NEUROPROP: BIO-INSPIRED LOCAL STBP FOR MEMORY-EFFICIENT SPIKING TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatiotemporal backpropagation (STBP) is among the strongest methods for directly training deep spiking neural networks (SNNs), yet its globally coupled graph retains neuronal states across both depth and time. We introduce *NeuroProp*, a bio-inspired framework that reduces this memory burden by localizing, rather than replacing, STBP. NeuroProp partitions a spiking Transformer into consecutive units, stops gradients at unit boundaries, and performs full surrogate-gradient STBP over the blocks and time steps within each unit. This converts the depth-wise activation graph from a whole-network dependency to one governed by the largest unit. To coordinate gradient-isolated units, we propose adaptive intrinsic plasticity (AdaIP), inspired by contextual neuromodulation of neuronal excitability. AdaIP broadcasts a semantic condition and transforms it into channel-wise synaptic gain and membrane bias, equivalently adjusting the effective firing threshold and equilibrium potential of LIF neurons. Detached inter-unit interfaces also admit pipeline-parallel execution. NeuroProp reduces peak memory by 48.29% on SpikFormer and 79.41% on SDTv3 in controlled CIFAR-100 experiments, while AdaIP recovers part of the accuracy lost by naive localization. Without global-STBP fine-tuning, NeuroProp reaches 71.30% ImageNet-1K top-1 accuracy. These results establish local STBP with adaptive neuronal modulation as a practical accuracy–memory trade-off for scaling spiking Transformers.

1 INTRODUCTION

Spiking neural networks (SNNs) combine event-driven communication with stateful neuronal dynamics, offering a computational substrate for efficient temporal processing (Maass, 1997; Zheng et al., 2024). The recent success of deep SNNs, particularly spiking Transformers (Zhou et al., 2023; Yao et al., 2024), is largely enabled by surrogate-gradient spatiotemporal backpropagation (STBP) (Wu et al., 2018; Zheng et al., 2021). STBP assigns credit jointly through network depth and neuronal time, and remains among the most reliable routes to high accuracy. We therefore treat STBP as an optimization strength to retain.

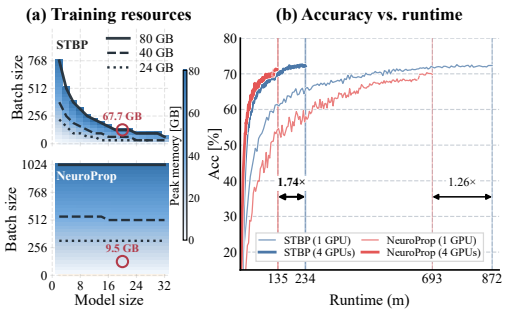


Figure 1: NeuroProp retains STBP within local units, coordinates them through adaptive neuronal modulation, and enables memory-efficient pipeline training.

Its strength comes with a severe memory cost. Global STBP stores activations, membrane potentials, and spike states for every layer and simulation step, so its activation footprint grows with network depth L and temporal horizon T (Liu et al., 2024; Du et al., 2025). Backpropagation-free and temporally local alternatives reduce this burden (Li et al., 2025; Mukhoty et al., 2023; Safa et al., 2021; Xiao et al., 2022), but also move away from the training rule that gives modern deep SNNs their performance. We instead ask: *can STBP remain the learning primitive while its globally coupled graph is made local?*

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.

We answer with *neural propagation (NeuroProp)*. NeuroProp partitions the network into consecutive units and stops gradients only at their boundaries. Each unit may contain one or more Transformer blocks and is trained by full STBP across its internal depth and all T time steps. Thus, NeuroProp localizes credit assignment along network depth without discarding temporal credit assignment. The retained activation graph is governed by the largest unit rather than the entire model, and detached interfaces create a unidirectional dependency suitable for pipeline parallelism.

Depth-local learning, however, removes the end-to-end error signal that normally coordinates intermediate representations. Biology suggests a complementary coordination channel: diffuse neuromodulatory signals regulate neuronal gain and excitability while local synaptic processes operate in distributed circuits (Schultz, 1998; Pawlak & Kerr, 2008; Gerstner et al., 2018). Motivated by this division of labor, we introduce *adaptive intrinsic plasticity (AdaIP)*. A shared semantic condition is converted into channel-wise input gain and membrane bias inside LIF dynamics. Unlike an auxiliary head that only evaluates a completed representation, AdaIP changes how neurons integrate current and emit spikes. The resulting algorithm remains supervised and uses surrogate gradients; its biological contribution is a concrete neuromodulatory design principle, not a claim of a fully biologically plausible learning rule.

As shown in Fig. 1(a), NeuroProp requires fewer training resources compared to STBP under identical configuration settings. This memory saving further improves the scalability of both the model and the training procedure. For example, the model can be trained with higher throughput under similar resource constraints, leading to faster convergence (see Fig. 1(b)). To the best of our knowledge, we are not aware of prior work that enables local training of spiking Transformers in this manner.

The main contributions of this work can be summarized as follows:

- NeuroProp retains STBP within bounded units while removing cross-unit backward dependencies, substantially reducing activation memory.
- AdaIP restores task-dependent coordination through neuromodulation-inspired control of intrinsic neuronal dynamics.
- experiments on static and event-based recognition characterize the resulting accuracy–memory trade-off and demonstrate pipeline scaling; NeuroProp reaches 71.30% ImageNet-1K top-1 accuracy without global-STBP fine-tuning.

2 Related Work

Two axes of credit locality in SNNs. Surrogate-gradient STBP assigns credit through both network depth and neuronal time, supporting accurate spiking CNNs and Transformers (Wu et al., 2018; Zheng et al., 2021; Cai et al., 2025; Zhou et al., 2023; Yao et al., 2024). Temporal-local alternatives avoid backpropagation through the complete history: e-prop uses eligibility traces (Bellec et al., 2020), OTTT performs online training (Xiao et al., 2022), and BrainTrace uses forward-mode gradients with linear temporal memory (Wang et al., 2026). NeuroProp instead retains full STBP over all T steps within each unit and truncates only cross-unit backward paths. Depth and temporal localization are thus complementary.

Depth-local supervised learning. DGL trains gradient-isolated modules with auxiliary predictors (Belilovsky et al., 2020); InfoPro preserves task-relevant information (Wang et al., 2021), AugLocal samples downstream layers to strengthen auxiliaries (Ma et al., 2024), and SGR reconciles neighboring local gradients (Yang et al., 2024). NeuroProp shares their depth partitioning principle but treats each module as a temporally unrolled spiking circuit trained by STBP, with coordination acting directly on LIF dynamics. Most closely, STDL localizes both spatial and temporal dependencies using memory-bounded downstream auxiliaries (Ma et al., 2025). In contrast, NeuroProp retains full within-unit BPTT and uses a lightweight semantic controller to modulate intrinsic excitability.

Neuromodulation and adaptive neuronal dynamics. Learnable membrane constants, thresholds, and synaptic dynamics shape SNN representations (Fang et al., 2021; Sun et al., 2023; Wei et al., 2023; Fan et al., 2025). Biological modulatory systems likewise regulate gain, excitability, and plasticity across distributed circuits (Dayan & Abbott, 2005; Schultz, 1998; Pawlak & Kerr, 2008; Gerstner et al., 2018). AdaIP translates this principle into a low-dimensional semantic signal that

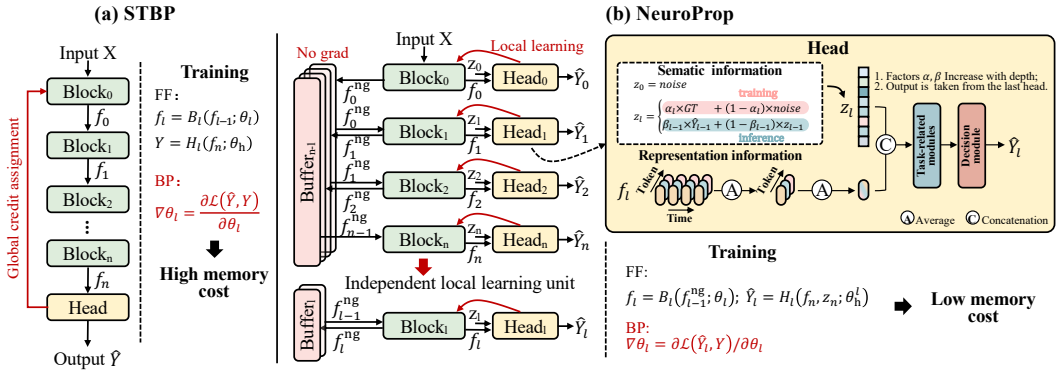


Figure 2: **Global versus unit-local STBP.** (a) A single readout drives whole-network credit assignment. (b) Illustrated one-block units exchange detached activity through buffers and learn from separate heads; each unit may more generally contain several blocks. Red arrows denote local updates, not cross-unit gradients. The head inset combines pooled activity with semantic context; exact conditioning protocols are specified in Section 3.4.

controls channel-wise membrane drive, while each unit remains optimized by supervised STBP. This is a bio-inspired design, not a claim that surrogate-gradient learning is biologically plausible.

3 Method

3.1 Design Objective and Scope

Our objective is to reduce the activation memory of strong direct SNN training without replacing its temporal learning rule. NeuroProp is therefore built around four constraints: (i) preserve STBP through all time steps and all blocks inside a unit; (ii) bound the depth of any live backward graph; (iii) allow downstream units to depend on upstream activity only through detached, forward-carried states; and (iv) require no target label during inference. These constraints distinguish the method from both end-to-end STBP and temporally online alternatives.

3.2 From Global to Unit-Local STBP

Let $f_{1:T}^{(0)}$ denote the input activity sequence obtained from sample X, where $1:T$ indexes the T simulation steps. For an SNN with L Transformer blocks, global STBP computes

$$f_{1:T}^{(l)} = B^{(l)}(f_{1:T}^{(l-1)}; \theta^{(l)}), \qquad \hat{y} = H(f_{1:T}^{(L)}), \tag{1}$$

where $B^{(l)}$, $\theta^{(l)}$, and $f_{1:T}^{(l)}$ are block l, its parameters, and its output sequence; H is the final readout and $\hat{y}$ the predicted class scores. Given target label y, a supervised loss $\ell(\hat{y}, y)$ sends gradients through every block and time step. Figure 2(a) depicts this globally coupled credit-assignment path.

NeuroProp groups consecutive blocks into G units $\{\mathcal{G}_g\}_{g=1}^{G}$ of depths $L_g \geq 1$, with $\sum_g L_g = L$. Below, superscript g indexes units rather than individual blocks. Define $\bar{f}_{1:T}^{(g-1)} = \mathrm{stopgrad}(f_{1:T}^{(g-1)})$ and $\bar{c}^{(g)} = \mathrm{stopgrad}(c^{(g)})$, where $c^{(g)}$ is the unit's semantic condition. The operator stopgrad preserves forward values but blocks derivatives to their producer. The unit computes

$$f_{1:T}^{(g)} = \mathcal{G}_g\left(\bar{f}_{1:T}^{(g-1)}, \bar{c}^{(g)}; \theta^{(g)}, \psi^{(g)}\right), \tag{2}$$

where $\theta^{(g)}$ contains the spiking-block parameters and $\psi^{(g)}$ the NAGT controller parameters implementing AdaIP (Section 3.3). This separates local synaptic optimization from contextual modulation of neuronal excitability.

A local head $h^{(g)}$, parameterized by $\phi^{(g)}$, produces class scores $u^{(g)} = h^{(g)}(f_{1:T}^{(g)}, \bar{c}^{(g)}; \phi^{(g)})$. In the head inset of Fig. 2(b), A denotes temporal and token averaging, and C concatenates the resulting

representation with semantic context before prediction. The figure's zero-based labels f_l^{ng}, z_l, and $\hat{Y}_l$ correspond to detached activity $\bar{f}_{1:T}^{(g)}$, condition $\bar{c}^{(g)}$, and prediction $u^{(g)}$, respectively, with $g = l + 1$. A representative local objective is

$$\mathcal{L}^{(g)} = \lambda_{\text{rep}} \|u^{(g)} - z(y)\|_2^2 + \lambda_{\text{ce}} \, \text{CE}(u^{(g)}, y). \tag{3}$$

Here $z(y)$ is the target representation (one-hot for standard classification), distinct from the figure's condition z_l; CE denotes cross-entropy, $\| \cdot \|_2$ the Euclidean norm, and $\lambda_{\text{rep}}, \lambda_{\text{ce}} \geq 0$ weight representation matching and classification.

With disjoint parameter sets $\Theta_g = \{\theta^{(g)}, \phi^{(g)}, \psi^{(g)}\}$, detachment on both input pathways gives

$$\nabla_{\Theta_j} \mathcal{L}^{(g)} = 0 \quad (j \neq g), \qquad \Theta_g \leftarrow \Theta_g - \eta_g \nabla_{\Theta_g} \mathcal{L}^{(g)}, \tag{4}$$

where j indexes another unit and η_g is a local learning rate; the update schematically represents local gradient optimization (AdamW in our experiments). The red learning loops in Fig. 2(b) thus terminate within their own units. Within a unit, surrogate gradients still traverse all constituent blocks and, for LIF neurons, the complete T-step membrane trajectory. After local backward computation, its graph can be released while detached outputs remain in the depicted buffers. These buffers carry activity, not backward error signals. *Local* therefore restricts depth-wise credit assignment without truncating within-unit temporal STBP.

Retained neuronal states and memory scaling. Let A_l be the storage required for block l's saved activations and neuronal states at one time step, for a fixed batch size and precision. The block-trajectory component scales as

$$M_{\text{global}}^{\text{act}} = \Theta\left(T \sum_{l=1}^{L} A_l\right), \qquad M_{\text{NeuroProp}}^{\text{act}} = \Theta\left(T \max_g \sum_{l \in \mathcal{G}_g} A_l\right). \tag{5}$$

Here M^{act} denotes saved block-state memory, $l \in \mathcal{G}_g$ indexes the blocks assigned to unit g, and $\Theta(\cdot)$ denotes asymptotic order. For uniform blocks of fixed state size, the depth–time dependence reduces from $\Theta(TL)$ to $\Theta(T \max_g L_g)$. The local expression assumes one live unit graph per device; coexisting graphs require summing their footprints. It explains the memory mechanism in Fig. 2, not total GPU allocation: tokenizer states, parameters, optimizer states, heads, detached buffers, and runtime workspaces remain. We therefore also report measured end-to-end peak memory.

Figure 3 illustrates the partition trade-off. Larger units preserve longer-range depth-wise credit assignment but retain more neuronal states. Smaller units reduce each live graph and increase pipeline granularity, while relying more on local supervision and semantic coordination. A unit is thus a bounded optimization circuit, not necessarily a single block.

Pipeline schedule. The buffered interfaces in Fig. 2(b) remove the need for unit g to await gradients from unit $g + 1$. On multiple devices, successive micro-batches can occupy different units concurrently, each performing local forward computation and STBP. Communication and pipeline bubbles remain, but cross-unit backward synchronization is removed. Global-STBP pipelines, by contrast, must also schedule the coupled backward path.

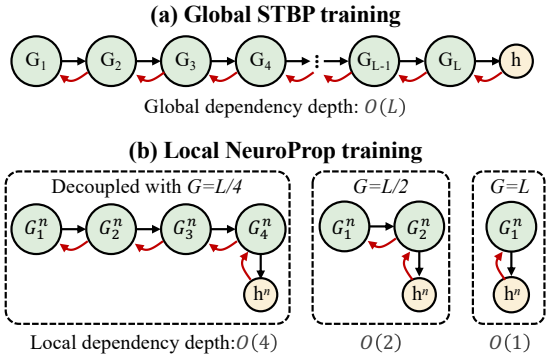


Figure 3: **A continuum of depth-local credit assignment.** Global STBP couples all L blocks. NeuroProp bounds the backward dependency by unit depth L_g; one block per unit gives $\mathcal{O}(1)$ depth-wise dependence while retaining STBP through time.

3.3 Adaptive Intrinsic Plasticity

Stopping inter-unit gradients creates efficient but weakly coordinated local circuits. AdaIP supplies a forward semantic broadcast analogous to contextual neuromodulation: a low-dimensional state

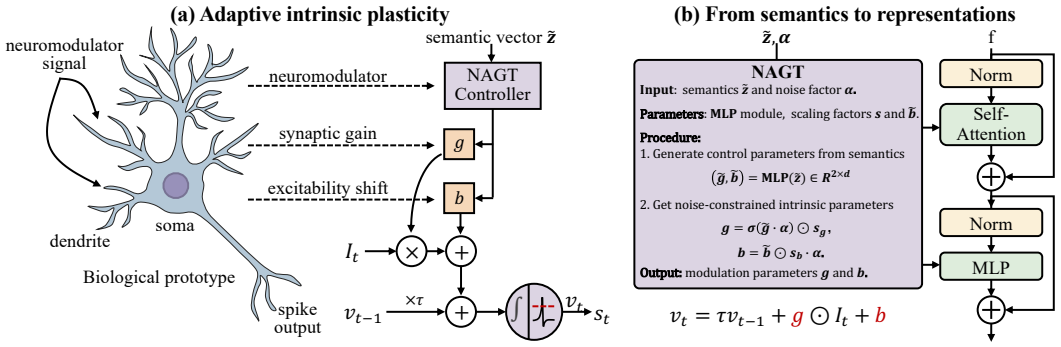


Figure 4: **Adaptive intrinsic plasticity.** A semantic state is transformed into channel-wise gain and bias and applied to the input current immediately before selected LIF neurons. This neuromodulation-inspired pathway changes effective excitability and shapes representations learned by local STBP.

controls the excitability of neuronal populations while synaptic optimization remains local to each unit.

For unit g, a lightweight neural adaptive guidance transform (NAGT) produces

$$(\tilde{g}^{(g)}, \tilde{b}^{(g)}) = \mathrm{MLP}(c^{(g)}), \qquad g^{(g)} = s_g\, \sigma(\alpha^{(g)} \tilde{g}^{(g)}), \quad b^{(g)} = s_b \alpha^{(g)} \tilde{b}^{(g)}, \tag{6}$$

where $g^{(g)}, b^{(g)} \in \mathbb{R}^D$, s_g and s_b bound the modulation, and $\alpha^{(g)}$ controls its unit-dependent strength. The parameters are broadcast over time and spatial tokens, and hence modulate embedding *channels*, not individual tokens. Immediately before a selected LIF node, AdaIP implements

$$v_t = \lambda v_{t-1} + g^{(g)} \odot I_t + b^{(g)}, \qquad s_t = H(v_t - V_{\mathrm{th}}). \tag{7}$$

Thus $g^{(g)}$ acts as a synaptic-current gain and $b^{(g)}$ shifts membrane drive. Ignoring reset only for this reparameterization, let $c = b^{(g)}/(1-\lambda)$ and $\bar{v} = (v - c)/g^{(g)}$ element-wise. Equation 7 becomes $\bar{v}_t = \lambda \bar{v}_{t-1} + I_t$ with effective threshold

$$V_{\mathrm{th}}^{\mathrm{eff}} = (V_{\mathrm{th}} - c)/g^{(g)}. \tag{8}$$

AdaIP can therefore be interpreted as semantic-dependent regulation of both the equilibrium potential and effective firing threshold, a direct computational analogue of context-dependent excitability. Importantly, it does not alter the decay coefficient λ itself. The local head and AdaIP consequently have non-redundant roles: the head converts a completed representation into a local supervisory signal, whereas AdaIP changes the membrane trajectory from which that representation is generated. Their parameters are optimized only by the loss of the unit that owns them.

3.4 Causal Semantic Propagation

We use a common semantic interface with two training instantiations. In teacher-conditioned training, unit g receives a stochastic target state

$$c^{(g)} = \sqrt{\alpha^{(g)}}\, z(y) + \sqrt{1 - \alpha^{(g)}}\, \epsilon^{(g)}, \qquad \epsilon^{(g)} \sim \mathcal{N}(0, I), \tag{9}$$

which resembles a noisy global modulatory signal rather than a layer-wise error derivative. This target-conditioned state is a training-only option; it is never evaluated from y at test time. In prediction-conditioned training, used for the ImageNet-1K model, the first unit receives a zero state and later units consume the detached aggregate prediction,

$$c_{\mathrm{train}}^{(1)} = 0, \qquad c_{\mathrm{train}}^{(g)} = \mathrm{stopgrad}(q^{(g-1)}) \quad (g > 1). \tag{10}$$

At inference, both variants use the same causal label-free pass: $q^{(0)} = 0$, unit g is conditioned only on $c_{\mathrm{test}}^{(g)} = q^{(g-1)}$, and

$$q^{(g)} = (1 - \beta^{(g)}) q^{(g-1)} + \beta^{(g)} u^{(g)}, \qquad \hat{y} = q^{(G)}. \tag{11}$$

Table 1: Classification accuracy (↑), grouped by dataset and training type (GT: global; LT: local). Best results are highlighted in **bold**. SpikFormer$_{1/2}$ denotes models with 4/8 Transformer modules.

Method	Type	Architecture	T	Acc. (%)
CIFAR-100				
STBP (Yao et al., 2023)	GT	SDT	4	**78.40**
STBP (Zhou et al., 2023)	GT	SpikFormer	4	77.86
NoProp (Li et al., 2025)	LT	FC (ANN)	-	45.93
LLS (Apolinario et al., 2025)	LT	Conv (ANN)	-	55.32
EIHL (Jiang et al., 2023)	LT	ResNet	0.5s	58.63
LOCALZO (Mukhoty et al., 2023)	LT	ResNet	4	76.89
NeuroProp (Ours)	LT	SpikFormer	4	76.94
NeuroProp (Ours)	LT	SpikFormer	8	**77.52**
Forward-forward (Krutsylo, 2025)	Other	SFF (ANN)	-	55.34
Distillation (Li et al., 2024)	Other	VGG	4	**73.60**
OTTT (Xiao et al., 2022)	Other	VGG	6	71.11
Hybrid training (Rathi et al., 2020)	Other	VGG	125	67.87
CIFAR10-DVS				
STBP (Zhou et al., 2023)	GT	SpikFormer	10	78.90
STBP (Zhou et al., 2023)	GT	SpikFormer	16	**80.90**
STBP (Li et al., 2022)	GT	Spikeformer	16	73.10
STBP (Yao et al., 2023)	GT	SDT	16	80.00
STEIHLDP (Safa et al., 2021)	LT	ResNet	0.5s	62.90
STDP (Safa et al., 2021)	LT	FC	0.005s	73.98
LOCALZO (Mukhoty et al., 2023)	LT	VGG	10	75.62
NeuroProp (Ours)	LT	SpikFormer	10	**79.80**
NeuroProp (Ours)	LT	SpikFormer	16	78.40
OTTT (Xiao et al., 2022)	Other	VGG	10	**77.10**
ImageNet-1K				
STBP (Yao et al., 2024)	GT	Meta-SpikeFormer	1	71.80
STBP (Yao et al., 2024)	GT	Meta-SpikeFormer	4	**74.10**
STBP (Zhou et al., 2023)	GT	SpikFormer	4	73.38
NeuroProp (Ours)	LT	SpikFormer$_1$	1	62.72
NeuroProp (Ours)	LT	SpikFormer$_1$	4	70.64
NeuroProp (Ours)	LT	SpikFormer$_2$	4	**71.30**
Distillation (Li et al., 2024)	Other	RegNetX	32	57.10
Distillation (Yang et al., 2025)	Other	ResNet	4	**68.12**
Hybrid training (Rathi et al., 2020)	Other	ResNet	250	61.48
Hybrid training (Rathi et al., 2020)	Other	VGG	250	65.19

Here $\beta^{(g)} \in [0, 1]$ controls how much the current unit revises the accumulated semantic state. This recursion excludes label leakage and preserves the forward-only dependency between units. Local heads and NAGT controllers are retained for this causal inference pass; the method therefore trades their modest parameter and compute overhead for inter-unit coordination, and does not hide that cost by discarding auxiliaries after training.

4 Experiments and Results

4.1 Experimental Settings

NeuroProp is evaluated on three representative benchmarks, including CIFAR-100 (Krizhevsky & Hinton, 2009) and ImageNet-1k (Deng et al., 2009) for static image classification and CIFAR10-DVS (Li et al., 2017) for event-based recognition, using two widely adopted deep SNN Transformer backbones, namely SpikFormer (Zhou et al., 2023) and SDTv3 (Yao et al., 2023). The main experiments are conducted on NVIDIA A800 80GB GPUs. In addition, an NVIDIA RTX 5090 Ti 24GB GPU is used to demonstrate the memory savings and efficiency gains of NeuroProp under

Table 2: CIFAR-100 ablation with a common reduced training budget. **Bold**: best accuracy or lowest memory per backbone. Memory reductions are relative to global STBP.

	Accuracy (%) ↑		Memory	
Method	Mean±std	Max	Peak (GB) ↓	Reduction
SpikFormer				
Global STBP	$\mathbf{72.21 \pm 0.15}$	**72.41**	73.81	–
NeuroProp w/o AdaIP	68.09 ± 0.15	68.32	**37.60**	49.06%
NeuroProp w/ AdaIP	69.65 ± 0.64	70.43	38.17	48.29%
SDTv3				
Global STBP	$\mathbf{71.62 \pm 0.38}$	**71.96**	22.68	–
NeuroProp w/o AdaIP	67.31 ± 0.24	67.44	**4.67**	79.41%
NeuroProp w/ AdaIP	68.55 ± 0.14	68.66	**4.67**	79.41%

a more constrained hardware budget. To further evaluate the scalability of NeuroProp on more complex downstream tasks, we also provide two representative case studies: continual learning on split CIFAR-100 (Rebuffi et al., 2017) and optical flow estimation on the DSEC (Gehrig et al., 2021) dataset. Detailed experimental settings are provided in **Tab. 6 of Appendix**.

4.2 Recognition Accuracy

The categoried representative methods by optimization paradigm are provided in Table 1. On CIFAR-100, NeuroProp obtains 76.94% and 77.52% at $T = 4$ and $T = 8$, respectively, approaching globally trained spiking Transformers while exceeding the listed local-training alternatives. On the event-based CIFAR10-DVS benchmark, it reaches 79.80% at $T = 10$ and 78.40% at $T = 16$. The non-monotonic change with T underscores that temporal horizon is a resource–optimization choice rather than a guaranteed source of accuracy.

We further evaluate NeuroProp on ImageNet-1k, a large-scale benchmark requiring deep spiking Transformers. NeuroProp provides the first local training results for SNN models on ImageNet-1k while achieving competitive performance compared to existing approaches. Specifically, SpikFormer$_2$ achieves 70.81% top-1 accuracy at T=4, outperforming conventional CNN-based SNNs (Li et al., 2024; Yang et al., 2025; Rathi et al., 2020) by a clear margin (e.g., +3.18% over Yang et al. (2025)). Compared with modern spiking Transformer models trained with global STBP (Yao et al., 2024; Zhou et al., 2023), NeuroProp yields slightly lower accuracy; however, this gap is modest given the substantial gains in training efficiency.

4.3 Contribution of Adaptive Intrinsic Plasticity

Table 2 separates the effect of graph localization from neuronal modulation. Naive localization reduces peak memory from 73.81 to 37.60GB on SpikFormer and from 22.68 to 4.67GB on SDTv3, but weakens accuracy because units no longer receive global gradients. AdaIP raises mean accuracy from 68.09% to 69.65% on SpikFormer and from 67.31% to 68.55% on SDTv3, with negligible additional memory. It therefore recovers a consistent portion of the coordination lost by local STBP rather than merely producing the memory reduction itself. **Appendix C** further analyzes how AdaIP translates semantic context into neuronal modulation.

4.4 Comparison with Memory-Efficient SNN Training

Table 3 compares four memory-efficient strategies on 4-block SpikFormer and 8-block SDTv3. They represent complementary mechanisms: checkpointing with lossless spike compression recomputes global STBP; STDL (Ma et al., 2025) localizes gradients in depth and time; OTTT (Xiao et al., 2022) avoids temporal backpropagation; and NeuroProp retains temporal STBP within bounded units. All methods use the same reduced-budget CIFAR-100 protocol within each backbone (150 epochs, AdamW, $T = 4$, batch size 512, learning rate 0.008, and one A800 GPU). To further reduce memory requirements, we also perform local decoupling on the Tokenizer module. **Appendix B** specifies measurement and software details.

Table 3: Comparison with other memory-efficient methods on CIFAR-100 dataset. **Bold** and underline mark the best and second-best result per backbone and metric, respectively.

Method	Accuracy (%) ↑	Peak mem. (GiB) ↓	Throughput ↑
SpikFormer: 4 blocks, 384 channels			
Checkpoint + comp.	**71.95**	**19.07**	325.2
STDL	<u>70.23</u>	25.83	503.2
OTTT	55.44	26.07	**700.1**
NeuroProp (Ours)	69.11	<u>24.18</u>	<u>687.6</u>
SDTv3: 8 blocks, 64 channels			
Checkpoint + comp.	**69.36**	2.74	<u>736.4</u>
STDL	67.27	**2.04**	204.9
OTTT	61.20	2.50	732.7
NeuroProp (Ours)	<u>67.46</u>	<u>2.47</u>	**1072.3**

Checkpoint + comp. denotes checkpointing with lossless spike compression.

Table 4: Comparison of largest feasible configurations (↑) on different training methods.

	NVIDIA A800 (80GB)			NVIDIA RTX 5090 Ti (24GB)		
Configuration	STBP	NeuroProp	Gain	STBP	NeuroProp	Gain
Batch size	512	**1024**	$+100\%$	512	**1024**	$+100\%$
Model size (blocks)	20	$\geq$ **48**	$\geq 140\%$	16	$\geq$ **64**	$\geq 300\%$
Time step (T)	16	**32**	$+100\%$	16	**36**	$+125\%$

Accuracy and memory remain competitive. On SpikFormer, NeuroProp reaches 69.11%, within 1.12 points of STDL, while avoiding OTTT's substantial drop to 55.44%. Its 24.18-GiB peak is second lowest, below STDL (25.83 GiB) and OTTT (26.07 GiB). On SDTv3, NeuroProp obtains the second-highest accuracy (67.46%), marginally exceeding STDL (67.27%) and outperforming OTTT (61.20%) by 6.26 points; its 2.47-GiB peak is again second lowest. NeuroProp therefore remains in the leading accuracy–memory group on both architectures.

The distinguishing advantage is throughput. NeuroProp processes 687.6 samples/s on SpikFormer during training, $1.37\times$ STDL and $2.11\times$ checkpointing, while retaining 98.2% of OTTT's throughput with 13.67 points higher accuracy. On SDTv3, it is the fastest method at 1072.3 samples/s—$1.46\times$ checkpointing and OTTT and $5.23\times$ STDL. NeuroProp is consequently the only method that ranks among the two lowest-memory and two highest-throughput solutions on both backbones without suffering OTTT's accuracy loss. These results position NeuroProp on a favorable practical Pareto frontier: competitive accuracy and memory, with substantially higher training throughput.

4.5 Memory and Feasible Training Region

We vary batch size, Transformer depth, and simulation length from a common 4-block SpikFormer setting (batch size 128, $T = 4$) and record the largest configuration before out-of-memory failure on an A800 (80 GB) and RTX 5090 Ti (24 GB). As shown in Table 4, NeuroProp doubles the maximum batch size to 1024 on both GPUs, extends feasible T from 16 to 32/36, and supports at least 48/64 blocks, whereas global STBP fails beyond 20/16 blocks. This agrees with Eq. 5: bounding the backward graph by unit size expands the feasible region along multiple scaling axes.

This enlarged region remains trainable (Figure 5): NeuroProp changes from 71.11% to 68.29% over batch sizes 64–1024, remains within 67.66–69.83% over 4–32 blocks, and improves from 69.17% to 71.05% as T increases from 4 to 16. Optional 10-epoch global-STBP refinement at batch size 32 improves every tested setting, with the largest gain of 2.43 percentage points at 32 blocks.

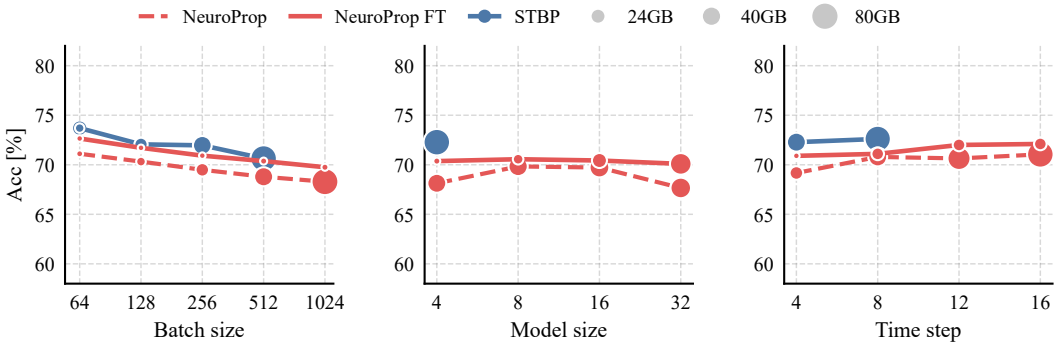


Figure 5: **Accuracy over different training regions on CIFAR-100.** Dashed red, solid red, and blue denote NeuroProp, NeuroProp with reduced-batch global-STBP fine-tuning (FT), and global STBP; marker area indicates the memory tier.

Table 5: Comparison of optical flow estimation (↓) on the DSEC event-based dataset.

Method	**AEE** ↓	**AAE** ↓	**3PE (%)** ↓
STBP	**2.138**	**5.882**	**13.967**
NeuroProp (Ours)	2.349	7.176	15.833

4.6 Pipeline-Parallel Efficiency

Detached interfaces allow different devices to run local STBP on separate micro-batches. Under an approximately 30-GB average per-GPU budget, NeuroProp provides 1.30×, 1.59×, and 1.89× speedups over global-STBP multi-GPU training for 16-, 24-, and 32-block models. Full settings are provided in **Appendix D**.

4.7 Performance on Complex Tasks

To assess performance beyond classification, we study optical flow and continual learning. On the event-based DSEC dataset (Gehrig et al., 2021), NeuroProp obtains 2.349 AEE and 15.833 3PE, trailing global STBP by 0.211 and 1.866 points, respectively. This confirms trainability on dense prediction while suggesting greater sensitivity to global credit assignment. Qualitative analysis and continual-learning results appear in **Appendix E**.

5 Conclusion

NeuroProp retains STBP within bounded units while removing cross-unit backward dependencies. Inspired by modular neural processing, it combines local depth–time credit assignment with AdaIP, which uses semantic context to modulate neuronal excitability. Across three recognition benchmarks, it offers a practical accuracy–memory trade-off and scales to ImageNet-1K without global-STBP fine-tuning. Matched comparisons show competitive accuracy and memory with high training throughput, while detached interfaces enable pipeline parallelism. NeuroProp also expands the feasible training region, with further accuracy gains from optional small-batch global refinement. Within-unit temporal storage and local-credit limitations remain. Together, these findings demonstrate how biologically inspired modularity can complement gradient-based optimization for resource-efficient spiking Transformer training.

AI use statement. Generative AI assisted language editing; the authors reviewed all outputs and take responsibility for the final content.

REFERENCES

Marco PE Apolinario, Arani Roy, and Kaushik Roy. LLS: local learning rule for deep neural networks inspired by neural activity synchronization. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7807–7816, 2025.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Decoupled greedy learning of CNNs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 736–745, 2020.

Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):3625, 2020.

Wuque Cai, Hongze Sun, Qianqian Liao, Jiayi He, Duo Chen, Dezhong Yao, and Daqing Guo. Robust spatiotemporal prototype learning for spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):18995–19009, 2025.

Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2005.

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, 2009.

Xin Du, WenTao Tong, Linshan Jiang, Di Yu, Zhiliang Wu, Qiang Duan, and Shuiguang Deng. SNN-IoT: Efficient partitioning and enabling of deep spiking neural networks in iot services. *IEEE Transactions on Services Computing*, 18(5):3191–3205, 2025.

Liangwei Fan, Hui Shen, Xiangkai Lian, Yulin Li, Man Yao, Guoqi Li, and Dewen Hu. A multisynaptic spiking neuron for simultaneously encoding spatiotemporal dynamics. *Nature Communications*, 16(1):7155, 2025.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2661–2671, 2021.

Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021. doi: 10.1109/LRA.2021.3068942.

Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in neural circuits*, 12:53, 2018.

Shuang Guo, Friedhelm Hamann, and Guillermo Gallego. Unsupervised joint learning of optical flow and intensity with event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7980–7989, 2025.

Tingting Jiang, Qi Xu, Xuming Ran, Jiangrong Shen, Pan Lv, Qiang Zhang, and Gang Pan. Adaptive deep spiking neural network with global-local learning via balanced excitatory and inhibitory mechanism. In *The twelfth international conference on learning representations*, 2023.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Andrii Krutsylo. Scalable forward-forward algorithm. *arXiv preprint, DOI:2501.03176*, 2025.

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 11:244131, 2017.

Qinyu Li, Yee Whye Teh, and Razvan Pascanu. NoProp: Training neural networks without back-propagation or forward-propagation. In *The 4th Conference on Lifelong Learning Agents*, 2025.

Yudong Li, Yunlin Lei, and Xu Yang. Spikeformer: A novel architecture for training high-performance low-latency spiking neural network. *arXiv preprint arXiv:2211.10686*, 2022.

Yuhang Li, Shikuang Deng, Xin Dong, and Shi Gu. Error-aware conversion from ANN to SNN via post-training parameter calibration. *International Journal of Computer Vision*, 132(9):3586–3609, 2024.

Qianhui Liu, Jiaqi Yan, Malu Zhang, Gang Pan, and Haizhou Li. LitE-SNN: designing lightweight and efficient spiking neural network through spatial-temporal compressive network search and joint optimization. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

Chenxiang Ma, Jibin Wu, Chenyang Si, and Kay Chen Tan. Scaling supervised local learning with augmented auxiliary networks. In *International Conference on Learning Representations*, 2024.

Chenxiang Ma, Xinyi Chen, Kay Chen Tan, and Jibin Wu. Spatio-temporal decoupled learning for spiking neural networks. *arXiv preprint arXiv:2506.01117*, 2025.

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

Bhaskar Mukhoty, Velibor Bojkovic, William De Vazelhes, Xiaohan Zhao, Giulia De Masi, Huan Xiong, and Bin Gu. Direct training of SNN using local zeroth order method. *Advances in Neural Information Processing Systems*, 36:18994–19014, 2023.

Verena Pawlak and Jason ND Kerr. Dopamine receptor activation is required for corticostriatal spike-timing-dependent plasticity. *Journal of Neuroscience*, 28(10):2435–2446, 2008.

Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. In *International Conference on Learning Representations*, 2020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2017.

Ali Safa, Hichem Sahli, André Bourdoux, Ilja Ocket, Francky Catthoor, and Georges GE Gielen. Learning event-based spatio-temporal feature descriptors via local synaptic plasticity: A biologically-realistic perspective of computer vision. *CoRR abs/2111.00791. URL: https://arxiv.org/abs/2111.00791*, 2021.

Wolfram Schultz. Predictive reward signal of dopamine neurons. *Journal of neurophysiology*, 80(1): 1–27, 1998.

Hongze Sun, Wuque Cai, Baoxin Yang, Yan Cui, Yang Xia, Dezhong Yao, and Daqing Guo. A synapse-threshold synergistic learning approach for spiking neural networks. *IEEE Transactions on Cognitive and Developmental Systems*, 16(2):544–558, 2023.

Yi Tian and Juan Andrade-Cetto. SDformerFlow: Spiking neural network transformer for event-based optical flow. In *International Conference on Pattern Recognition*, pp. 475–491, 2024.

Chaoming Wang, Xingsi Dong, Zilong Ji, Mingqing Xiao, Jiedong Jiang, Xiao Liu, Yuxiang Huan, and Si Wu. Model-agnostic linear-memory online learning in spiking neural networks. *Nature Communications*, 17:1745, 2026. doi: 10.1038/s41467-026-68453-w.

Yulin Wang, Zanlin Ni, Shiji Song, Le Yang, and Gao Huang. Revisiting locally supervised learning: An alternative to end-to-end training. In *International Conference on Learning Representations*, 2021.

Wenjie Wei, Malu Zhang, Hong Qu, Ammar Belatreche, Jian Zhang, and Hong Chen. Temporal-coded spiking neural networks with dynamic firing threshold: Learning with event-driven backpropagation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10552–10562, 2023.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, Volume 12 - 2018, 2018.

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.

Shu Yang, Chengting Yu, Lei Liu, Hanzhi Ma, Aili Wang, and Erping Li. Efficient ANN-guided distillation: Aligning rate-based features of spiking neural networks through hybrid block-wise replacement. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10025–10035, 2025.

Yibo Yang, Xiaojie Li, Motasem Alfarra, Hasan Hammoud, Adel Bibi, Philip Torr, and Bernard Ghanem. Towards interpretable deep local learning with successive gradient reconciliation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56196–56215, 2024.

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, 2023.

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 *The Twelfth International Conference on Learning Representations*, 2024.

Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11062–11070, 2021.

Hanle Zheng, Zhong Zheng, Rui Hu, Bo Xiao, Yujie Wu, Fangwen Yu, Xue Liu, Guoqi Li, and Lei Deng. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15(1):277, 2024.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng YAN, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

A Implementation and Training Details

For the teacher-conditioned protocol, Eq. 9 supplies noisy target semantics during training, whereas evaluation begins from $q^{(0)} = 0$ and follows Eq. 11. The ImageNet-1K implementation instead uses prediction conditioning during both training and inference: the first unit receives zero context, and every later unit receives a detached semantic prediction from its predecessor. In all settings, gradients terminate at unit boundaries and no result in Tab. 1 uses a subsequent global-STBP fine-tuning stage.

B Comparison Protocol and Resource Measurements

Table 3 reports seed 42 for both backbones. The SpikFormer NeuroProp run evaluates three semantic update schedules and uses the best result from the linear schedule from $\beta = 0.1$ to 0.9. These reduced-budget results must not be combined with the longer training schedules of Table 1.

Training throughput is the median over epochs 2–150 of the seed-42 run. Timings include data loading and device transfer. SpikFormer NeuroProp training throughput is reconstructed from epoch times and the 49,664 samples processed per epoch with `drop_last=True`. Peak memory uses the maximum `torch.cuda.max_memory_allocated` value recorded during the same run.

The SpikFormer checkpoint/compression runs use PyTorch 2.6.0/CUDA 11.8 and SpikingJelly 2.0.0rc1, whereas the other SpikFormer runs use PyTorch 1.10.1/CUDA 11.1 and SpikingJelly 0.0.0.0.15. Consequently, SpikFormer throughput ratios combine algorithmic and software effects.

Table 6: Full-budget training configurations for the main classification results in Table 1.

Configuration	**CIFAR-100**	**CIFAR10-DVS**	**ImageNet-1K**
Optimization			
Batch size	256	64	4096
Learning rate	0.0032	0.002	0.0005
Epochs	800	400	100
Time steps (T)	4 / 8	10 / 16	1 / 4 / 4
Optimizer	AdamW	AdamW	AdamW
Weight decay	0.001	0.0004	0.05
Gradient clipping	1.0	1.0	1.0
Warmup epochs	30	20	10
Scheduler	Cosine	Cosine	Cosine
Architecture			
Input size	32×32	128×128	224×224
Channels	3	2	3
Embedding dimension	384	384	512
Depth	4	4	4 / 8
Heads	12	12	8
MLP ratio	4	4	4
Objectives			
λ_{CE}	2.0	2.0	2.0
λ_{MSE}	1.0	0.5	1.0
λ_{FR}	–	–	5.0
Label smoothing	0.0	–	0.1
Augmentation			
RandAugment	(2,15)	–	RandAug
Mixup	1.2	–	–
CutMix	1.0	–	–
Random erasing	0.45	–	0.2
Color jitter	–	–	0.4
Hardware and execution			
GPUs	1	1	4
Pipeline parallelism	No	No	No / Yes / Yes

All completed SDTv3 runs use the latter common stack on the same A800. SDTv3 contains stateless MultiSpike activations rather than recurrent LIF neurons; its OTTT row is therefore an explicitly labeled instantaneous-loss, accumulated-update adapter and is not evidence about eligibility-trace OTTT on recurrent neurons.

C AdaIP Modulation Analysis

To further understand the role of AdaIP in shaping internal representations, we perform a detailed analysis of its token-level modulation behavior, as shown in Fig. 6.

We first examine whether the modulation induced by AdaIP is semantically meaningful rather than stochastic. For each class, we compute the average modulation across all samples and perform pairwise statistical comparisons among all class pairs. As shown in Fig. 6(a), the majority of class pairs exhibit statistically significant differences in modulation patterns (upper triangular matrix), accompanied by substantial effect sizes (lower triangular matrix). This result demonstrates that AdaIP does not introduce arbitrary heterogeneity; instead, it produces class-dependent and discriminative modulation, indicating that semantic signals are consistently translated into structured neuronal control.

We then analyze how this modulation is distributed across token dimensions. Fig. 6(b) shows that the modulation is highly non-uniform: only a subset of token positions exhibits strong responses

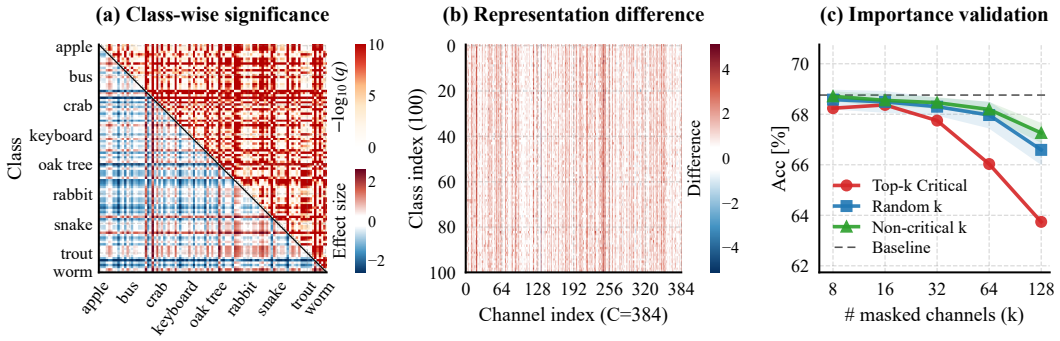


Figure 6: **AdaIP channel modulation.** (a) Class-wise significance and effect size. (b) Non-uniform modulation over embedding channels. (c) Masking strongly modulated channels causes the largest accuracy degradation.

across classes. This suggests that AdaIP selectively amplifies specific semantic channels rather than uniformly scaling all features, hinting at an implicit mechanism of feature prioritization.

To verify whether these highly modulated tokens are indeed more informative, we conduct a masking-based importance analysis. Token importance is defined by the class-averaged modulation magnitude, and channels are ranked accordingly. As shown in Fig. 6(c), masking the most important tokens (Top-k Critical) leads to a significantly larger performance drop than random masking, while masking the least important tokens (Non-critical k) has the smallest impact. The consistent ordering 'Top-k Critical $\leq$ Random k $\leq$ Non-critical k' confirms that AdaIP-derived importance scores are predictive of functional relevance.

Taken together, these results establish a clear link between semantic conditioning and intrinsic neuronal modulation. From a biological perspective, AdaIP resembles neuromodulatory mechanisms that selectively gate neuron populations based on task context. From a statistical perspective, it induces a structured, class-dependent transformation that concentrates information into a subset of critical dimensions. This unifies biological plausibility with statistical efficiency: AdaIP does not merely stabilize local learning, but actively reorganizes representation geometry to enhance discriminability at the token level.

D Pipeline-Parallel Efficiency

To evaluate the efficiency of NeuroProp's pipeline-parallel training, we compare it with STBP-based multi-GPU training under a fixed memory budget. Specifically, we constrain the average peak memory per GPU to approximately 30 GB to ensure comparable hardware usage across methods. We consider three model configurations with increasing depth, including 16-layer and 24-layer models trained on 4 GPUs, and a 32-layer model trained on 6 GPUs. For all settings, we measure the wall-clock time required to complete the same number of training epochs under identical optimization configurations.

As shown in Fig. 7, NeuroProp consistently reduces training time compared to STBP across all configurations, achieving $1.30\times$, $1.59\times$, and $1.89\times$ speedups on 16-, 24-, and 32-layer models, respectively. Notably, the efficiency gain becomes more pronounced as model depth increases, indicating that NeuroProp scales more effectively in deeper architectures, where global synchronization overhead in STBP becomes increasingly dominant.

This improvement stems from the structural decoupling introduced by NeuroProp. On one hand, the reduced memory footprint enables more flexible resource allocation under a fixed hardware budget, allowing more efficient training configurations. On the other hand, and more critically, the elimination of cross-unit backward dependencies enables pipeline execution across GPUs without global synchronization, thereby significantly reducing device idle time. Together, these factors lead to improved hardware utilization efficiency, and the increasing speedup with depth further highlights the advantage of removing globally coupled computation in large-scale spiking Transformer training.

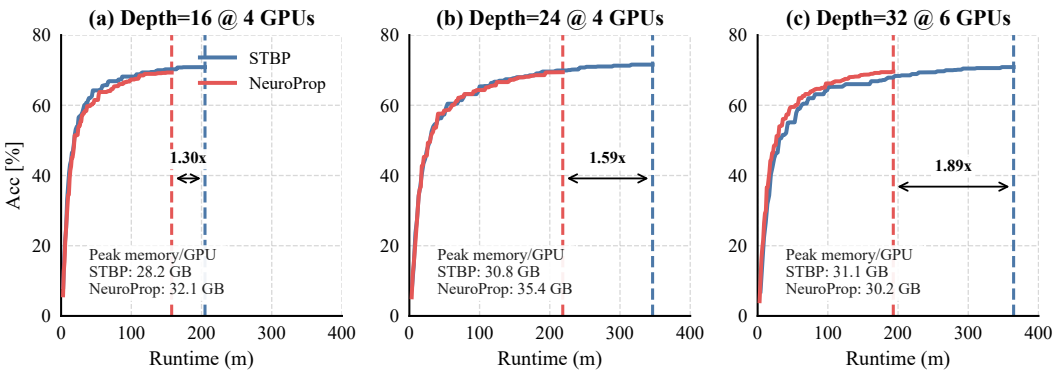


Figure 7: **Pipeline efficiency under a limited memory budget.** NeuroProp retains STBP locally while removing cross-unit backward synchronization.

Table 7: Final average accuracy on Split CIFAR-100 continual learning. Pre-trained: ×, trained from scratch; ✓, initialized from an STBP-pretrained checkpoint. Bold marks the highest value within each initialization group; shading identifies NeuroProp.

Method	Pre-trained	Label type	Mean±std (%) ↑	Max (%) ↑
STBP	×	One-hot	48.23 ± 0.38	49.10
NeuroProp	×	One-hot	49.54 ± 0.43	50.31
NeuroProp	×	Prototype	$\mathbf{51.05 \pm 0.43}$	**51.74**
STBP	✓	One-hot	64.20 ± 0.26	64.50
NeuroProp	✓	One-hot	54.79 ± 0.19	55.03
NeuroProp	✓	Prototype	$\mathbf{66.78 \pm 0.35}$	**67.17**

E Performance on Complex Tasks

E.1 Continual Learning on Split CIFAR-100

We evaluate whether NeuroProp's localized updates can support sequential adaptation on Split CIFAR-100 (Rebuffi et al., 2017) with SpikFormer. Table 7 compares training from scratch and initialization from an STBP-pretrained checkpoint; NeuroProp additionally uses either one-hot or prototype-based supervision (Cai et al., 2025).

From scratch, one-hot NeuroProp improves final average accuracy from STBP's $48.23 \pm 0.38\%$ to $49.54 \pm 0.43\%$, and prototype supervision further raises it to $51.05 \pm 0.43\%$. With STBP pretraining, one-hot NeuroProp falls to $54.79 \pm 0.19\%$ versus $64.20 \pm 0.26\%$ for STBP, whereas prototypes recover performance to $66.78 \pm 0.35\%$, exceeding STBP by 2.58 points. Prototype-based NeuroProp therefore achieves the highest mean in both initialization settings.

These results demonstrate that the inter-module coupling in NeuroProp remains largely unaffected throughout the iterative data updates in continual learning. However, the structured relationships inherited from the pretrained model may interfere with the subsequent optimization of NeuroProp. Nevertheless, when more informative class-level representations are employed as supervision, the model is still able to achieve competitive performance.

E.2 Dense Optical-Flow Prediction

We further evaluate NeuroProp on optical flow estimation over the DSEC dataset (Gehrig et al., 2021), a dense spatiotemporal prediction task that requires fine-grained spatial and temporal coordination (Tian & Andrade-Cetto, 2024; Guo et al., 2025). As shown in Tab. 5, NeuroProp achieves an AEE of 2.349, an AAE of 7.176, and a 3PE of 15.833%, compared to 2.138, 5.882, and 13.967% for STBP. We attribute the gap to the reduced global coordination inherent in decoupled training,

which is more critical for dense low-level prediction. A qualitative visualization is provided in Fig. 8. Overall, these results suggest that NeuroProp generalizes to complex spatiotemporal tasks, while also highlighting an important direction for improving local learning in structured prediction settings.

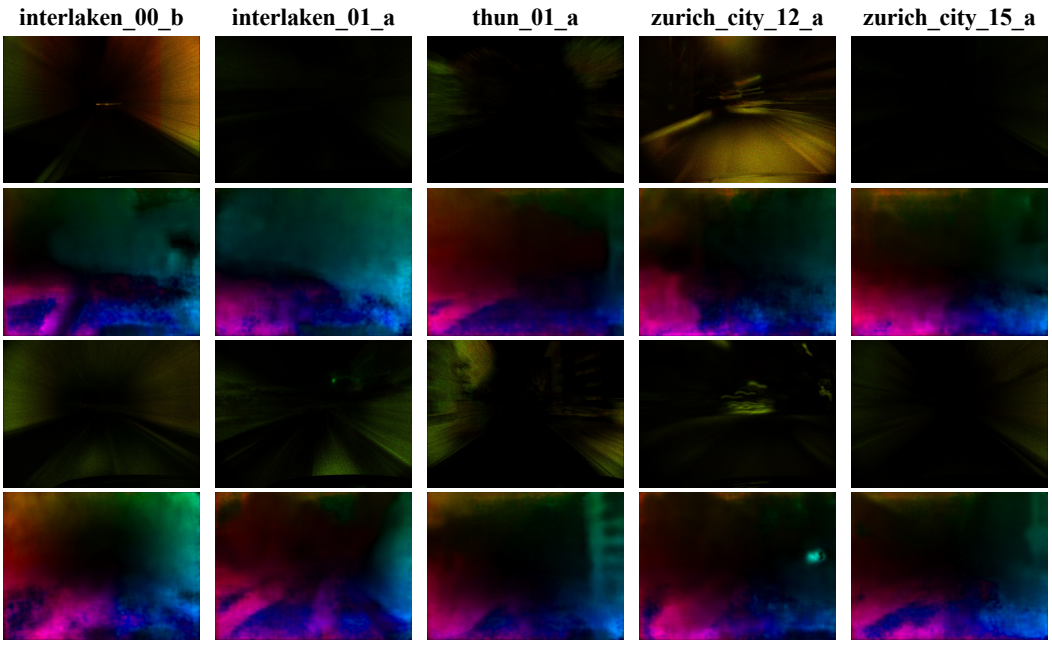


Figure 8: **DSEC optical flow with NeuroProp.** Event frames are shown above their corresponding flow estimates.

F LIMITATIONS

Computational scope. NeuroProp bounds the depth-wise activation graph by the largest unit, but retains the complete temporal trajectory required by STBP inside that unit. Its memory therefore still grows with T and unit size. Total device memory also includes the tokenizer, auxiliary heads, parameters, optimizer states, and communication buffers; consequently, the end-to-end reduction can be smaller when these components dominate. Pipeline gains likewise depend on partition balance, inter-device communication, and the target hardware.

Optimization and task scope. Stopping gradients at unit boundaries removes end-to-end credit assignment and produces an accuracy trade-off relative to global STBP. Increasing batch size or depth is not guaranteed to improve accuracy, even when NeuroProp makes the configuration feasible. Reduced-batch global-STBP fine-tuning can recover part of this gap, but reconstructs the global backward graph and is therefore a hybrid refinement rather than a fully localized method. The remaining deficit on DSEC further indicates that dense, multi-scale prediction is more sensitive to global spatial coordination than classification; richer boundary signals or task-specific local objectives may be required.