RETHINKING PAIRWISE TOKEN INTERACTION IN SPIKING TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking Transformers inherit token interaction mechanisms from conventional Transformers, yet their sparse binary representations fundamentally alter how token-to-token communication is established. In particular, spike-based query–key matching produces highly sparse and input-dependent interaction patterns, coupling information propagation to the instantaneous availability of matching spike events. This motivates a different interaction paradigm in which long-range communication does not rely solely on pairwise spike coincidence. We therefore propose **Gated Spike Axial Propagation (GSAP)**, a spike-native token interaction mechanism that decouples information propagation from context selection. Instead of directly determining communication through query–key matching, GSAP first propagates spike-based context along the horizontal and vertical axes, allowing information to reach distant tokens through structured sequential propagation. A receiver-conditioned gate then determines how much of the propagated context is incorporated at each token, while a lightweight local pathway preserves fine-grained neighborhood information. In this way, GSAP reformulates token interaction as a propagate-then-select process, enabling structured long-range communication while retaining the sparse event-driven nature of spiking representations.

1 INTRODUCTION

Spiking Neural Networks (SNNs) provide a biologically inspired paradigm for energy-efficient computation by representing information through sparse spike events and temporally evolving neuronal states (Maass, 1997; Gilra & Gerstner, 2017). Their event-driven nature offers promising opportunities for neuromorphic hardware and resource-constrained intelligent systems (Roy et al., 2019). However, the discontinuous dynamics of spike generation introduce significant optimization challenges for deep SNNs. Recent advances in surrogate-gradient learning, learnable neuronal dynamics, and residual spiking architectures have substantially improved the trainability and scalability of SNNs (Neftci et al., 2019; Fang et al., 2021b;a; Eshraghian et al., 2023), enabling increasingly expressive spiking models for visual recognition.

Transformers achieve powerful representation capability by modeling long-range dependencies through dynamic token interaction (Vaswani et al., 2017; Dosovitskiy et al., 2020). To combine the interaction capability of Transformers with spike-driven computation, Spikformer introduced Spiking Self-Attention (SSA), which adapts self-attention to binary spike representations by removing Softmax and operating on spike-based queries, keys, and values (Zhou et al., 2022). Following this direction, subsequent studies have explored improved spike-driven attention mechanisms, hierarchical architectures, temporal representation learning, and efficient deployment strategies (Yao et al., 2023a; 2024; Zhou et al., 2026; 2024a; Shi et al., 2024; Shen et al., 2024; Lee et al., 2025; Liu et al., 2024). These advances have established Spiking Transformers as a promising architecture family that combines Transformer-style representation learning with event-driven computation.

Despite these advances, existing Spiking Transformers largely inherit the token interaction principle of conventional Transformers, where information exchange is established through explicit pairwise relationships between tokens. While effective for dense-valued representations, this interaction paradigm introduces a potential mismatch with sparse spike-based representations. In Spiking Attention, the contribution of a source token to a receiving token is determined by the coincidence

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.

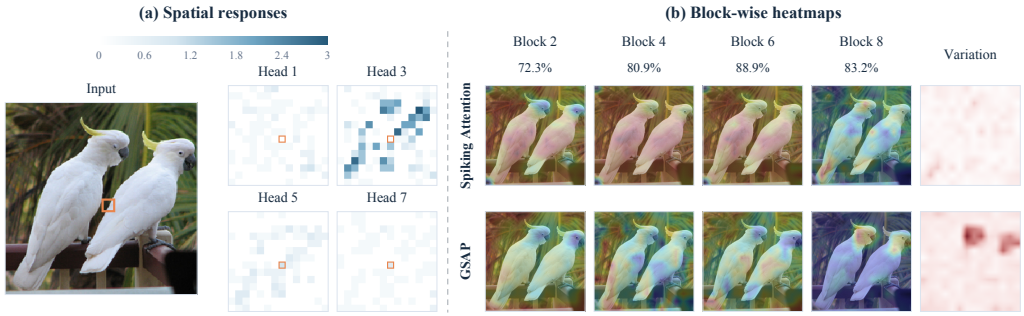


Figure 1: (a) Spatial response maps for selected heads of Spiking Self-Attention (SSA) in an ImageNet-1K 8-768 model with $T = 4$. (b) Block-wise response visualizations for SSA and GSAP. The percentages annotate the fractions of tokens without Q–K matches over four time steps for the displayed example. *Variation* visualizes changes across blocks.

between their query and key spike patterns. Therefore, communication pathways are explicitly constructed through discrete matching events rather than continuously available interaction patterns. Under sparse spike activity, effective information exchange becomes dependent on whether compatible spike patterns exist between source and receiver tokens (Guo et al., 2025; Xiao et al., 2025; Zhao et al., 2026).

This observation motivates revisiting token interaction in Spiking Transformers from the perspective of spike-based communication. Instead of explicitly constructing source–receiver affinities, we argue that token interaction can be decomposed into two complementary functions: *global context interaction* and *receiver-conditioned selection*. The former determines how information is exchanged across spatial locations, whereas the latter controls how each token utilizes the received contextual information according to its current state.

Figure 1 illustrates the sparse interaction patterns observed in SSA. The visualization shows that token communication is strongly influenced by discrete spike matching events, thereby impairing the ability to capture global dependencies and motivating the exploration of alternative interaction mechanisms beyond explicit pairwise affinity construction. Based on this perspective, we propose **Gated Spike Axial Propagation (GSAP)**, a spike-native token interaction mechanism for Spiking Transformers. GSAP replaces explicit QKV matching with spike-driven global context interaction and receiver-conditioned selection. By decoupling communication from context selection, GSAP enables flexible long-range information exchange while preserving the temporal characteristics of spiking computation.

We evaluate GSAP on static-image classification, event-based recognition, and semantic segmentation tasks using multiple Spiking Transformer backbones. Experimental results demonstrate consistent improvements over existing interaction mechanisms while maintaining competitive computational efficiency. Extensive ablation studies further analyze the roles of global context interaction and receiver-conditioned selection, providing insights into how spike-driven token interaction can be redesigned. Our contributions are summarized as follows:

- We identify a mismatch between pairwise query–key matching and sparse spike representations: source-to-receiver communication relies on discrete spike coincidences, limiting the interactions available for global feature modeling.
- We propose **GSAP**, a spike-native interaction module that decouples spatial communication from receiver-conditioned selection. Sequential horizontal–vertical spike propagation builds long-range structural paths, while a receiver-side gate selects global context before fusion with local and identity features.
- Token-perturbation analyses reveal long-range communication through sequential axial propagation alongside spatially concentrated output sensitivity. Component ablations quantify the contributions of axial propagation, receiver-conditioned selection, and local refinement to recognition performance.

2 Related Work

2.1 Spiking Transformer Architectures

Spiking Transformers aim to combine the representation capability of Transformers with the sparse and event-driven computation of SNNs. Spikformer first established this paradigm by introducing Spiking Self-Attention (SSA) and spike-based Transformer representations (Zhou et al., 2022). Subsequent studies investigated more spike-compatible architectures by redesigning computational pathways, residual structures, and feature extraction mechanisms. Spike-driven Transformer and Spikingformer reduced non-spike operations and improved spike-driven execution, while Meta-SpikeFormer further extended Spiking Transformers toward general-purpose visual backbones (Yao et al., 2023a; 2024).

Beyond single-stage architectures, recent works explored hierarchical and multi-scale Spiking Transformers. Spikformer V2 introduced a spiking convolutional stem for improved visual representation, while QKFormer and SpikingResformer developed hierarchical designs with multi-stage feature processing (Zhou et al., 2024a; Shi et al., 2024). Meanwhile, Spiking Transformers have been extended to dense prediction and other vision tasks, including semantic segmentation and point-cloud understanding (Lei et al., 2025; Wu et al., 2025; Lu et al., 2026).

These studies mainly focus on improving backbone scalability, representation capability, and task generalization. In contrast, we investigate the fundamental token interaction mechanism within Spiking Transformers.

2.2 Token Interaction in Spiking Transformers

Token interaction is a core component of Transformer architectures. Existing Spiking Transformers generally inherit attention-based interaction from conventional Transformers and adapt it to binary spike representations. Spikformer introduced SSA by replacing continuous attention computation with spike-form query, key, and value representations (Zhou et al., 2022). Subsequent approaches further enhanced spatial and temporal interaction through improved attention structures, including temporal-aware interaction mechanisms and spatiotemporal attention designs (Shen et al., 2024; Lee et al., 2025; Lu et al., 2025).

However, the sparse nature of spike representations may weaken the relational information captured by explicit query–key matching. Recent works therefore attempt to improve spike-based attention by enriching or recovering token relationships. α-XNOR incorporates non-spike matches into binary similarity computation to alleviate information loss (Xiao et al., 2025); SLI-ACF introduces additional local pathways to recover neighborhood interactions (Zhao et al., 2026); and HAST and SQKformer enhance attention through composite tokens, channel-wise interaction, or multi-path modeling (Fan et al., 2025; Chen et al., 2026).

Different from these approaches, which preserve the fundamental source–receiver matching formulation and improve its effectiveness, GSAP revisits the interaction principle itself. Instead of explicitly computing pairwise affinities, GSAP establishes token communication through spike propagation and performs receiver-conditioned context selection.

3 Proposed Method

We introduce **Gated Spike Axial Propagation (GSAP)**, a spike-native token interaction mechanism for visual Spiking Transformers. GSAP separates spatial communication from receiver-conditioned context selection: sequential horizontal and vertical propagation aggregates information across spatial locations, while a receiver-side gate regulates the utilization of the aggregated context. Spiking neurons maintain temporal membrane states within these computations, connecting spatial interaction with temporal processing. As illustrated in Fig. 2(c), the selected context is fused with local features and the input spikes to update token representations. We first review the neuronal dynamics and attention formulation that motivate this design, then describe the individual GSAP components.

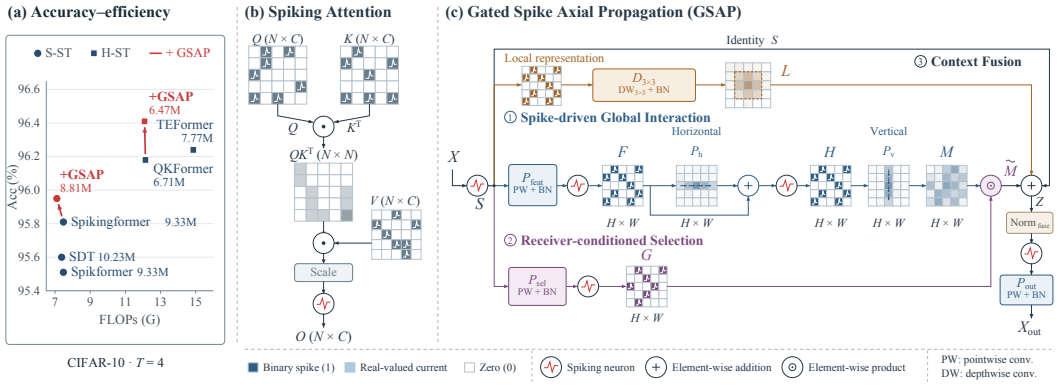


Figure 2: Overview of GSAP as a spike-native token interaction mechanism for Spiking Transformers. (a) Accuracy–efficiency comparison with existing approaches. (b) Existing Spiking Attention establishes token interaction through explicit pairwise query–key matching. (c) GSAP replaces pairwise matching with propagation-based spike communication and receiver-conditioned context selection.

3.1 Background and Motivation: Spiking Transformer

Spiking Transformers combine the temporal dynamics of spiking neurons with the token interaction capability of Transformers. A typical Spiking Transformer block mainly consists of spiking neuron dynamics and spike-driven token interaction. We briefly introduce these components and discuss the limitation arising from their combination.

Spiking Neuron Dynamics. Spiking neurons provide the temporal computation mechanism in SNNs. Following the leaky integrate-and-fire (LIF) model, the membrane state $U[t]$ and spike generation $S[t]$ are formulated as

$$\begin{aligned} U[t] &= \lambda U[t-1] + I[t], \\ S[t] &= H(U[t] - V_{\text{th}}), \end{aligned} \tag{1}$$

where $U[t]$ denotes the membrane potential, $I[t]$ is the input current, λ represents the leakage factor, and $H(\cdot)$ denotes the Heaviside function. The membrane state provides temporal memory across simulation steps and enables spike-driven information processing.

Spiking Self-Attention. To introduce Transformer-style token interaction into SNNs, Spikformer proposed Spiking Self-Attention (SSA), where query, key, and value representations are encoded as binary spike sequences. For one attention head at time step t, SSA performs

$$\mathbf{J}^t = s\left(\mathbf{Q}^t(\mathbf{K}^t)^\top\right)\mathbf{V}^t, \tag{2}$$

where $\mathbf{Q}^t, \mathbf{K}^t, \mathbf{V}^t \in \{0,1\}^{N\times d}$, N denotes the number of tokens, and d is the head dimension. The interaction strength between tokens is determined by query–key spike coincidence, which explicitly constructs communication relationships before context aggregation.

Although SSA enables content-dependent token interaction, it couples communication and context selection through sparse spike matching. GSAP instead decouples these two functions by using spike-driven global context propagation for communication and receiver-conditioned selection for adaptive context utilization.

3.2 Gated Spike Axial Propagation

As shown in Fig. 2(c), GSAP processes the input spike representation through parallel propagation, selection, and local pathways. The propagation pathway projects the input into spike features and performs horizontal aggregation, spike re-encoding, and vertical aggregation in sequence to obtain

the contextual representation M. The selection pathway generates a binary gate G from the receiver-side input and its neuronal state, yielding the selected context $\widetilde{M} = G \odot M$. The local pathway extracts neighborhood features L, which are added to $\widetilde{M}$ and the identity input S. Normalization, spike re-encoding, and output projection then produce the updated representation. The following paragraphs formalize these computations.

Spike-driven Global Interaction. GSAP establishes long-range token interaction through spike-driven global context aggregation. Given an input spike representation $\mathbf{S} \in \{0,1\}^{T\times B\times C\times H\times W}$, GSAP first projects the input spikes into spike features and subsequently aggregates contextual information along two orthogonal spatial directions:

$$\begin{aligned} \mathbf{F} &= \mathrm{SN}_{\mathrm{feat}}\left(\mathcal{P}_{\mathrm{feat}}(\mathbf{S})\right), \\ \mathbf{H} &= \mathrm{SN}_{h}\left(\mathcal{P}_{h}(\mathbf{F}) + \mathbf{F}\right), \\ \mathbf{M} &= \mathcal{P}_{v}(\mathbf{H}), \end{aligned} \tag{3}$$

where $\mathcal{P}_{\mathrm{feat}}(\cdot)$ denotes a pointwise projection, and $\mathcal{P}_{h}(\cdot)$ and $\mathcal{P}_{v}(\cdot)$ represent horizontal and vertical spatial aggregation operators, respectively. The intermediate spiking neuron re-encodes the horizontally aggregated responses and preserves temporal membrane states, allowing global context aggregation to be modulated by spike-driven temporal dynamics. Through sequential axial aggregation, information from distant spatial locations can contribute to each receiving token without explicitly constructing source–receiver affinities. Different from attention, where the interaction topology is determined by query–key similarity before aggregation, GSAP establishes token communication through learned spatial aggregation dynamics and neuronal temporal states.

Receiver-conditioned Selection. After global context aggregation, the received information does not necessarily contribute equally to every token. GSAP therefore introduces a receiver-conditioned selection mechanism, where each token determines how the aggregated context should be integrated according to its own representation and temporal neuronal state.

A parallel branch generates a receiver-dependent selection signal:

$$\begin{aligned} \mathbf{G} &= \mathrm{SN}_{\mathrm{sel}}\left(\mathcal{P}_{\mathrm{sel}}(\mathbf{S})\right), \\ \widetilde{\mathbf{M}} &= \mathbf{G} \odot \mathbf{M}, \end{aligned} \tag{4}$$

where $\mathcal{P}_{\mathrm{sel}}(\cdot)$ denotes a lightweight projection, $\mathrm{SN}_{\mathrm{sel}}$ represents a spiking neuron, and $\odot$ denotes element-wise multiplication. The selection signal has the same spatial and channel dimensions as the aggregated context $\mathbf{M}$. Unlike source-dependent weighting mechanisms, the selection signal is generated from the receiver-side representation, allowing each token to regulate context utilization according to its current state.

Unlike attention-based interaction, where source–receiver relationships and context selection are jointly determined by affinity estimation, GSAP decouples communication from selection. Specifically, global propagation enables information exchange across tokens, while receiver-conditioned selection adaptively regulates the utilization of received context according to each token's representation and temporal state.

Context Fusion. While global context interaction enables long-range information exchange, local spatial structures remain important for visual representation learning. GSAP therefore preserves local information through a lightweight local representation pathway:

$$\begin{aligned} \mathbf{L} &= \mathcal{D}_{3\times3}(\mathbf{S}), \\ \mathbf{Z} &= \mathbf{S} + \mathbf{L} + \widetilde{\mathbf{M}}, \\ \mathbf{X}_{out} &= \mathcal{P}_{\mathrm{out}}\left(\mathrm{SN}_{\mathrm{fuse}}\left(\mathrm{Norm}_{\mathrm{fuse}}(\mathbf{Z})\right)\right). \end{aligned} \tag{5}$$

where $\mathcal{D}_{3\times3}$ denotes a depthwise convolution operating on spike representations and $\mathrm{Norm}_{\mathrm{fuse}}$ denotes the normalization operation before spike re-encoding. The fused representation combines the original spike state, local spatial structures, and receiver-selected global context.

Through this fusion process, GSAP integrates local refinement and global context propagation to generate updated token representations. Different from attention-based interaction that relies on explicit pairwise affinity estimation, GSAP forms implicit communication pathways through spike-driven propagation and receiver-conditioned message selection.

4 EXPERIMENTS

We evaluate GSAP as a spike-native token interaction mechanism for Spiking Transformers across different architectures and tasks. Specifically, we conduct experiments on image classification, event-based recognition, and semantic segmentation. In all experiments, GSAP is integrated by replacing the original token interaction modules while keeping the remaining backbone components unchanged, ensuring a fair comparison with existing Spiking Transformer architectures. Dataset, implementation, and training details are provided in Appendices C.1 and C.2, with energy and computational complexity analysis in Appendix E.

4.1 IMAGE CLASSIFICATION

We first evaluate GSAP on image classification tasks, which serve as the standard benchmark for Spiking Transformers. Experiments are conducted on ImageNet-1K, CIFAR-10, CIFAR-100, and TinyImageNet to investigate the effectiveness of GSAP under different model scales and architectural configurations (Deng et al., 2009; Krizhevsky et al., 2010).

As shown in Table 1, GSAP consistently improves the performance of Spiking Transformer backbones across different model scales. Specifically, GSAP improves Spikingformer-8-384, Spikingformer-8-512, and Spikingformer-8-768 from 72.45%, 74.79%, and 75.85% to 73.29%,

Table 1: ImageNet-1K classification results. * denotes an input resolution of 288×288; all other models use 224×224. GSAP achieves significantly better performance than the baseline across nearly all model sizes while using fewer parameters.

Methods	Type	Architecture	Params (M)	Time Steps	Acc.(%)
ResNet (He et al., 2016)	ANN	Res-CNN-104	54.21	1	76.87
ViT (Dosovitskiy et al., 2020)	ANN	ViT-B/16	86.59	1	77.90
DeiT (Touvron et al., 2021)	ANN	DeiT-B	86.59	1	81.80
		DeiT-B*	86.59	1	83.10
Spiking ResNet (Hu et al., 2021)	SNN	Spiking-ResNet-34	21.79	350	71.61
		Spiking-ResNet-51	25.56	350	72.75
SEW ResNet (Fang et al., 2021a)	SNN	SEW-ResNet-34	21.79	4	67.04
		SEW-ResNet-101	44.55	4	68.76
		SEW-ResNet-152	60.19	4	69.26
Att MS-ResNet (Yao et al., 2022)	SNN	Att-MS-ResNet-18	11.69	6	63.10
		Att-MS-ResNet-34	21.80	6	69.42
		Att-MS-ResNet-104	77.28	5	76.02
Spikformer (Zhou et al., 2022)	SNN	Spikformer-8-384	16.81	4	70.24
		Spikformer-8-512	29.68	4	73.38
		Spikformer-8-768	66.34	4	74.81
SDT (Yao et al., 2023a)	SNN	SDT-8-384	16.81	4	72.28
		SDT-8-512	29.68	4	74.57
		SDT-8-768*	66.34	4	77.07
Spikingformer (Zhou et al., 2026)	SNN	Spikingformer-8-384	16.81	4	72.45
		Spikingformer-8-512	29.68	4	74.79
		Spikingformer-8-768	66.34	4	75.85
α-SSA (Xiao et al., 2025)	SNN	α-SSA-ViT-8-384	16.80	4	71.36
		α-SSA-ViT-8-512	29.70	4	75.39
SEMM (Zhou et al., 2024b)	SNN	Spikingformer-8-384	16.05	4	73.58
		Spikingformer-8-512	28.22	4	76.03
GSAP(ours)	SNN	Spikingformer-8-384	15.85	4	**73.29**
		Spikingformer-8-512	27.87	4	**76.07**
		Spikingformer-8-768	62.04	4	**78.19**

Table 2: Classification results on CIFAR-10, CIFAR-100, and TinyImageNet. S-ST denotes single-stage Spiking Transformer, while H-ST denotes hierarchical Spiking Transformer.

Method	Architecture	CIFAR10			CIFAR100			TinyImageNet†		
		Params (M)	T	Acc. (%)	Params (M)	T	Acc. (%)	Params (M)	T	Acc. (%)
Spikformer (Zhou et al., 2022)	S-ST	9.33	4	95.51	9.37	4	78.21	9.40	4	64.78
SDT (Yao et al., 2023a)	S-ST	10.23	4	95.60	10.28	4	78.40	9.39	4	66.03
RevSFormer (Zhang & Zhang, 2024)	S-ST	9.33	4	95.34	9.37	4	79.04	-	-	-
STAttn+Spikf (Lee et al., 2025)	S-ST	9.33	4	94.36	-	4	75.85	-	-	-
Spikingformer (Zhou et al., 2026)	S-ST	9.33	4	95.81	9.36	4	79.21	9.39	4	65.13
GSAP (ours)	S-ST	8.81	4	**95.95**	8.84	4	**80.21**	8.92	4	**66.84**
QKFormer (Zhou et al., 2024a)	H-ST	6.71	4	96.18	6.74	4	81.15	6.78	4	64.07
STAttn+QKF (Lee et al., 2025)	H-ST	6.71	4	95.35	6.74	4	80.20	-	-	-
TEFormer (Shen et al., 2026)	H-ST	7.77	4	96.24	7.80	4	79.84	-	-	-
GSAP (ours)	H-ST	6.47	4	**96.41**	6.51	4	**81.64**	6.60	4	**66.65**

76.07%, and 78.19%, respectively. Meanwhile, GSAP reduces the corresponding parameter counts from 16.81M, 29.68M, and 66.34M to 15.85M, 27.87M, and 62.04M. Interestingly, the performance gain becomes more pronounced with larger model scales, suggesting that propagation-based interaction can better exploit large-scale token communication. These results demonstrate that effective token interaction in Spiking Transformers can be achieved through propagation-based communication without introducing additional parameters.

To further investigate whether GSAP is coupled with a specific Spiking Transformer architecture, we evaluate it on both single-stage and hierarchical Spiking Transformer families. As shown in Table 2, GSAP consistently improves different architectural variants under comparable settings. For single-stage Spiking Transformers, GSAP improves Spikingformer from 79.21% to 80.21% on CIFAR-100 while reducing the parameter count from 9.36M to 8.84M. For hierarchical Spiking Transformers, GSAP improves QKFormer from 81.15% to 81.64% with fewer parameters. Similar improvements are observed on TinyImageNet, where GSAP achieves 66.84% and 66.65% for single-stage and hierarchical architectures, respectively.

Overall, these results demonstrate that GSAP provides a general spike-native token interaction mechanism that is not tailored to a specific backbone, but can be effectively applied to diverse Spiking Transformer architectures.

4.2 Event-based Datasets

Beyond frame-based image inputs, we further evaluate GSAP on event-based tasks to investigate its compatibility with sparse and temporally structured spike-driven data. Experiments are conducted on CIFAR10-DVS, N-Caltech101, and UCF101-DVS (Li et al., 2017; Orchard et al., 2015; Bi et al., 2020).

As shown in Table 3, replacing the original token interaction modules with GSAP consistently improves recognition performance across different event-based datasets. In particular, GSAP improves the accuracy on N-Caltech101 from 84.45% to 85.54%. This result indicates that propagation-based interaction is effective for processing sparse temporal event streams, where information is naturally distributed across time and accumulated through dynamic interactions. These observations further support the motivation of GSAP as a spike-native interaction mechanism.

Table 3: Classification results on CIFAR10-DVS, N-Caltech101, and UCF101-DVS.

Method	CIFAR10-DVS			N-Caltech101			UCF101-DVS†		
	Params (M)	T	Acc. (%)	Params (M)	T	Acc. (%)	Params (M)	T	Acc. (%)
SALT (Kim & Panda, 2021)	-	20	67.1	-	20	55.0	-	-	-
Res-SNN18+RM (Yao et al., 2023b)	-	-	-	-	30 × 10	81.2	-	8	58.5
NDA (Li et al., 2022)	-	10	81.5	-	10	79.5	-	-	-
EventMix+ResNet (Shen et al., 2023)	-	10	78.0	-	10	78.6	-	-	-
Spikformer (Zhou et al., 2022)	2.59	16	80.60	2.61	10	83.96	2.61	10	64.83
SDT (Yao et al., 2023a)	2.57	16	80.00	2.59	10	83.23	2.59	10	65.23
TIM (Shen et al., 2024)	2.57	16	81.60	2.59	10	79.00	2.59	10	63.80
Spikingformer (Zhou et al., 2026)	2.57	16	81.30	2.59	10	84.45	2.59	10	65.56
GSAP (ours)	2.46	16	**81.80**	2.48	10	**85.54**	2.48	10	**65.64**

4.3 Semantic Segmentation

We further investigate whether GSAP can benefit dense prediction tasks, where effective spatial information exchange among tokens is essential for preserving fine-grained structures.

Table 4: Semantic segmentation results on ADE20K and PASCAL VOC. Each +GSAP row replaces the interaction modules of the preceding baseline.

Model	Dataset	mIoU	mAcc	aAcc
QKFormer	ADE20K	32.63	43.33	75.56
+GSAP		**36.90**	47.88	77.63
Spikingformer		31.64	41.85	74.99
+GSAP		**33.82**	43.63	76.49
QKFormer	VOC	60.76	90.44	73.73
+GSAP		**62.23**	91.13	72.59

As shown in Table 4, integrating GSAP into different Spiking Transformer backbones consistently improves semantic segmentation performance on ADE20K and PASCAL VOC (Zhou et al., 2017; Everingham et al., 2010). On ADE20K, GSAP improves QKFormer from 32.63% to 36.90% mIoU and improves Spikingformer from 31.64% to 33.82% mIoU. On PASCAL VOC, GSAP also achieves improvements over the corresponding baseline.

These results demonstrate that GSAP provides a generalizable spike-native token interaction mechanism across different architectures, input modalities, and prediction tasks.

5 Analysis & Discussion

5.1 Ablation Study

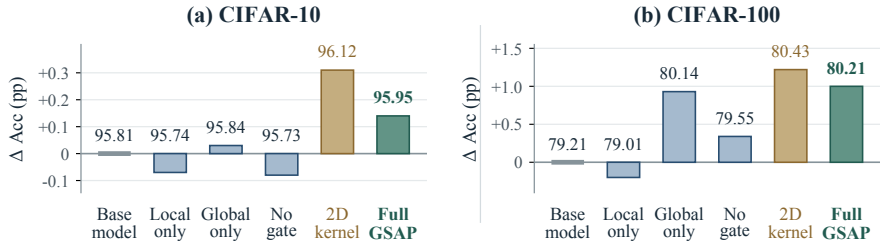


Figure 3: Ablation studies of GSAP on CIFAR-10 and CIFAR-100. Component ablations evaluate global propagation, receiver gating, and local refinement, while propagation-path ablations compare axial propagation with full two-dimensional propagation. The latter replaces both axial operators and the intermediate LIF neuron and row residual connection.

We conduct ablation studies on CIFAR-10 and CIFAR-100 using the Spikingformer backbone with $T = 4$ to investigate the contribution of different design choices in GSAP, including long-range propagation, receiver-conditioned message selection, and propagation structures.

Starting from the full GSAP model, we progressively remove components to analyze their effects. The complete model achieves 95.95% and 80.21% accuracy on CIFAR-10 and CIFAR-100, respectively. Removing the global propagation branch decreases accuracy by 0.21 and 1.20 percentage points, demonstrating the importance of long-range message aggregation. Removing the receiver-conditioned gate further leads to accuracy drops of 0.22 and 0.66 percentage points, even when both local and global propagation pathways remain available. This indicates that token interaction requires not only context aggregation, but also adaptive selection of propagated messages. Meanwhile, removing the local branch causes smaller but consistent drops of 0.11 and 0.07 percentage points, suggesting that local refinement provides complementary spatial information.

We further compare different propagation structures within GSAP. Replacing axial propagation with a full two-dimensional depthwise kernel improves accuracy by 0.17 and 0.22 percentage points on CIFAR-10 and CIFAR-100, respectively. However, this improvement introduces substantially higher parameter costs. Specifically, axial propagation requires only $2Ck$ spatial kernel parameters, compared with Ck^2 for a full two-dimensional kernel, reducing the propagation parameters by 86.7% when $k = 15$. Although full two-dimensional propagation provides stronger spatial interaction capability, axial propagation achieves a better balance between representation capability and model

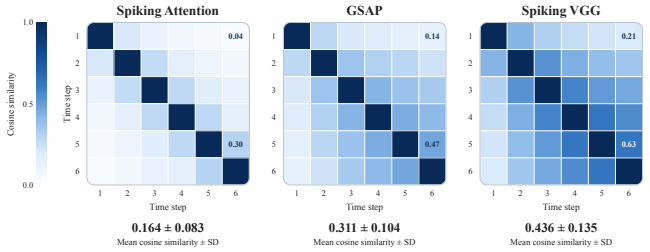
complexity, making it a more suitable design choice for GSAP. The corresponding ablation results are summarized in Fig. 3.

5.2 GSAP FOR ONLINE LEARNING

Beyond inference-time representation learning, token interaction also influences the temporal propagation of neural activity and learning signals during optimization. Unlike attention-based interaction that relies on instantaneous token affinity, GSAP introduces structured propagation across tokens, which may provide more consistent information flow under online learning settings (Kaiser et al., 2020).

We investigate this issue within an OTTT-style online learning framework (Xiao et al., 2022), where gradients combine temporally accumulated neural traces with current-step learning signals. In Spiking Attention, multiplicative terms involving sparse spike activations can suppress learning signals and cause gradient directions to fluctuate across time steps, hindering effective gradient accumulation. GSAP alleviates this bottleneck through structured spatial propagation that avoids reliance on pairwise spike matching. To isolate the contribution of propagation dynamics, we evaluate an ungated GSAP variant, which exhibits higher cross-step gradient cosine similarity and improved classification accuracy under OTTT. The corresponding analysis is provided in Appendix D.

Figure 4 compares the cross-step gradient cosine similarity and final task performance between GSAP and Spiking Attention under OTTT. The ungated GSAP variant achieves higher gradient similarity across time steps, indicating more consistent learning signals during online updates. Furthermore, under the same experimental settings as the Spiking Attention baseline, GSAP without receiver gating consistently improves performance across the evaluated OTTT variants and tasks. In particular, GSAP achieves a gain of more than 10 percentage points on CIFAR-100 under OTTT-A (Fig. 4). These results demonstrate that propagation-based token interaction **not only improves inference-time representation learning, but also provides more stable information flow for online optimization in Spiking Transformers.**



CIFAR-10			Acc. (%)
Model	**Steps**	**OTTT-A**	**OTTT-O**
Spiking Attention	6	82.51	89.30
GSAP	6	**83.97**	**91.19**
VGG11	6	92.07	92.57
CIFAR-100			Acc. (%)
Model	**Steps**	**OTTT-A**	**OTTT-O**
Spiking Attention	6	51.48	63.05
GSAP	6	**62.07**	**68.49**
VGG11	6	69.22	70.81

OTTT-A: Accumulate T-step gradients; update once.
OTTT-O: Update parameters at every time step.

Figure 4: Gradient consistency and task performance under OTTT-based online learning. Spiking Attention struggles to maintain consistent gradient directions across time steps, hindering gradient accumulation and effective parameter updates. GSAP denotes the ungated variant used to isolate the effect of propagation-based token interaction.

6 CONCLUSION

In this work, we revisit token interaction in Spiking Transformers from a spike-native perspective and propose Gated Spike Axial Propagation (GSAP). Instead of relying on explicit pairwise token affinity, GSAP enables token communication through structured spike propagation and receiver-conditioned message selection. Experiments across image classification, event-based recognition, and semantic segmentation demonstrate the effectiveness and generality of GSAP. Ablation studies reveal that long-range propagation, adaptive message selection, and local refinement contribute to token interaction. Token-perturbation analyses show that long-range propagation can coexist with spatially localized sensitivity. This highlights the value of combining broad spatial communication with selective context utilization in spiking representations. Our findings suggest that Spiking Transformers may benefit from interaction mechanisms designed around the temporal and sparse properties of spiking computation rather than directly adapting conventional attention mechanisms.

AI USE STATEMENT

Generative AI tools were used solely for language translation and final-stage polishing to improve the clarity, fluency, and readability of the manuscript. They were not used for research ideation, methodology design, experimental design, data analysis, result interpretation, or the formulation of scientific claims. The authors take full responsibility for the content of the manuscript.

REPRODUCIBILITY STATEMENT

Detailed experimental configurations, including implementation and training settings, are provided in the Appendix. The complete experimental code necessary to reproduce the results is included in the supplementary material submitted with this paper.

REFERENCES

Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE transactions on image processing*, 29:9084–9098, 2020.

Yunhua Chen, Zequan Xie, Jinyu Zhong, Pinghua Chen, and Jinsheng Xiao. Sqkformer: Spiking sparse qkformer with adaptive batch normalization for membrane potential. *Neurocomputing*, pp. 132666, 2026.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Jason K Eshraghian, Max Ward, Emre O Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Xiongfei Fan, Hong Zhang, and Yu Zhang. Hybrid attention spike transformer. *IET Cyber-Systems and Robotics*, 7(1):e70010, 2025.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in neural information processing systems*, 34:21056–21069, 2021a.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2641–2651. IEEE, 2021b.

Aditya Gilra and Wulfram Gerstner. Predicting non-linear dynamics by stable local learning in a recurrent spiking neural network. *Elife*, 6:e28295, 2017.

Yufei Guo, Xiaode Liu, Yuanpei Chen, Weihang Peng, Yuhan Zhang, and Zhe Ma. Spiking transformer: Introducing accurate addition-only spiking self-attention for transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24398–24408. IEEE, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Yangfan Hu, Huajin Tang, and Gang Pan. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5200–5205, 2021.

Jacques Kaiser, Hesham Mostafa, and Emre Neftci. Synaptic plasticity dynamics for deep continuous local learning (decolle). *Frontiers in Neuroscience*, 14:424, 2020.

Jaehyun Kim, Heesu Kim, Subin Huh, Jinho Lee, and Kiyoung Choi. Deep neural networks with weighted spikes. *Neurocomputing*, 311:373–386, 2018.

Youngeun Kim and Priyadarshini Panda. Optimizing deeper spiking neural networks for dynamic vision sensing. *Neural Networks*, 144:686–698, 2021.

Alex Krizhevsky, Geoff Hinton, et al. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010.

Donghyun Lee, Yuhang Li, Youngeun Kim, Shiting Xiao, and Priyadarshini Panda. Spiking transformer with spatial-temporal attention. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13948–13958. IEEE, 2025.

Zhenxin Lei, Man Yao, Jiakui Hu, Xinhao Luo, Yanye Lu, Bo Xu, and Guoqi Li. Spike2former: Efficient spiking transformer for high-performance image segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 1364–1372, 2025.

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:309, 2017.

Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks. In *European conference on computer vision*, pp. 631–649. Springer, 2022.

Yue Liu, Shanlin Xiao, Bo Li, and Zhiyi Yu. Sparsespikformer: A co-design framework for token and weight pruning in spiking transformer. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6410–6414. IEEE, 2024.

Chengzhuo Lu, Huilin Du, Wenjie Wei, Qian Sun, Yuchen Wang, Dingyi Zeng, Wenyu Chen, Malu Zhang, and Yang Yang. Estsformer: Efficient spatio-temporal spiking transformer. *Neural Networks*, 191:107786, 2025.

Yijie Lu, Zhiyi Pan, Renrui Zhang, Yanhao Jia, Ronggang Wang, and Zhaokun Zhou. Spiking discrepancy transformer for point cloud analysis. In *International Conference on Learning Representations*, volume 2026, pp. 109747–109769, 2026.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Hesham Mostafa. Supervised learning based on temporal coding in spiking neural networks. *IEEE transactions on neural networks and learning systems*, 29(7):3227–3235, 2017.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

Nitin Rathi and Kaushik Roy. Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):3174–3182, 2021.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017.

Guobin Shen, Dongcheng Zhao, and Yi Zeng. Eventmix: An efficient data augmentation strategy for event-based learning. *Information Sciences*, 644:119170, 2023.

Sicheng Shen, Dongcheng Zhao, Guobin Shen, and Yi Zeng. Tim: an efficient temporal interaction module for spiking transformer. *arXiv preprint arXiv:2401.11687*, 2024.

Sicheng Shen, Mingyang Lv, Bing Han, Dongcheng Zhao, Guobin Shen, Feifei Zhao, and Yi Zeng. Teformer: Structured bidirectional temporal enhancement modeling in spiking transformers. *arXiv preprint arXiv:2601.18274*, 2026.

Xinyu Shi, Zecheng Hao, and Zhaofei Yu. Spikingresformer: Bridging resnet and vision transformer in spiking neural networks. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5610–5619. IEEE, 2024.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

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

Peixi Wu, Bosong Chai, Hebei Li, Menghua Zheng, Yansong Peng, Zeyu Wang, Xuan Nie, Yueyi Zhang, and Xiaoyan Sun. Spiking point transformer for point cloud classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21563–21571, 2025.

Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, and Zhouchen Lin. Online training through time for spiking neural networks. *Advances in neural information processing systems*, 35:20717–20730, 2022.

Yichen Xiao, Shuai Wang, Dehao Zhang, Wenjie Wei, Yimeng Shan, Xiaoli Liu, Yulin Jiang, and Malu Zhang. Rethinking spiking self-attention mechanism: Implementing α-xnor similarity calculation in spiking transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5444–5454. IEEE, 2025.

Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *arXiv preprint arXiv:2209.13929*, 2022.

Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. *Advances in neural information processing systems*, 36:64043–64058, 2023a.

Man Yao, Hengyu Zhang, Guangshe Zhao, Xiyu Zhang, Dingheng Wang, Gang Cao, and Guoqi Li. Sparser spiking activity can be better: Feature refine-and-mask spiking neural network for event-based visual recognition. *Neural Networks*, 166:410–423, 2023b.

Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In *International conference on learning representations*, volume 2024, pp. 52885–52907, 2024.

Hong Zhang and Yu Zhang. Memory-efficient reversible spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16759–16767, 2024.

Dongcheng Zhao, Sicheng Shen, Zhenyu Yang, Zhiyuan Li, Jinyan Yu, Yongjian Wang, Tiechui Yao, Wenli Zhang, and Tielin Zhang. Spiking local interaction and adaptive complementary fusion for spiking transformer. *arXiv preprint arXiv:2608.19238*, 2026.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5122–5130. IEEE, 2017.

Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. Qkformer: Hierarchical spiking transformer using qk attention. *Advances in Neural Information Processing Systems*, 37:13074–13098, 2024a.

Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Han Zhang, Jiaqi Wang, Huihui Zhou, Zhengyu Ma, and Yonghong Tian. Spikingformer: A key foundation model for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2236–2244, 2026.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. *arXiv preprint arXiv:2209.15425*, 2022.

Zhaokun Zhou, Yijie Lu, Yanhao Jia, Kaiwei Che, Jun Niu, Liwei Huang, Xinyu Shi, Yuesheng Zhu, Guoqi Li, Zhaofei Yu, et al. Spiking transformer with experts mixture. *Advances in Neural Information Processing Systems*, 37:10036–10059, 2024b.

A Appendix

A.1 Neuron Encoding

Let $x(\mathbf{p}) \in [0,1]$ denote the normalized input intensity or feature value at coordinate $\mathbf{p}$ and let $t \in \{1,\dots,T\}$ denote the simulation step. We write I_t for the input supplied by an encoder. Unlike internal binary neuron outputs, I_t may be real-valued under direct or amplitude-weighted encoding.

Direct Encoding. The input value is directly injected into the network at every time step:

$$I_t(\mathbf{p}) = x(\mathbf{p}), \qquad t = 1,\dots,T. \tag{6}$$

Thus, the same input current is presented throughout the entire simulation window, resulting in a temporally constant input sequence (Rathi & Roy, 2021).

Phase Encoding. We first quantize the normalized input into an 8-bit integer,

$$v(\mathbf{p}) = \lfloor 256\, x(\mathbf{p}) \rfloor, \tag{7}$$

and encode its binary representation across time as

$$I_t(\mathbf{p}) = \begin{cases} 2^{-(b+1)}, & \text{if } v_{7-b}(\mathbf{p}) = 1, \\ 0, & \text{otherwise}, \end{cases} \qquad b \equiv (t-1) \pmod 8, \tag{8}$$

where $v_k(\mathbf{p})$ denotes the k-th bit of the 8-bit integer $v(\mathbf{p})$, with $k = 7$ corresponding to the most significant bit. The encoder sequentially traverses the eight bit planes and assigns larger weights to more significant bits (Kim et al., 2018).

Rate Encoding. For rate encoding, spikes are sampled independently at each time step according to a Bernoulli distribution:

$$I_t(\mathbf{p}) \sim \text{Bernoulli}(x(\mathbf{p})), \qquad \mathbb{E}[I_t(\mathbf{p})] = x(\mathbf{p}), \qquad t = 1,\dots,T. \tag{9}$$

Therefore, the expected firing rate over the simulation window is proportional to the magnitude of the input feature (Rueckauer et al., 2017).

Time-to-First-Spike (TTFS) Encoding. For TTFS encoding, each input neuron emits a single spike, whose latency is inversely related to the input magnitude. We define the firing time as

$$t^*(\mathbf{p}) = 1 + \left\lfloor \big(1 - x(\mathbf{p})\big)(T-1) \right\rfloor. \tag{10}$$

The corresponding spike sequence is

$$I_t(\mathbf{p}) = \begin{cases} \dfrac{1}{t^*(\mathbf{p})}, & t = t^*(\mathbf{p}), \\ 0, & \text{otherwise}. \end{cases} \tag{11}$$

Hence, larger input values produce earlier spikes, whereas smaller values produce later spikes. The factor $1/t^*(\mathbf{p})$ can be used to modulate the spike amplitude; alternatively, a binary spike with unit amplitude may be adopted (Mostafa, 2017).

A.2 Experiments on Encoding Methods

Table 5: Performance of baselines & GSAP on different neuron encodings.

Model	Direct	Phase	Rate	TTFS
Spikformer	95.51	82.63	82.68	81.87
SDT	95.60	85.33	84.06	84.52
TIM	94.20	81.43	81.48	80.66
QKFormer	96.18	87.76	83.77	84.69
TEFormer	96.24	89.92	84.74	87.46
GSAP (ours)	95.95	**92.51**	**85.46**	**90.96**

We compare direct, phase, rate, and TTFS input encoding to assess GSAP across different spiking representations. GSAP reaches 92.51%, 85.46%, and 90.96% accuracy under the three alternative encodings, respectively, outperforming the compared models in each setting. TEFormer, the strongest alternative-encoding baseline in the table, reaches 89.92%, 84.74%, and 87.46%. Under direct encoding, GSAP reaches 95.95%, below TEFormer at 96.24%. Thus, GSAP performs well across the tested alternative encodings even relative to models with stronger direct-encoding results, supporting its applicability beyond direct encoding.

B GSAP Global Modeling Capability

Global spatial propagation and receiver gating. To understand how GSAP performs token interaction beyond explicit pairwise affinity modeling, we analyze the propagation behavior of a trained ImageNet-1K Spikingformer-8-768 model with $T = 4$. Input images are resized and center-cropped to 224×224 with a crop ratio of 1.0, followed by ImageNet normalization. We examine the fourth block, which operates on a 14×14 token grid, in evaluation mode with spiking-neuron states reset before each reference or perturbed replay. To isolate the global propagation pathway, we zero the central token at the output of the feature spiking neuron across all channels and time steps, while fixing the receiver gate to its unperturbed value and leaving the local and identity branches unchanged. We visualize the absolute difference between the reference and perturbed activations, averaged over channels and time. As shown in

$$\begin{aligned} &\text{Since} \quad |j-v| \le W-1 \le \frac{k-1}{2} \quad \text{and} \quad |i-u| \le H-1 \le \frac{k-1}{2}, \\ &\implies \quad (i,j) \to (i,v) \to (u,v) \quad \text{(valid structural path)}, \\ &\therefore \quad \mathcal{R}_{\text{struct}}(M^t_{u,v}) = \bigcup_{i=1}^{H} \{(i,j) : 1 \le j \le W\} = \{1,\dots,H\} \times \{1,\dots,W\}. \end{aligned} \tag{12}$$

Fig. 5(a), the perturbation first propagates along the source row and subsequently extends across rows through column propagation, producing a two-dimensional response within the same simulation step. Meanwhile, the receiver gate selectively attenuates the propagated current before spike fusion. This behavior directly reflects the axial propagation pathway described in Eq. 12, providing empirical evidence that GSAP establishes structured long-range communication while retaining receiver-dependent control over the transmitted signal.

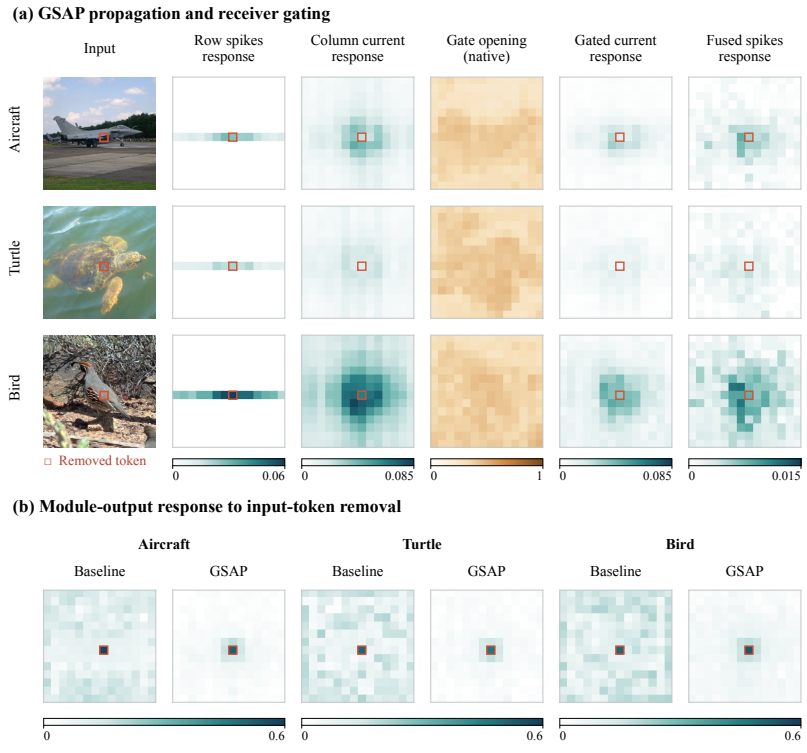


Figure 5: Token perturbation analysis of GSAP on the trained ImageNet-1K model. (a) Removing a central feature token reveals axial propagation and receiver-conditioned message selection. (b) Interaction responses after removing an input token, where the GSAP gate responds naturally. Response magnitude indicates perturbation sensitivity.

Spatial sensitivity to token removal. We further examine whether such global propagation leads to spatially indiscriminate responses by comparing the trained baseline and GSAP models under the same token-removal protocol. For each model, we zero the central input token of the fourth interaction block across all channels and time steps, and measure the resulting change in the interaction-branch output before the outer residual addition. The absolute response is averaged over channels and time, with each model receiving its own intermediate features; unlike the preceding pathway-isolation experiment, the GSAP receiver gate is allowed to respond naturally to the perturbed input. All response maps in Fig. 5(b) share a common absolute color scale. In the displayed examples, the baseline exhibits broadly distributed output changes, whereas GSAP produces a more spatially concentrated response around the removed token, with weaker changes at distant positions. Together with the propagation visualization in panel (a), this result shows that GSAP combines long-range information propagation with localized sensitivity to input perturbations: information can be transmitted over distant spatial positions without inducing uniformly strong influence across the token grid. The response maps characterize perturbation sensitivity rather than absolute activation magnitude, and their absolute scales also depend on the intermediate feature distributions. The complete intervention protocol is provided in Appendix B.

C EXPERIMENT

C.1 IMAGENET

Table 6: Architecture specifications of the eight-block GSAP models on ImageNet. S-ST denotes a Spikingformer-style single-stage backbone, and H-ST denotes a QKFormer-style hierarchical backbone. C_S and C_H are their final channel dimensions. S-ST applies all eight Transformer blocks at a fixed resolution, whereas H-ST distributes them across three resolutions with depths $(1, 2, 5)$. Axial kernel sizes are specified for $H = W = 224$ and follow $k = 2\max(H_s, W_s) - 1$ at each feature resolution. PW and DW denote pointwise and depthwise convolutions, respectively; SN denotes a LIF neuron, with the subscript indicating a firing threshold of 0.5. Normalization and neuron placements in the tokenizer, patch embedding, and MLP follow the respective backbone implementations.

Stage	**Spatial resolution**	**Layer specification**		**GSAP / S-ST**	**GSAP / H-ST**
Tokenizer	$H \times W$ ↓ Feature grid	Convolutions	Conv 3×3, stride 1	$C_S/8,\ C_S/4,\ C_S/2,\ C_S,\ C_S$	$C_H/8,\ C_H/4,\ C_H/4$
		Downsampling	MaxPool 3×3, stride 2	$\times 4$	$\times 2$
		Output	Resolution; channel dimension	$\frac{H}{16} \times \frac{W}{16};\ C_S$	$\frac{H}{4} \times \frac{W}{4};\ C_H/4$
Fine	$\frac{H}{4} \times \frac{W}{4}$	Blocks	GSAP Transformer block	—	$\times 1$, dim $C_H/4$
		Axial kernels	DWConv $1 \times k$; DWConv $k \times 1$	—	$k = 111$
Intermediate	$\frac{H}{8} \times \frac{W}{8}$	Downsampling	Residual patch embedding, stride 2	—	$C_H/4 \to C_H/2$
		Blocks	GSAP Transformer block	—	$\times 2$, dim $C_H/2$
		Axial kernels	DWConv $1 \times k$; DWConv $k \times 1$	—	$k = 55$
Coarse	$\frac{H}{16} \times \frac{W}{16}$	Downsampling	Residual patch embedding, stride 2	—	$C_H/2 \to C_H$
		Blocks	GSAP Transformer block	$\times 8$, dim C_S	$\times 5$, dim C_H
		Axial kernels	DWConv $1 \times k$; DWConv $k \times 1$	$k = 27$	$k = 27$
Head	1×1	Aggregation	Spatial and temporal averaging	Global average pooling; temporal mean	
		Classifier	Fully connected layer	$C_S \to 1000$	$C_H \to 1000$
Block specifications					
Residual patch embedding	Main pathway		Conv $3 \times 3 \to$ MaxPool, stride $2 \to$ Conv 3×3		
	Shortcut		Conv 1×1, stride 2; additive fusion		
GSAP	Input encoding		LIF neuron: $\mathbf{S} = \mathrm{SN}(\mathbf{X})$		
	Local pathway		DWConv 3×3 + BN $\to \mathbf{L}$		
	Feature projection		PWConv 1×1 + BN + SN $\to \mathbf{F}$		
	Axial propagation		$\mathbf{H} = \mathrm{SN}\big(\mathbf{F} + \mathrm{BN}(\mathrm{DWConv}_{1\times k}(\mathbf{F}))\big)$ $\mathbf{M} = \mathrm{BN}(\mathrm{DWConv}_{k\times 1}(\mathbf{H}))$		
	Receiver gate		PWConv 1×1 + BN + $\mathrm{SN}_{0.5} \to \mathbf{G}$		
	Gated fusion		$\mathbf{A} = \mathrm{SN}_{0.5}\big(\mathrm{BN}(\mathbf{S} + \mathbf{L} + \mathbf{M} \odot \mathbf{G})\big)$		
	Output projection		PWConv 1×1 + BN; channel dimension preserved		
GSAP Transformer block	Spatial interaction		$\mathbf{Y} = \mathbf{X} + \mathrm{GSAP}(\mathbf{X})$		
	Channel MLP		Conv $1 \times 1 \times 2$, expansion ratio 4; $\mathbf{Out} = \mathbf{Y} + \mathrm{MLP}(\mathbf{Y})$		

ImageNet training configurations. We use RGB inputs of size $3 \times 224 \times 224$ and $T = 4$ time steps. Images are normalized using channel-wise means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$, with bicubic interpolation for resizing. Both the baseline and GSAP models are trained with AdamW, a weight decay of 0.05, and a per-GPU batch size of 24. The training schedule comprises 300 epochs followed by a 10-epoch cooldown, with 20 warmup epochs starting from a learning rate of 10^{-6} and cosine decay to a minimum learning rate of 10^{-5}. The baseline uses a peak learning rate of 5×10^{-4}. For GSAP, this value is scaled linearly by $B/288$, where B denotes the global training batch size. Training augmentation includes random horizontal flipping with probability 0.5, RandAugment (`rand-m9-mstd0.5-inc1`), Mixup with $\alpha = 0.8$, CutMix with $\alpha = 1.0$, and random erasing with probability 0.25. Mixup and CutMix are selected with equal probability. We use label smoothing of 0.1 and automatic mixed-precision training.

C.2 SEGMENTATION

Table 7: Architecture specifications of the GSAP segmentation models on ADE20K. Both backbone adaptations use a shared multi-scale layout, with GSAP replacing all eight attention modules in the final two stages. P_1–P_4 denote the backbone features passed to the FPN neck after temporal averaging. The axial kernels are configured for 512×512 training crops, corresponding to a 32×32 feature grid in both GSAP stages. PW and DW denote pointwise and depthwise convolutions; BN and SN denote batch normalization and LIF spiking neurons. The subscript 0.5 indicates the firing threshold.

Stage	**Spatial resolution**	**Layer specification**		**GSAP** Spikingformer-based	**GSAP** QKFormer-based
1	$\frac{H}{2} \times \frac{W}{2}$	Downsampling	Conv 7×7, stride 2, dim 32	Shared configuration	
		Blocks	Conv-based SNN block	$\times 1 \to P_1$	$\times 1 \to P_1$
2	$\frac{H}{4} \times \frac{W}{4}$	Downsampling	Conv 3×3, stride 2, dim 64	Shared configuration	
		Blocks	Conv-based SNN block	$\times 1 \to P_2$	$\times 1 \to P_2$
3	$\frac{H}{8} \times \frac{W}{8}$	Downsampling	Conv 3×3, stride 2, dim 128	Shared configuration	
		Blocks	Conv-based SNN block	$\times 2 \to P_3$	$\times 2 \to P_3$
4	$\frac{H}{16} \times \frac{W}{16}$	Downsampling	Conv 3×3, stride 2, dim 256	Shared configuration	
		Blocks	GSAP Transformer block	GSAP $\times 6$	GSAP $\times 6$
5	$\frac{H}{16} \times \frac{W}{16}$	Channel expansion	Conv 3×3, stride 1, dim 360	Shared configuration	
		Blocks	GSAP Transformer block	GSAP $\times 2 \to P_4$	GSAP $\times 2 \to P_4$
Block specifications					
Conv-based SNN block	SepConv		PW 1×1 (ratio 2) $\to$ DW $7 \times 7 \to$ PW 1×1; membrane shortcut		
	Channel convolution		Conv $3 \times 3 \times 2$; expansion ratio 4; membrane shortcut		
GSAP Transformer block	Local pathway		DW 3×3 + BN, applied to input spikes		
	Propagation pathway		PW 1×1 + BN + SN $\to$ DW 1×63 + BN $\to$ feature shortcut + SN $\to$ DW 63×1 + BN		
	Receiver gate		PW 1×1 + BN + $\mathrm{SN}_{0.5}$; element-wise gating of propagated features		
	Fusion and projection		Input spikes + local features + gated propagated features $\to$ BN + $\mathrm{SN}_{0.5} \to$ PW 1×1 + BN		
	Channel MLP		Conv $1 \times 1 \times 2$; expansion ratio 4		
	Outer residuals		$\mathbf{Y} = \mathbf{X} + \mathrm{GSAP}(\mathbf{X}), \quad \mathbf{Z} = \mathbf{Y} + \mathrm{MLP}(\mathbf{Y})$		
Feature pyramid and segmentation head					
Feature interface	Backbone outputs		P_1–P_4: channels $(32, 64, 128, 360)$; spatial strides $(2, 4, 8, 16)$		
	Temporal aggregation		Temporal mean of each feature map over $T = 4$ steps		
FPN	Lateral projection		1×1 convolution; 128 output channels per level		
	Pyramid fusion		Top-down upsampling and addition; 3×3 output convolutions		
Decode head	Multi-scale fusion		128-channel scale branches; bilinear resizing and summation		
	Prediction		1×1 classifier, 150 classes; resize logits to the image resolution		

Segmentation configurations. Table 7 details the GSAP configurations for semantic segmentation on ADE20K. We adapt the Spikingformer-based and QKFormer-based backbones to a shared multi-scale layout using 512×512 training crops and $T = 4$ time steps. The first three stages contain $(1, 1, 2)$ convolutional SNN blocks with channel widths $(32, 64, 128)$. The final two stages operate at an output stride of 16 and contain six and two GSAP Transformer blocks with channel widths of 256 and 360, respectively. Each Transformer block uses an MLP expansion ratio of 4. GSAP replaces all eight attention modules while retaining the corresponding backbone's convolutional stages, channel transitions, MLPs, and residual connections. For the resulting 32×32 training feature grids, each GSAP module uses depthwise axial kernels of sizes 1×63 and 63×1.

The backbone outputs four feature maps with channel widths $(32, 64, 128, 360)$ at spatial strides $(2, 4, 8, 16)$. These features are averaged over time and passed to a 128-channel FPN neck and a 128-channel segmentation head for 150-class prediction. Both models are trained from scratch, and evaluation uses whole-image inference. The axial kernel sizes remain fixed during inference; their spatial coverage therefore does not automatically expand for images larger than the training crops.

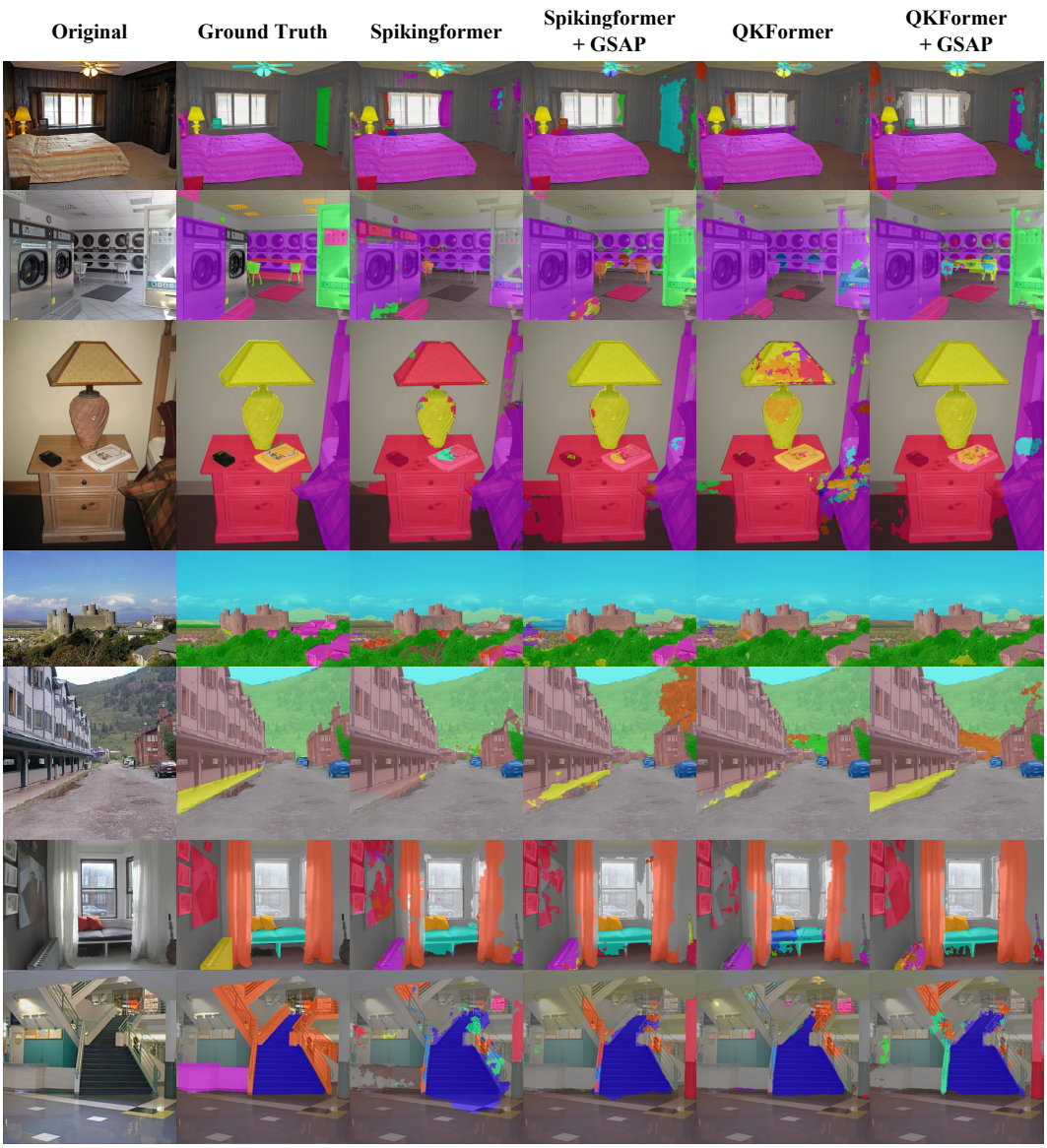


Figure 6: Selected qualitative semantic segmentation results on the ADE20K validation set. Columns show the input image, ground truth, Spikingformer, Spikingformer with GSAP, QKFormer, and QKFormer with GSAP. Rows show a bedroom, a laundromat, a lamp, a castle, a street scene, a curtained room, and an indoor staircase.

Visualization Figure 6 illustrates local improvements in semantic region consistency with GSAP across two backbones. In the lamp example, GSAP corrects the lampshade prediction in Spikingformer and reduces fragmented labels in QKFormer. In the laundromat, both variants recover more of the rightmost machine with the ground-truth class. The outdoor examples show more coherent vegetation labeling around the castle, particularly for Spikingformer, and more complete sidewalk predictions in the street scene, particularly for QKFormer. The added indoor examples show more consistent curtain labeling and recovery of parts of the staircase structure. These examples were selected to highlight improvements; errors remain in some regions, such as the bedroom doorway and street vegetation.

D ONLINE LEARNING ADAPTION

SNNs Online Learning In online learning for SNNs, gradients are typically accumulated during single-step forward propagation, followed by periodic backward updates. Taking OTTT as an example, neuronal dynamics accumulate temporal traces $\hat{\mathbf{a}}^l[t]$ (Eq. 13), while the learning signal is propagated through the computational graph to the neurons and combined with these traces to construct the full gradients of the corresponding weights (Eq. 14). This learning paradigm preserves both computational efficiency and strict causality (Xiao et al., 2022).

$$\hat{\mathbf{a}}^l[t] = \lambda \hat{\mathbf{a}}^l[t-1] + \mathbf{s}^l[t], \tag{13}$$

$$\nabla_{\mathbf{W}^l} \mathcal{L}_t = \mathbf{g}^{l+1}[t] \left(\hat{\mathbf{a}}^l[t]\right)^\top. \tag{14}$$

vfsd

Spiking Transformer Adaptation. Existing online learning algorithms have been developed primarily for spike-based CNNs, and we find that they are difficult to adapt effectively to Spiking Transformers. According to Eqs. 13 and 14, weight gradients are constructed from temporal traces and learning signals propagated through the chain rule. However, sparse QKV activity can suppress components of the learning signal, as indicated by the red term in Eq. 15.

$$\nabla_{\mathbf{W}_Q^l} \mathcal{L}_t = \mathbf{g}_Q^{l+1}[t] \left(\hat{\mathbf{a}}^l[t]\right)^\top, \mathbf{g}_Q^{l+1}[t] = \phi'\Big(\mathbf{U}_Q^{l+1}[t] - \vartheta\Big) \odot \big(s\mathbf{D}_t \mathbf{V}_t^\top \mathbf{K}_t\big). \tag{15}$$

Sparse learning signals can limit gradient contributions within each time step, while weak alignment across steps may further reduce the effectiveness of gradient accumulation. As shown in Fig. 4, under OTTT-A, Spiking Attention exhibits lower cross-step gradient cosine similarity than the ungated GSAP variant.

We remove the receiver gate from GSAP and evaluate the resulting ungated variant on the same benchmarks under identical experimental settings. Across both OTTT variants and evaluated tasks, this variant consistently improves accuracy over the Spiking Attention baseline. In particular, under OTTT-A on CIFAR-100, it improves accuracy by more than 10 percentage points (Fig. 4). These results support propagation-based token interaction as a promising approach to improving online learning in Spiking Transformers.

OTTT Configuration We evaluate the spiking-attention baseline and GSAP on CIFAR-10 and CIFAR-100 using OTTT-A and OTTT-O. Both variants compute gradients at each time step. OTTT-A accumulates these gradients over the complete sequence and updates the parameters once, whereas OTTT-O updates the parameters immediately after each time step. Thus, the two variants perform one and T parameter updates per input sequence, respectively.

For both datasets, we use $3 \times 32 \times 32$ inputs and present each image for $T = 6$ time steps. Training runs for 200 epochs with a batch size of 128, using SGD with momentum 0.9 and zero weight decay. The learning rate starts at 0.1 and follows a cosine annealing schedule with a period of 200 epochs and a minimum value of zero, without warmup. Training augmentation consists of random cropping with four-pixel padding, Cutout, and random horizontal flipping. Both datasets use channel-wise normalization with means $(0.4914, 0.4822, 0.4465)$ and standard deviations $(0.2023, 0.1994, 0.2010)$. The LIF membrane time constant is $\tau = 2$, and the sigmoid surrogate gradient uses $\alpha = 4$. At each time step, the loss is $[0.95\mathcal{L}_{\mathrm{CE}} + 0.05\mathcal{L}_{\mathrm{MSE}}]/T$, where the MSE term compares the model output with the one-hot class target. Neuron states are reset at the start of each input sequence, and predictions are obtained by summing the outputs over all time steps. We use a random seed of 2022.

E Complexity and Energy Analysis

Configurations and counting. We analyze the full local implementations using a common effective activity rate of $\rho = 0.1$ to isolate architectural cost. CIFAR-10 and TinyImageNet use inputs of 32×32 and 64×64, respectively, with $T = 4$, four blocks, a final width of 384, and an MLP expansion ratio of four. Their QK backbones use stage widths of 96/192/384 and block counts of 1/1/2. CIFAR10-DVS uses two-channel 128×128 event-count frames and $T = 16$. The single-stage DVS backbones use two blocks and 256 channels. For the two-stage DVS QK backbone, we report both its native MLP ratio of one and a ratio-four structural control, keeping the ratio unchanged within each baseline–GSAP pair. These are architecture-level comparisons without checkpoint accuracy evaluation.

The FLOP counts in Tab. 8 include convolutions, linear projections, and executed matrix products over the complete temporal sequence, with two FLOPs per multiply–accumulate (MAC). Parameter counts include all registered parameters. We obtain operation counts from full-size forward executions and verify them against an independent operator counter and analytical GSAP counts. The accounting excludes neuron-state updates, normalization, pooling, bias additions, general elementwise operations, memory access, and communication.

Normalized synaptic energy. Let A denote the number of MACs in the input encoder, non-spiking classification heads, and non-spiking temporal mixing. Let C denote the remaining convolution, linear, and attention-matrix MAC capacity, and R the accumulation capacity of QK token reduction and SDSA reduction. All counts already include T. Following the arithmetic-cost convention in QKFormer Zhou et al. (2024a), we use $E_{\mathrm{MAC}} = 4.6$ pJ and $E_{\mathrm{AC}} = 0.9$ pJ and define

$$E_{\mathrm{norm}}(\rho) = \left[4.6A + 0.9\rho(C + R)\right] \times 10^{-9}\ \mathrm{mJ}. \tag{16}$$

The common rate is assigned to effective inputs of the normalized synaptic and reduction operations, including inputs after binary masking. For QK token attention and SDSA, each reduced input provides one accumulation opportunity, giving $R = TNC$ per block. This convention does not assume independence between the underlying spike tensors. The input convolution is charged at every executed time step. Event-count frames and temporally or spatially averaged classifier inputs retain their MAC cost; the SDT head is normalized as AC because it is preceded by a spiking neuron. GSAP computes its axial message before applying its gate, so the gate introduces no additional discount on axial convolution cost.

Equation equation 16 is a controlled structural normalization. Equal marginal LIF firing rates alone do not determine joint mask activity or the cost of multi-valued residual inputs. In particular, some Spikformer and QKFormer projections consume sums of spikes. Their implementation may require dense MACs or multiple event accumulations, depending on the hardware mapping. Thus, the reported values are normalized synaptic energy rather than measured device energy or a complete energy prediction from firing rate alone.

Attention complexity. For one time step and one block with $N = HW$ tokens and C channels, the implemented SSA evaluates $(QK^{\mathsf{T}})V$ and has MAC capacity

$$\mathcal{C}_{\mathrm{SSA}} = 4NC^2 + 2N^2C. \tag{17}$$

GSAP uses three pointwise projections, a depthwise 3×3 local convolution, and two depthwise axial convolutions of length k, yielding

$$\mathcal{C}_{\mathrm{GSAP}} = 3NC^2 + NC(9 + 2k). \tag{18}$$

The current implementation sets $k = 2S - 1$ for a square $S \times S$ feature map. Consequently, its spatial propagation scales as $O(CN^{3/2})$, while its intermediate feature representation scales as $O(NC)$ and avoids an explicit $N \times N$ attention matrix. QK token attention already uses $3NC^2$ projection capacity and an $O(NC)$ reduction, so replacing these early-stage modules adds spatial mixing cost that offsets part of the savings from replacing later SSA blocks.

Comparison. On the Spikingformer backbone, GSAP reduces full-network FLOPs by 4.98%, 8.35%, and 2.99% on CIFAR-10, TinyImageNet, and CIFAR10-DVS, respectively. At $\rho = 0.1$, the corresponding normalized energy reductions are 4.65%, 7.81%, and 1.33%. On QKFormer,

Table 8: Complexity and normalized energy at $\rho = 0.1$. Arrows indicate baseline $\rightarrow$ GSAP. Parentheses report reductions relative to the corresponding baseline.

Dataset	Backbone	Params (M)	FLOPs (G)	Norm. energy (mJ)
CIFAR-10	Spikingformer	$9.3263 \rightarrow 8.8056$	$7.4723 \rightarrow 7.1003$ ($\downarrow 4.98\%$)	$0.3602 \rightarrow 0.3435$ ($\downarrow 4.65\%$)
	QKFormer	$6.7062 \rightarrow 6.4748$	$12.1406 \rightarrow 12.0886$ ($\downarrow 0.43\%$)	$0.5703 \rightarrow 0.5679$ ($\downarrow 0.42\%$)
TinyImageNet	Spikingformer	$9.3933 \rightarrow 8.9217$	$31.0972 \rightarrow 28.5020$ ($\downarrow 8.35\%$)	$1.4955 \rightarrow 1.3787$ ($\downarrow 7.81\%$)
	QKFormer	$6.7794 \rightarrow 6.5971$	$49.1663 \rightarrow 48.9084$ ($\downarrow 0.52\%$)	$2.3088 \rightarrow 2.2970$ ($\downarrow 0.51\%$)
CIFAR10-DVS	Spikingformer	$2.5687 \rightarrow 2.4586$	$12.1132 \rightarrow 11.7514$ ($\downarrow 2.99\%$)	$1.2261 \rightarrow 1.2098$ ($\downarrow 1.33\%$)
	QKFormer	$1.5041 \rightarrow 1.4602$	$16.2404 \rightarrow 16.1843$ ($\downarrow 0.35\%$)	$1.0714 \rightarrow 1.0688$ ($\downarrow 0.24\%$)

the FLOP reductions are 0.43%, 0.52%, and 0.35%, with normalized energy reductions of 0.42%, 0.51%, and 0.24%; the DVS comparison uses the native MLP ratio of one. These results quantify the cost of substituting GSAP within each backbone under the stated normalization.