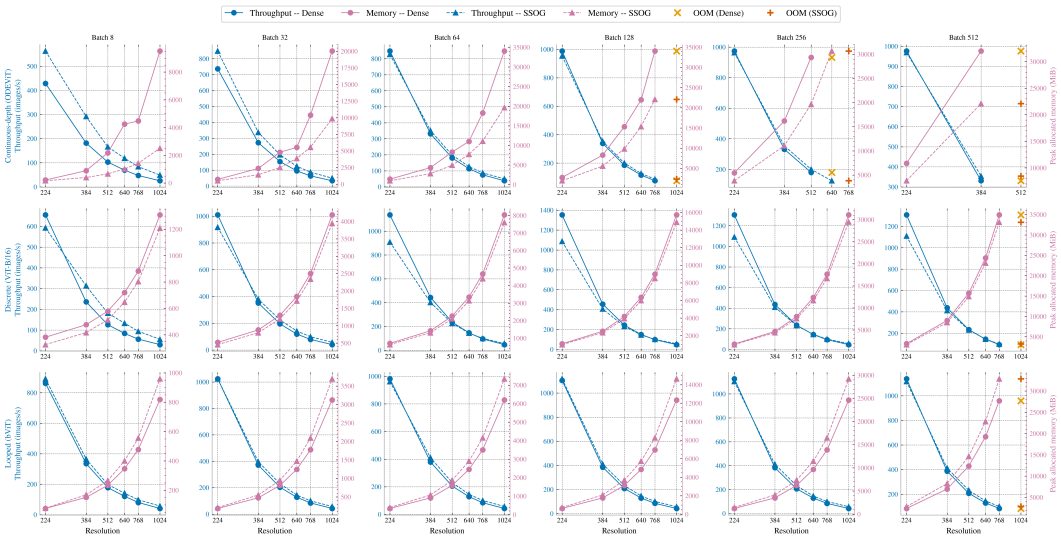


Figure 5: Throughput and peak memory versus input resolution, dense attention versus SSoG, across all three architecture settings (rows: continuous-depth ODE-ViT, discrete ViT-B/16, looped bViT) and six batch sizes (columns: 8, 32, 64, 128, 256, 512), at the paper's flagship width ($D = 768$, $H = 12$; NFE 12 for the continuous-depth setting, depth 12 for the discrete and looped settings). Each panel shares a log-scaled resolution axis (224-1024) with two y-axes: throughput in images/s (blue, left) and peak allocated memory in MiB (magenta, right). Solid circles/lines denote dense attention; dashed triangles denote SSoG. An $\times$ (dense) or $+$ (SSoG) marks the first resolution at which that configuration ran out of GPU memory; its vertical position is the last successfully measured value at that batch size, not a measured point at the marked resolution, and no further resolutions are plotted for that curve beyond it. All measurements are FP16 autocast (model parameters and inputs in FP32) on a single NVIDIA A40, randomly initialized weights, a single run per configuration (no repeated-seed or repeated-measurement uncertainty), 10 warmup and 50 measured iterations per point.

Table 11: Capacity and depth ablations on CIFAR-100 for the three SSoG model families. The Depth column gives the number of blocks for the Discrete family, the number of loop iterations for the Looped family, and the number of ODE function evaluations (NFE) for the Continuous family. The Heads and Atoms blocks are evaluated at NFE 12 where applicable, while the Depth block varies depth or NFE at fixed $D = 768$, $H = 12$, $R = 4$. The column highlighted in blue is the parameter varied within each block; bold marks the best metric within each block.

Family	Ablation	D	H	R	Depth	Par(M)	Acc@1	Acc@3	Acc@5
DISCRETE	Heads	768	6	4	12	71.6	**79.97**	89.99	91.94
			12				78.63	**90.12**	**93.38**
			24				79.52	88.59	90.14
	Atoms	768	12	1	12	71.6	79.62	89.38	91.52
				2			**80.34**	89.83	91.69
				4			78.63	**90.12**	**93.38**
				8			79.93	88.67	90.35
	Depth	768	12	4	6	36.2	79.55	**90.93**	**93.89**
					8	48.0	**79.70**	90.24	92.66
					12	71.6	78.63	90.12	93.38
LOOPED	Heads	768	6	4	12	6.7	79.32	90.49	92.85
			12				**79.80**	90.58	93.10
			24				79.51	**90.59**	**93.32**
	Atoms	768	12	1	12	6.7	**80.31**	**90.68**	93.28
				2			79.67	90.57	93.09
				4			79.80	90.58	93.10
				8			79.73	90.60	**93.34**
	Depth	768	12	4	6	6.7	79.23	**91.18**	**93.96**
					8		78.88	90.18	93.17
					12		**79.80**	90.58	93.10
CONTINUOUS	Heads	768	6	4	12	10.3	73.62	88.39	92.11
			12				74.22	**88.59**	**92.38**
			24				**74.40**	88.44	92.37
	Atoms	768	12	1	12	10.3	73.42	88.25	92.05
				2			**74.13**	88.36	91.95
				4			74.22	**88.59**	92.38
				8			73.99	88.58	**92.47**
	Depth	768	12	4	6	10.3	69.70	85.74	90.23
					8		72.16	87.30	91.05
					12		74.22	88.59	92.38
					16		**74.92**	**89.02**	**92.46**

Table 12: MHA capacity and depth ablations on CIFAR-100 for the Discrete MHA (ViT) and Looped MHA (bViT) families. The column highlighted in blue is the parameter varied within each block; bold marks the best metric within each block.

Family	Ablation	D	H	Depth	Params	Acc@1	Acc@3	Acc@5
DISCRETE	Heads	768	6	12	85.9M	**75.63**	**87.51**	**90.55**
			12			74.67	86.89	90.19
			24			73.70	86.0	89.30
	Depth	768	12	6	43.3M	70.47	84.50	88.61
				8	57.5M	72.22	85.68	89.60
				12	85.9M	**74.67**	**86.89**	**90.19**
LOOPED	Heads	768	6	12	7.9M	**73.18**	**85.80**	**89.41**
			12			72.30	84.37	88.33
			24			70.07	83.13	86.89
	Depth	768	12	6	7.9M	68.72	83.44	87.69
				8		70.99	**84.57**	**88.60**
				12		**72.30**	84.37	88.33

Table 13: Patch-size robustness on CIFAR-100: models trained at patch 16 (224px, 14×14 tokens) evaluated at patch 14 and 8 without retraining. Patch-embedding kernels are PI-resized and position embeddings bicubically interpolated for every model; the rescaled SSoG row also scales μ, σ by the grid ratio so each Gaussian covers the same image area. Acc@1 (%), change vs. patch 16 in parentheses.

Setting	Operator	Patch 16 (14×14)	Patch 14 (16×16)	Patch 8 (28×28)
Continuous	MHA	70.47	70.23 (+0.80)	58.05 (−11.38)
	SSoG	**74.20**	**74.85** (+0.65)	57.44 (−16.76)
	SSoG (rescaled μ, σ)	**74.20**	74.30 (+0.10)	**64.39** (−9.81)
Discrete	MHA	74.67	74.91 (−0.18)	56.92 (−18.17)
	SSoG	**78.63**	79.34 (+0.71)	63.49 (−15.53)
	SSoG (rescaled μ, σ)	**78.63**	**79.48** (+0.85)	**72.09** (−6.93)
Looped	MHA	72.30	72.21 (−0.09)	52.24 (−20.06)
	SSoG	**79.80**	**80.98** (+1.18)	73.56 (−6.74)
	SSoG (rescaled μ, σ)	**79.80**	80.12 (+0.42)	**73.59** (−6.71)

Table 14: ImageNet width sweep, fixed SSoG, $R = 4$, NFE 12.

D	H	Acc@1	Acc@5
384	6	55.65 (245)	78.46
576	9	62.43 (273)	83.37
768	12	**64.02 (254)**	84.45

Table 15: From-scratch augmentation ablation, $D = 768$, $R = 4$, NFE 12.

Augmentation	Best Acc@1 (epoch)	Final Acc@1	Runtime (h)
Strong	**74.22 (488)**	**74.22**	47.77
Moderate	73.09 (487)	72.72	34.08
Weak	62.70 (452)	62.27	33.26

Table 16: ImageNet→CIFAR-100 transfer, $D = 768$ backbone from an ImageNet checkpoint, across all three architecture settings.

Setting	Protocol / Augmentation		Best Acc@1
Continuous	Linear probe (frozen)	Hard	66.26 (43)
	Linear probe (frozen)	Weak	**70.01 (45)**
	Full fine-tune	Hard	80.25 (91)
	Full fine-tune	Moderate	**83.84 (83)**
Discrete	Linear probe (frozen)	Hard	66.77 (48)
	Linear probe (frozen)	Weak	**69.05 (43)**
	Full fine-tune	Hard	86.71 (92)
	Full fine-tune	Moderate	**87.09 (95)**
Looped	Linear probe (frozen)	Hard	68.71 (48)
	Linear probe (frozen)	Weak	**70.96 (47)**
	Full fine-tune	Hard	82.98 (97)
	Full fine-tune	Moderate	**84.32 (99)**

A.6 LIPSCHITZ CONTINUITY OF SSOG

In the continuous-depth setting of equation 16, the token matrix $X(t)$ evolves under the vector field $F(\cdot, \cdot|\theta)$. By the Picard–Lindelöf theorem (Coddington & Levinson, 1955), if F is continuous in t and Lipschitz continuous in X with a constant independent of t and of X, i.e. globally Lipschitz, then equation 16 has a unique solution on the whole interval $[0, 1]$ for every initial condition X_0. If F is only Lipschitz on bounded sets, the same theorem guarantees existence and uniqueness only on a possibly shorter interval, which extends to $[0, 1]$ provided the trajectory remains bounded.

The vector field $F(\cdot, \cdot|\theta)$ is built from the SSoG spatial operator, the value and output projections W_V^h and W_O, the feed-forward block and layer normalization. Since sums and compositions of globally Lipschitz maps are globally Lipschitz, it suffices to bound each component. Linear maps are globally Lipschitz with constant equal to their spectral norm; the feed-forward block is globally Lipschitz with constant at most the product of the spectral norms of its weight matrices and the Lipschitz constant of its activation; and layer normalization with a stabilizing constant $\varepsilon_{\mathrm{LN}} > 0$ is globally Lipschitz (see the discussion following Proposition 1). The only non-trivial component is therefore the SSoG spatial operator. We show that fixed SSoG is globally Lipschitz (Proposition 1), so that ODE-SSoG with fixed kernels has a unique solution on $[0, 1]$, whereas content-steered SSoG is only Lipschitz on bounded sets (Proposition 2), so that for this variant only the local guarantee applies.

Notation. We recall the notation of §2. For head h, the value projection is $V^h = XW_V^h$ (equation 1). For each atom r and axis $a \in \{x, y\}$, with $\delta_a := q_a - s_a$, let $g_{r,a}(\delta_a) := \exp\big(\log G^a(\delta_a)/\tau\big)$ denote the temperature-scaled Gaussian kernel of atom r, where $\log G^a$ is given by equation 2 and τ is the temperature of equation 3. The normalized axis kernel $A_{r,a}(\delta_a)$ coincides with $A_a(\delta_a)$ from equation 3 evaluated for atom r. The mixture weights $(\lambda_r)_r$ lie in the probability simplex (equation 4), and the per-block class-token aggregate is $v_{\mathrm{cls}} = \frac{1}{N-1}\sum_{p=1}^{N-1} v_p$ (equation 14). All norms on token matrices are Frobenius; norms on linear maps are spectral.

We drop the batch index, since every operation is separable across the batch.

Lemma 1 (Kernel norm)**.** *For each atom r and each axis $a \in \{x, y\}$, the axis kernel of equation 3 satisfies $\|A_{r,a}\|_2 \le \sqrt{\kappa_{r,a}}$, where*

$$\kappa_{r,a} = \frac{\max_{s_a} \sum_{q_a} g_{r,a}(\delta_a)}{\min_{q_a} \sum_{s_a} g_{r,a}(\delta_a)}, \quad \delta_a := q_a - s_a \ \ge\ 1 \tag{24}$$

and $g_{r,a}(\delta_a)$ is the temperature-scaled Gaussian kernel $g_{r,a}(\delta_a) = \exp\big(\log G^a(\delta_a)/\tau\big)$, with $\log G^a$ as in equation 2 evaluated for atom r and τ the temperature of equation 3.

Proof. Since all entries of $A_{r,a}$ are positive, $\|A_{r,a}\|_\infty$ and $\|A_{r,a}\|_1$ are its maximum row sum and maximum column sum, respectively. We bound each in turn.

By equation 3, each row of $A_{r,a}$ is a softmax over the source coordinate s_a, so $\sum_{s_a} A_{r,a}(q_a, s_a) = 1$ for every q_a. Hence $\|A_{r,a}\|_\infty = 1$.

For a fixed column s_a, each entry $A_{r,a}(q_a, s_a)$ equals $g_{r,a}(q_a - s_a)$ divided by the row sum $\sum_{s'_a} g_{r,a}(q_a - s'_a)$. Bounding every such denominator from below by its minimum over q_a gives

$$\sum_{q_a} A_{r,a}(q_a, s_a) \ =\ \sum_{q_a} \frac{g_{r,a}(q_a - s_a)}{\sum_{s'_a} g_{r,a}(q_a - s'_a)} \ \le\ \frac{\sum_{q_a} g_{r,a}(q_a - s_a)}{\min_{q_a} \sum_{s'_a} g_{r,a}(q_a - s'_a)}.$$

The denominator on the right no longer depends on s_a, so taking the maximum over s_a on both sides yields $\|A_{r,a}\|_1 \le \kappa_{r,a}$, with $\kappa_{r,a}$ as in equation 24.

Finally, every real matrix M satisfies $\|M\|_2 \le \sqrt{\|M\|_1 \|M\|_\infty}$, since $\|M\|_2^2 = \rho(M^\top M) \le \|M^\top M\|_\infty \le \|M^\top\|_\infty \|M\|_\infty = \|M\|_1 \|M\|_\infty$. Applying this to $A_{r,a}$ with the two bounds above gives $\|A_{r,a}\|_2 \le \sqrt{\kappa_{r,a}}$. □

Remark. The constant $\kappa_{r,a}$ is finite for every choice of atom parameters, which is all that Proposition 1 requires; with truncated boundaries, however, it does not approach one as the grid grows. A query q_a at the edge of the axis only retains the part of $g_{r,a}(q_a, \cdot)$ that falls inside the grid, whereas an interior column collects the full kernel mass. Consequently $\kappa_{r,a} \approx 1$ only when the effective width $\sqrt{\tau}\,\sigma_{r,a}$ is small compared with the patch spacing and the offset $\mu_{r,a}$ is close to zero; for a centred kernel of larger width $\kappa_{r,a}$ is close to 2, and it grows rapidly with $|\mu_{r,a}|$, since edge rows then retain only a tail of the kernel. The bound $\sqrt{\kappa_{r,a}}$ is therefore conservative.

Proposition 1 (Fixed SSoG is globally Lipschitz)**.** *Let $X \in \mathbb{R}^{N\times D}$ consist of one class token and $N-1 = n_y n_x$ patch tokens arranged on an $n_y \times n_x$ grid. For each head $h \in \{1,\dots,H\}$, assume that the atom parameters $\mu_{r,a}$, $\sigma_{r,a}$, ℓ_r and the temperature τ are fixed, i.e. independent of X. Let $W_V^h \in \mathbb{R}^{D\times D^h}$ be the value projections of equation 1, $W_V = W_V^1 \,\|\, \cdots \,\|\, W_V^H \in \mathbb{R}^{D\times D}$ their concatenation, and $W_O \in \mathbb{R}^{D\times D}$ the output projection. Let*

$$\mathcal{A}^h = \sum_{r=1}^{R} \lambda_r^h \left(A_{r,y}^h \otimes A_{r,x}^h\right) \in \mathbb{R}^{(N-1)\times(N-1)} \tag{25}$$

be the effective spatial kernel of head h on the patch tokens. Then fixed SSoG is a linear map on $\mathbb{R}^{N\times D}$ and, with the Frobenius norm on token matrices,

$$\mathrm{Lip}(\mathrm{SSoG}) \;\le\; \|W_V\|_2\, \|W_O\|_2\, K_{\mathcal{A}}, \tag{26}$$

where

$$K_{\mathcal{A}} := \max_h \sqrt{\tfrac{1}{N-1} + \|\mathcal{A}^h\|_2^2} \;\le\; \max_h \sqrt{\tfrac{1}{N-1} + \max_r \kappa_{r,y}^h\, \kappa_{r,x}^h}, \tag{27}$$

and $\kappa_{r,a}^h$ is the constant of Lemma 1 for atom r of head h. The bound depends only on the parameters and not on the input.

Proof. Fix a head h, order the tokens with the class token last, as in §2, and write $V^h = XW_V^h = \begin{pmatrix} V_{\mathrm{P}}^h \\ v_{\mathrm{cls}}^h \end{pmatrix}$, where $V_{\mathrm{P}}^h \in \mathbb{R}^{(N-1)\times D^h}$ collects the patch rows. Vectorizing the grid in row-major order and using $\mathrm{vec}(AUB^\top) = (A \otimes B)\,\mathrm{vec}(U)$, equation 8 gives the filtered patch rows $\mathcal{A}^h V_{\mathrm{P}}^h$, while by equation 14 the class row is $\frac{1}{N-1}\mathbf{1}^\top V_{\mathrm{P}}^h$. Hence the output of head h is $M^h X W_V^h$, with

$$M^h = \begin{pmatrix} \mathcal{A}^h & 0 \\ \frac{1}{N-1}\mathbf{1}^\top & 0 \end{pmatrix} \in \mathbb{R}^{N\times N}.$$

In the fixed variant M^h does not depend on X. Denoting by

$$Z^h = M^h X W_V^h \in \mathbb{R}^{N\times D^h}$$

the output of head h, the SSoG output is obtained by concatenating the heads and applying the output projection,

$$\mathrm{SSoG}(X) = \left(Z^1 \,\|\, \cdots \,\|\, Z^H\right) W_O,$$

which is linear in X; its Lipschitz constant is therefore its operator norm. Since the heads occupy disjoint column blocks,

$$\begin{aligned} \|\mathrm{SSoG}(X)\|_F \le& \|W_O\|_2 \Big(\sum_h \|Z^h\|_F^2\Big)^{1/2} \le \|W_O\|_2 \max_h \|M^h\|_2\, \|XW_V\|_F \\ \le& \|W_V\|_2 \|W_O\|_2 \max_h \|M^h\|_2\, \|X\|_F. \end{aligned}$$

Since the last column of M^h vanishes, $\|M^h\|_2$ equals the spectral norm of its first $N-1$ columns; since, moreover, the squared spectral norm of a matrix equals the spectral norm of its Gram matrix,

$$\|M^h\|_2^2 = \left\| \begin{pmatrix} \mathcal{A}^h \\ \frac{1}{N-1}\mathbf{1}^\top \end{pmatrix} \right\|_2^2 = \left\| (\mathcal{A}^h)^\top \mathcal{A}^h + \tfrac{1}{(N-1)^2}\mathbf{1}\mathbf{1}^\top \right\|_2 \le \|\mathcal{A}^h\|_2^2 + \tfrac{1}{N-1},$$

where the inequality follows from the triangle inequality and $\|\mathbf{1}\mathbf{1}^\top\|_2 = N-1$, so that $\max_h \|M^h\|_2 \le K_{\mathcal{A}}$, which gives equation 26. For the inequality in equation 27, $\|A\otimes B\|_2 = \|A\|_2\|B\|_2$ and $(\lambda_r^h)_r$ lies in the simplex by equation 4, so equation 25 is a convex combination and, by Lemma 1, $\|\mathcal{A}^h\|_2 \le \max_r \|A_{r,y}^h\|_2 \|A_{r,x}^h\|_2 \le \max_r \sqrt{\kappa_{r,y}^h \kappa_{r,x}^h}$. □

Proposition 1 has two immediate consequences. For the discrete and looped models, the residual block of equation 15 is globally Lipschitz with constant at most $1 + \|W_V\|_2\|W_O\|_2 K_{\mathcal{A}}$. Since $K_{\mathcal{A}}$ is finite for any choice of parameters, the only factors not determined by the kernel are the projection norms, which spectral normalization bounds explicitly and weight decay keeps moderate in practice. For ODE-SSoG, the vector field $F(\cdot, \cdot|\theta)$ of equation 16 composes fixed SSoG with the feed-forward block and layer normalization. The feed-forward block is globally Lipschitz, with constant at most the product of the spectral norms of its weight matrices and the Lipschitz constant of its activation. Layer normalization with gain γ and stabilizing constant $\varepsilon_{\mathrm{LN}} > 0$ added to the variance has Jacobian of spectral norm at most $\max_i |\gamma_i|/\sqrt{\varepsilon_{\mathrm{LN}}}$, so it is globally Lipschitz as well. Hence $F(\cdot, \cdot|\theta)$ is continuous in t and globally Lipschitz in X with a constant independent of t, and the Picard–Lindelöf theorem yields a unique solution of equation 16 on $[0, 1]$.

We now turn to content-steered SSoG. Steering acts on each head independently: the predictors of equation 9 modify only the atom parameters of their own head, and different heads interact only through the output projection W_O, which is handled exactly as in Proposition 1. We therefore first analyse a single head. In Lemma 2 and its proof we fix one head and, to lighten notation, omit the superscript h from all of its quantities; the superscript is restored in Proposition 2, where all heads are combined.

Intuition and notation for Lemma 2. Steering can only move each atom within a bounded range. Since $|\tanh| < 1$, equation 9 shifts the center $\mu_{r,a}$ by at most $\alpha_\mu M$ and multiplies the spread $\sigma_{r,a}$ by a factor between $e^{-\alpha_\sigma}$ and e^{α_σ}. Together with the floor $\sigma_{r,a} \geq \sigma_{\min}$, this confines every steered spread to an interval $[\sigma_\star, \sigma^\star]$, and every distance between a grid offset $\delta_a \in \{-(n_a-1), \ldots, n_a-1\}$ and a steered center to at most Δ, where

$$\sigma_\star = \sigma_{\min} e^{-\alpha_\sigma}, \qquad \sigma^\star = \max_{r,a} \sigma_{r,a} e^{\alpha_\sigma}, \qquad \Delta = \max_{r,a} \big(n_a - 1 + |\mu_{r,a}|\big) + \alpha_\mu M. \tag{28}$$

Thus $\sigma_\star$ is the narrowest spread any atom can reach, $\sigma^\star$ the widest, and Δ the farthest a steered center can lie from a grid offset. These three quantities control how sharply the kernel reacts to its parameters: a narrow Gaussian evaluated far from its center changes quickly when its center or spread move. Lemma 2 makes this precise: the steered parameters stay in a box (i), and the kernel row is Lipschitz in the query token (ii), with a constant that combines the sensitivity of the predictors ($\|W_\mu\|_2$, $\|W_\sigma\|_2$, $\|W_\lambda\|_2$), the steering scales, and the sensitivity of the kernel to its parameters.

Lemma 2 (Steered kernel)**.** *Consider a single head, and let its atom parameters be steered by the query token x_q as in equation 9, with maximum displacement M, predictor weights $W_\mu, W_\sigma, W_\lambda$ and scales $\alpha_\mu, \alpha_\sigma, \alpha_\lambda > 0$ as in equation 12, and with $\sigma_{r,a} \geq \sigma_{\min}$ and $\tau \geq \tau_0$ as in equation 4. Denote by $A_{r,a}(x_q) \in \mathbb{R}^{n_a}$ the row of $A_{r,a}$ associated with the query position q, and by $\mathcal{A}(x_q) = \sum_r \lambda_r(x_q)\, A_{r,y}(x_q) \otimes A_{r,x}(x_q)$ the corresponding row of the effective kernel. Let $\sigma_\star$, $\sigma^\star$ and Δ be as in equation 28. Then:*

(i) for every x_q, r and a,

$$\big(\mu_{r,a}(x_q), \sigma_{r,a}(x_q)\big) \in \Theta_{r,a} := [\mu_{r,a} - \alpha_\mu M,\ \mu_{r,a} + \alpha_\mu M] \times [\sigma_\star, \sigma^\star]; \tag{29}$$

(ii) the row $\mathcal{A}(x_q)$ of the effective kernel is Lipschitz in x_q with respect to the Euclidean norm, with constant

$$L_{\mathcal{A}} = \tfrac{1}{2}\sqrt{R/2}\ \alpha_\lambda \|W_\lambda\|_2 + \max_r \big(L_{r,y} + L_{r,x}\big), \tag{30}$$

where

$$L_{r,a} = \frac{\sqrt{n_a}}{2\tau_0} \left(\frac{\Delta}{\sigma_\star^2}\, \alpha_\mu M \|W_\mu\|_2 + \Big(\frac{1}{\sigma_\star} + \frac{\Delta^2}{\sigma_\star^3}\Big) \sigma^\star \alpha_\sigma \|W_\sigma\|_2 \right). \tag{31}$$

Proof. (i) Since $|\tanh| \leq 1$ and $\sigma_{r,a} \geq \sigma_{\min}$, $|\mu_{r,a}(x_q) - \mu_{r,a}| \leq \alpha_\mu M$ and $\sigma_\star \leq \sigma_{r,a}(x_q) \leq \sigma^\star$.

(ii) Fix two tokens x_q and x'_q. For the center, write $u = [W_\mu x_q + b_\mu]_{r,a}$ and $u' = [W_\mu x'_q + b_\mu]_{r,a}$ for the pre-activations of its predictor, so that $|u - u'| \leq \|W_\mu\|_2\, \|x_q - x'_q\|_2$. Since $\tanh$ is 1-Lipschitz,

$$|\mu_{r,a}(x_q) - \mu_{r,a}(x'_q)| = \alpha_\mu M\, |\tanh u - \tanh u'| \leq \alpha_\mu M\, \|W_\mu\|_2\, \|x_q - x'_q\|_2.$$

For the spread, write likewise $v = [W_\sigma x_q + b_\sigma]_{r,a}$ and $v' = [W_\sigma x'_q + b_\sigma]_{r,a}$, and regard the steered spread as the function $v \mapsto \sigma_{r,a} \exp(\alpha_\sigma \tanh v)$. Its derivative satisfies

$$0 \le \frac{\partial \sigma_{r,a}}{\partial v} = \sigma_{r,a}\, e^{\alpha_\sigma \tanh v}\, \alpha_\sigma \tanh'(v) \le \sigma^\star \alpha_\sigma,$$

because the steered spread never exceeds $\sigma^\star$ and $0 < \tanh' \le 1$. By the mean value theorem,

$$|\sigma_{r,a}(x_q) - \sigma_{r,a}(x'_q)| \le \sigma^\star \alpha_\sigma\, |v - v'| \le \sigma^\star \alpha_\sigma\, \|W_\sigma\|_2\, \|x_q - x'_q\|_2.$$

The axis row $A_{r,a}(x_q)$ is the softmax of $z \in \mathbb{R}^{n_a}$, $z_s = \log g_{r,a}(q_a - s)$, where $\log g_{r,a} = \tau^{-1} \log G^a$, with $\log G^a$ as in equation 2, is evaluated at $\mu_{r,a}(x_q)$ and $\sigma_{r,a}(x_q)$; let z' be the same vector at x'_q. On $\Theta_{r,a}$, every grid offset satisfies $|\delta_a - \mu| \le \Delta$, so $\tau \ge \tau_0$ gives

$$\left|\frac{\partial \log g_{r,a}}{\partial \mu}\right| \le \frac{\Delta}{\tau_0\, \sigma_\star^2}, \qquad \left|\frac{\partial \log g_{r,a}}{\partial \sigma}\right| \le \frac{1}{\tau_0}\left(\frac{1}{\sigma_\star} + \frac{\Delta^2}{\sigma_\star^3}\right). \tag{32}$$

Since $\Theta_{r,a}$ is convex, the segment joining $\big(\mu_{r,a}(x_q), \sigma_{r,a}(x_q)\big)$ and $\big(\mu_{r,a}(x'_q), \sigma_{r,a}(x'_q)\big)$ stays in $\Theta_{r,a}$, where the bounds equation 32 hold. The mean value theorem along this segment, together with the two estimates above, gives for every source coordinate s

$$\begin{aligned} |z_s - z'_s| \le & \frac{\Delta}{\tau_0 \sigma_\star^2}\, \big|\mu_{r,a}(x_q) - \mu_{r,a}(x'_q)\big| \\ & + \frac{1}{\tau_0}\Big(\frac{1}{\sigma_\star} + \frac{\Delta^2}{\sigma_\star^3}\Big)\big|\sigma_{r,a}(x_q) - \sigma_{r,a}(x'_q)\big| \le \frac{2}{\sqrt{n_a}}\, L_{r,a}\, \|x_q - x'_q\|_2, \end{aligned} \tag{33}$$

where the last step is the definition equation 31 of $L_{r,a}$. It remains to pass from the logits z to the normalized row. The softmax is $\frac{1}{2}$-Lipschitz in the Euclidean norm, because its Jacobian $\mathrm{diag}(p) - pp^\top$ is positive semidefinite with spectral norm at most $\frac{1}{2}$. Hence

$$\|A_{r,a}(x_q) - A_{r,a}(x'_q)\|_2 \le \tfrac{1}{2}\|z - z'\|_2 \le \tfrac{1}{2}\sqrt{n_a}\, \max_s |z_s - z'_s| \le L_{r,a}\, \|x_q - x'_q\|_2. \tag{34}$$

For the effective kernel, write $c_r = A_{r,y}(x_q) \otimes A_{r,x}(x_q)$, c'_r for its value at x'_q, and $d = \lambda(x_q) - \lambda(x'_q)$, so that

$$\mathcal{A}(x_q) - \mathcal{A}(x'_q) = \sum_r d_r\, c_r + \sum_r \lambda_r(x'_q)\, (c_r - c'_r). \tag{35}$$

We bound the two sums separately.

Second sum. All rows are probability vectors, so their Euclidean norm is at most one. For any vectors a, a', b, b',

$$\|a \otimes b - a' \otimes b'\|_2 \le \|a - a'\|_2\, \|b\|_2 + \|a'\|_2\, \|b - b'\|_2, \tag{36}$$

and applying it with $a = A_{r,y}(x_q)$, $b = A_{r,x}(x_q)$ and their counterparts at x'_q gives

$$\|c_r - c'_r\|_2 \le \big(L_{r,y} + L_{r,x}\big)\, \|x_q - x'_q\|_2. \tag{37}$$

Since the weights $\lambda_r(x'_q)$ form a convex combination, the second sum is at most $\max_r (L_{r,y} + L_{r,x})\, \|x_q - x'_q\|_2$.

First sum. Its coefficients add up to zero, $\sum_r d_r = 0$, so splitting d into its positive and negative parts gives $\sum_r d_r c_r = \frac{1}{2}\|d\|_1\, (P - Q)$ for two probability vectors P and Q. Since $\|P - Q\|_2^2 \le \|P - Q\|_1 \|P - Q\|_\infty \le 2$,

$$\Big\| \sum_r d_r c_r \Big\|_2 \le \frac{\|d\|_1}{\sqrt{2}} \le \sqrt{R/2}\, \|d\|_2. \tag{38}$$

Finally, the mixture weights are a softmax of $\ell_r + \alpha_\lambda \tanh(\cdot)$; since the softmax is $\frac{1}{2}$-Lipschitz and $\tanh$ is 1-Lipschitz,

$$\|d\|_2 \le \tfrac{1}{2}\, \alpha_\lambda\, \|W_\lambda\|_2\, \|x_q - x'_q\|_2. \tag{39}$$

Adding the bounds on the two sums gives equation 30. □

The constants $L_{r,a}$ and $L_{\mathcal{A}}$ of Lemma 2 are finite because of three design choices of §2 and the finiteness of the grid: the floor $\sigma_{\min}$ keeps the spread away from zero, the cap M bounds the displacement, the offset τ_0 keeps the temperature away from zero, and δ_a ranges over a finite set. Removing any one of them makes the corresponding bound in equation 32, and hence $L_{r,a}$, diverge.

Proposition 2 (Steered SSoG is Lipschitz on bounded sets)**.** *Let $\rho > 0$, let X, W_V^h, W_V and W_O be as in Proposition 1, and let the atom parameters of every head h be steered as in equation 9, under the assumptions of Lemma 2, so that the effective kernel $\mathcal{A}^h(X) \in \mathbb{R}^{(N-1)\times(N-1)}$ has rows $\mathcal{A}^h(x_q)$ as in that lemma. For each head h, atom r and axis a, let $\Theta_{r,a}^h$ be the set $\Theta_{r,a}$ of Lemma 2 for head h, and let*

$$\bar{\kappa}_{r,a}^h = \frac{\max_{s_a} \sum_{q_a} \sup_{\theta\in\Theta_{r,a}^h} g_\theta(q_a - s_a)}{\min_{q_a} \inf_{\theta\in\Theta_{r,a}^h} \sum_{s_a} g_\theta(q_a - s_a)}, \tag{40}$$

where g_θ denotes the kernel $g_{r,a}$ of head h evaluated at $\theta = (\mu, \sigma)$. Then, for all X, X' with $\|X\|_F, \|X'\|_F \le \rho$,

$$\|\mathrm{SSoG}(X) - \mathrm{SSoG}(X')\|_F \le \|W_V\|_2 \|W_O\|_2 \Big(\bar{K}_{\mathcal{A}} + \max_h L_{\mathcal{A}}^h\, \rho\Big) \|X - X'\|_F, \tag{41}$$

where

$$\bar{K}_{\mathcal{A}} := \max_h \sqrt{\tfrac{1}{N-1} + \textstyle\sum_r \bar{\kappa}_{r,y}^h \bar{\kappa}_{r,x}^h} \tag{42}$$

and $L_{\mathcal{A}}^h$ is the constant equation 30 for head h. In particular, the Lipschitz bound on the ball of radius ρ grows linearly in ρ.

Proof. With the class token last, as in the proof of Proposition 1, let $M^h(X) = \begin{pmatrix} \mathcal{A}^h(X) & 0 \\ \frac{1}{N-1}\mathbf{1}^\top & 0 \end{pmatrix}$. Denoting by

$$Z^h(X) = M^h(X) X W_V^h \in \mathbb{R}^{N\times D^h}$$

the output of head h, the SSoG output is obtained by concatenating the heads and applying the output projection,

$$\mathrm{SSoG}(X) = \big(Z^1(X) \,\|\, \cdots \,\|\, Z^H(X)\big) W_O.$$

Since the class row of M^h does not depend on X,

$$Z^h(X) - Z^h(X') = M^h(X)(X - X')W_V^h + \begin{pmatrix} \big(\mathcal{A}^h(X) - \mathcal{A}^h(X')\big) X'_{\mathrm{P}} W_V^h \\ 0 \end{pmatrix},$$

where X'_{P} collects the patch rows of X'. Arguing over the heads as in the proof of Proposition 1 and using the triangle inequality, the first terms contribute at most:

$$\|W_V\|_2 \|W_O\|_2 \max_h \|M^h(X)\|_2\, \|X - X'\|_F$$

and, since $\|X'_{\mathrm{P}} W_V\|_F \le \|W_V\|_2 \|X'\|_F$, the second at most:

$$\|W_V\|_2 \|W_O\|_2 \max_h \|\mathcal{A}^h(X) - \mathcal{A}^h(X')\|_2\, \|X'\|_F.$$

By Lemma 2(i), every steered parameter lies in $\Theta_{r,a}^h$, a compact set on which g_θ is continuous and positive on the finite grid, so $\bar{\kappa}_{r,a}^h < \infty$. Every row of $\mathcal{A}^h(X)$ is a probability vector, so $\|\mathcal{A}^h(X)\|_\infty = 1$. Each entry of a row of $A_{r,a}^h$ is at most

$$\sup_{\theta\in\Theta_{r,a}^h} g_\theta(q_a - s_a) \big/ \min_{q_a} \inf_{\theta\in\Theta_{r,a}^h} \sum_{s_a} g_\theta(q_a - s_a)$$

and $\lambda_r^h \le 1$ since λ^h is a softmax, so the column of $\mathcal{A}^h(X)$ indexed by $s = (s_y, s_x)$ sums to at most $\sum_r \bar{\kappa}_{r,y}^h \bar{\kappa}_{r,x}^h$. Hence

$$\|\mathcal{A}^h(X)\|_2 \le \big(\|\mathcal{A}^h(X)\|_1 \|\mathcal{A}^h(X)\|_\infty\big)^{1/2} \le \Big(\sum_r \bar{\kappa}_{r,y}^h \bar{\kappa}_{r,x}^h\Big)^{1/2}$$

and, exactly as in Proposition 1,

$$\|M^h(X)\|_2 \le \big(\tfrac{1}{N-1} + \sum_r \bar\kappa^h_{r,y}\bar\kappa^h_{r,x}\big)^{1/2},$$

so that $\max_h \|M^h(X)\|_2 \le \bar K_{\mathcal{A}}$. By Lemma 2(ii),

$$\|\mathcal{A}^h(X)-\mathcal{A}^h(X')\|_2 \le \|\mathcal{A}^h(X)-\mathcal{A}^h(X')\|_F = \big(\sum_q \|\mathcal{A}^h(x_q)-\mathcal{A}^h(x'_q)\|_2^2\big)^{1/2} \le L^h_{\mathcal{A}}\|X-X'\|_F.$$

Combining the two contributions,

$$\|\mathrm{SSoG}(X) - \mathrm{SSoG}(X')\|_F \le \|W_V\|_2\|W_O\|_2\Big(\bar K_{\mathcal{A}} + \max_h L^h_{\mathcal{A}}\,\|X'\|_F\Big)\|X - X'\|_F,$$

and $\|X'\|_F \le \rho$ gives equation 41. □

Remark. The dependence on ρ in equation 41 cannot be removed in general. For non-degenerate steering, fix a query position q and its token x_q, and scale the remaining patch tokens by a factor c: a perturbation of x_q changes the kernel row $\mathcal{A}^h(x_q)$ at a rate that does not depend on c, while the values it multiplies are of size $O(c)$, so the directional derivative of steered SSoG grows linearly in c. When the steering does not act, for instance when $W_\mu = W_\sigma = W_\lambda = 0$, the operator reduces to fixed SSoG and Proposition 1 applies.

The preceding Remark has the same structure as the obstruction that Kim et al. (2021) identify for dot-product attention, namely a data-dependent mixing matrix multiplying a data-dependent value matrix, but not the same rate. Attention logits are inner products between two token features, so the Lipschitz constant of the mixing matrix itself grows with $\|X\|$ and the available upper bounds on a ball of radius ρ grow as ρ^2; the steered kernel depends only on the query token and only through a bounded $\tanh$, so $L^h_{\mathcal{A}}$ does not depend on X and the bound of Proposition 2 grows as ρ. Relative to dot-product attention, steering thus lowers the exponent of this bound from two to one, and the fixed variant removes the dependence on ρ altogether. We therefore regard Proposition 1 as a further reason to prefer the fixed operator, alongside the accuracy results of §3.2.