OPTICAL-FLOW-FREE TWO-STREAM SPIKING NETWORKS WITH CROSS-DIMENSIONAL FUSION FOR ACTION RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) model temporal evolution through recurrent membrane dynamics, while action recognition also depends on clip-level context that varies more slowly than frame-to-frame motion. We propose CF-TS-SNN, an optical-flow-free two-stream SNN for RGB and event-based action recognition. The temporal pathway processes chronologically sampled frames or event windows, whereas the spatial pathway repeatedly encodes a representative clip summary using a structurally matched spiking backbone. A controlled routing study identifies additive interaction after the second backbone stage, followed by final fusion after the third stage, as the most effective configuration. Building on this routing, cross-dimensional fusion derives temporal, channel, and spatial weights from the final spatial feature to modulate the temporal representation. The same spatial feature also produces a threshold map from attention, activity, and normalized spatial entropy, enabling clip-level context to regulate spike generation in the final temporal stage. At 20 time steps, CF-TS-SNN achieves 98.15% accuracy on DVS128Gesture and 94.05% on UCF-50, improving the temporal-only baseline by 1.25 and 7.75 percentage points, respectively.

1 INTRODUCTION

Spiking neural networks (SNNs) communicate through discrete spikes and preserve temporal history in membrane potentials. This recurrent state is well suited to video, where evidence accumulates across successive observations. Surrogate-gradient learning has made deep SNNs practical to optimize Neftci et al. (2019); Wu et al. (2018; 2019); Li et al. (2021); Deng et al. (2022); Zheng et al. (2021); Fang et al. (2021), and spiking Transformers have extended their temporal receptive field Zhou et al. (2023); Yao et al. (2023a). Nevertheless, visual SNN research remains concentrated on static images and neuromorphic event streams, with conventional RGB action recognition receiving less attention.

An action is identified not only by how the image changes, but also by cues that remain comparatively stable within a clip, such as pose, manipulated objects, and scene layout. When both sources are processed by one temporal pathway, persistent context and transient change compete within the same recurrent state. An explicit context pathway can preserve these complementary cues and make them available throughout temporal feature extraction, which is particularly useful for classes with similar motion patterns but different objects, poses, or environments.

Classical two-stream models separate appearance from motion by pairing RGB frames with dense optical flow Simonyan & Zisserman (2014). Subsequent architectures avoid explicit flow through 3D convolutions, dual-rate pathways, temporal shifts, and video Transformers Carreira & Zisserman (2017); Feichtenhofer et al. (2019); Lin et al. (2019); Bertasius et al. (2021); Tong et al. (2022). Recent spiking methods either process sampled frames in a single SNN or combine heterogeneous ANN and SNN branches Yu et al. (2025); Xiao et al. (2025). Less explored is how a persistent clip summary should enter temporal spike processing when both representations are produced by matched SNN backbones. The central question is therefore not merely whether a second pathway is useful, but where its context should be injected and whether that context should influence only the final representation or also the spike dynamics that form it.

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.

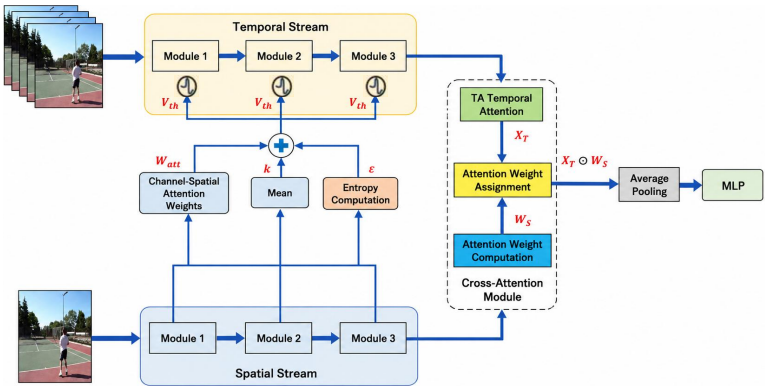

Figure 1: Conceptual overview of CF-TS-SNN. The temporal pathway processes T time-indexed observations, while the spatial pathway repeatedly encodes a clip summary. The final spatial representation provides cross-dimensional modulation and the adaptive threshold used in temporal Block 3. Figure 2 further details the shared SpikeResNet stages and the intermediate and final routing patterns, while Eq. 1 gives the exact computation.

Figure 1 summarizes CF-TS-SNN. The temporal stream receives chronologically sampled RGB frames or consecutive event windows, whereas the spatial stream repeats a center RGB frame or a full-clip event summary over the same number of simulation steps. Both pathways adopt the same SpikeResNet architecture. Their stage-wise features therefore have compatible shapes and can be combined directly without modality-specific projections or a separate high-capacity encoder.

The two streams interact at both representation and neuronal levels. An intermediate additive connection injects the spatial feature before the final temporal stage, allowing context to influence temporal feature formation. The independently propagated spatial output then generates temporal, channel, and spatial weights for the completed temporal representation and is retained through residual fusion. In parallel, the same output determines the firing threshold in the final temporal stage through activity, entropy, and channel–spatial attention.

Our contributions are:

- We develop an optical-flow-free, homogeneous two-stream SNN in which a temporally sampled sequence and a repeated clip summary are encoded by stage-aligned spiking backbones.
- We introduce spatial-to-temporal cross-dimensional fusion and a context-dependent threshold regulator. Both are generated from the final spatial feature, but they act at different points: one modulates the completed temporal representation and the other changes the spikes that produce it.
- We analyze stream choice, fusion operator and position, repeated interaction, attention axes, and threshold terms on RGB and event data. CF-TS-SNN achieves 98.15% on DVS128Gesture and 94.05% on UCF-50 at 20 time steps.

2 Related Work

2.1 Directly Trained Spiking Networks

Early deep SNNs relied on convolutional spike processing or ANN-to-SNN conversion Xu et al. (2018); Cao et al. (2015); Kim et al. (2020); Sengupta et al. (2019). Surrogate gradients enabled end-to-end training Neftci et al. (2019); Wu et al. (2018; 2019), while differentiable spike functions, temporal objectives, residual connections, temporal normalization, and learnable membrane constants improved optimization and representation quality Li et al. (2021); Deng et al. (2022); Vicente-Sola et al. (2022); Zheng et al. (2021); Fang et al. (2021). Spiking Transformers further combine attention with spike-driven computation Zhou et al. (2023); Yao et al. (2023a). Rather than

redesigning the backbone, our work focuses on how a matched spatial pathway can guide temporal features and firing thresholds.

2.2 Spiking Video Recognition

Event cameras record brightness changes asynchronously and are well suited to SNN processing Gallego et al. (2022); Amir et al. (2017); Shrestha & Orchard (2018); Meng et al. (2022). RGB video additionally contains dense appearance, camera motion, and background variation. SVFormer directly trains a spiking Transformer on sampled video sequences Yu et al. (2025), whereas ReSpike couples an ANN key-frame branch with an SNN residual-frame branch Xiao et al. (2025). In contrast, both pathways in our model are directly trained SNNs, and the temporal input is used without residual-frame construction.

2.3 Two-Stream Fusion and Adaptive Spiking Dynamics

Two-stream recognition separates appearance from motion Simonyan & Zisserman (2014), while subsequent work shows that fusion position affects cross-pathway information exchange Wang et al. (2016); Feichtenhofer et al. (2016). SlowFast and TSM model temporal interactions without dense optical flow Feichtenhofer et al. (2019); Lin et al. (2019). Channel–spatial attention Hu et al. (2018); Woo et al. (2018) and attention SNNs Yao et al. (2023b) further enable feature selection across spatial, channel, and temporal dimensions, while adaptive neuronal parameters regulate spiking dynamics Fang et al. (2021). Our method unifies these ideas by deriving both axis-wise temporal modulation and threshold regulation from the same spatial SNN representation.

3 Method

3.1 Design Overview

Figure 1 presents the complete spatial-to-temporal guidance mechanism, while Fig. 2 shows the backbone structure and the routing alternatives evaluated in our fusion study. The complete model adopts the two-stage topology in Fig. 2(b): the Block-2 features are combined by addition, the spatial pathway continues independently through Block 3, and its final feature provides both the cross-dimensional weights and the adaptive threshold:

$$\begin{aligned} \widehat{F}_t^{(2)} &= F_t^{(2)} + F_s^{(2)}, \\ F_s^{(3)} &= B_s^{(3)}(F_s^{(2)}), \\ W_s &= \mathcal{G}(F_s^{(3)}), \qquad V_{th} = \mathcal{R}(F_s^{(3)}), \\ F_t^{(3)} &= B_t^{(3)}(\widehat{F}_t^{(2)}; V_{th}), \\ F &= W_s \odot F_t^{(3)} + F_s^{(3)}. \end{aligned} \tag{1}$$

Here, $\mathcal{G}$ denotes cross-dimensional weight generation and $\mathcal{R}$ denotes threshold regulation. The spatial pathway first produces $F_s^{(3)}$, from which W_s and V_{th} are computed. Temporal Block 3 is then evaluated using the regulated threshold, after which its output $F_t^{(3)}$ is modulated by W_s and combined residually with $F_s^{(3)}$. The adaptive threshold is applied only inside temporal Block 3, whereas cross-dimensional weighting acts on the completed block output.

3.2 Input Construction and Spiking Backbone

The network operates for $T = 20$ time steps. For RGB video, each clip is divided into T temporal segments, and one frame is sampled from each segment while preserving chronological order. These frames form the temporal input. The spatial stream repeats the center frame for the same T steps, providing a fixed RGB reference throughout the simulation.

For DVS data, the event clip is partitioned into T consecutive windows, and the events within each window are accumulated for the temporal stream. The spatial stream aggregates the complete clip

into one event summary and repeats it over time, providing a persistent description of clip-level spatial activity. In both modalities, the two inputs share the same temporal length and spatial resolution, so their features can interact stage by stage.

Repeating the clip summary does not introduce an additional motion representation. Its purpose is to keep the same contextual reference available throughout the T simulation steps. Because the two pathways share temporal length and stage-wise tensor shapes, intermediate and final interactions can be implemented directly, without temporal interpolation or modality-specific alignment layers.

Both pathways adopt the same three-block SpikeResNet architecture, whose stage structure is expanded in the upper panel of Fig. 2. Each block stacks residual spiking convolutional units with BNTT normalization and spiking activations, while the stage transitions increase the channel width and reduce the spatial resolution when required. The two pathways use the same block design but maintain independent parameters. For a leaky integrate-and-fire neuron,

$$u_i^t = \lambda u_i^{t-1}(1 - s_i^{t-1}) + I_i^t, \tag{2}$$

$$s_i^t = H(u_i^t - V_{th,i}^t), \tag{3}$$

where λ is the membrane decay, I_i^t is the synaptic input, and $H(\cdot)$ is the Heaviside function. During training, its derivative is replaced by the surrogate function used in STBP Wu et al. (2018). Because the reset term depends on the previous spike, changing the threshold affects both the current spike decision and the membrane state available at the next step. We therefore apply the context-dependent threshold inside temporal Block 3 rather than only reweighting its output.

3.3 Fusion Operators and Positions

Let $F_t^{(\ell)}, F_s^{(\ell)} \in \mathbb{R}^{T\times C_\ell \times H_\ell \times W_\ell}$ denote aligned temporal and spatial features after Block ℓ. We compare four basic operators:

$$\mathcal{F}_{\text{add}}(X,Y) = X + Y, \qquad \mathcal{F}_{\max}(X,Y) = \max(X,Y), \tag{4}$$

$$\mathcal{F}_{\text{cat}}(X,Y) = [X;Y], \qquad \mathcal{F}_{\text{conv}}(X,Y) = \text{Conv}([X;Y]). \tag{5}$$

Addition and maximum preserve the feature dimension and add no parameters. Concatenation increases the input dimensionality of the subsequent layer, while convolutional fusion introduces an additional learned projection.

The lower part of Fig. 2 separates the effect of an intermediate interaction from that of preserving a final spatial output. In Fig. 2(a), the single-stage layout fuses the two Block 2 features and sends the merged representation through temporal Block 3:

$$\widehat{F}_t^{(2)} = \mathcal{F}_m(F_t^{(2)}, F_s^{(2)}), \qquad F_{ts} = B_t^{(3)}(\widehat{F}_t^{(2)}). \tag{6}$$

This layout allows spatial context to influence the final temporal abstraction, but the spatial pathway terminates at the intermediate fusion and therefore contributes no independent high-level feature to the classifier.

Figure 2(b) retains the intermediate interaction while allowing the spatial pathway to continue through its own Block 3. The two high-level outputs are then combined by a final operator:

$$\widehat{F}_t^{(2)} = \mathcal{F}_e(F_t^{(2)}, F_s^{(2)}), \tag{7}$$

$$F_t^{(3)} = B_t^{(3)}(\widehat{F}_t^{(2)}), \qquad F_s^{(3)} = B_s^{(3)}(F_s^{(2)}), \tag{8}$$

$$F_{ts} = \mathcal{F}_l(F_t^{(3)}, F_s^{(3)}), \tag{9}$$

where $\mathcal{F}_e$ and $\mathcal{F}_l$ denote the intermediate and final operators, respectively. This decomposition clarifies the role of the two interactions: the intermediate operator changes the input to temporal Block 3, whereas the final operator combines the completed temporal representation with the independently propagated spatial representation. The routing study selects addition for $\mathcal{F}_e$. In the complete model, the topology of Fig. 2(b) is retained, but the uniform $\mathcal{F}_l$ is replaced by the spatially conditioned computation in Eq. 1.

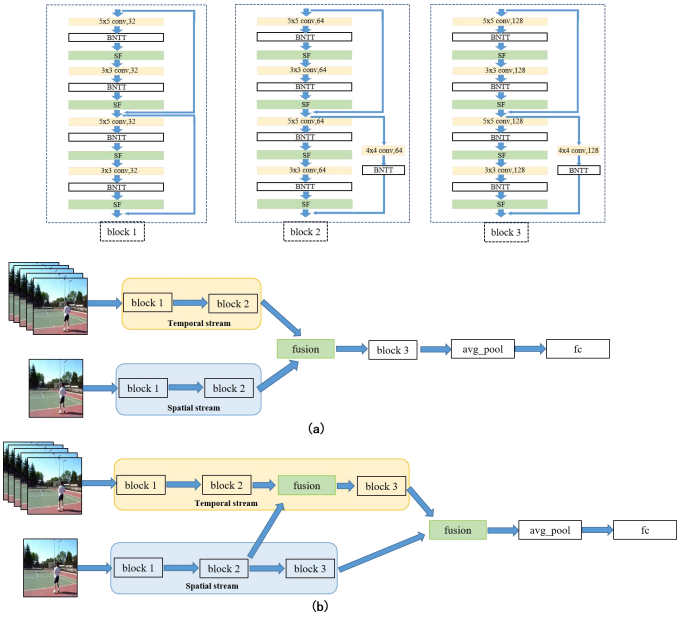

Figure 2: Backbone stages and fusion layouts used in the routing study. The upper panel expands the three-block SpikeResNet shared in architecture by the temporal and spatial pathways; each stage contains residual spiking convolutional units with BNTT normalization and spiking activations. (a) Single-stage intermediate fusion combines $F_t^{(2)}$ and $F_s^{(2)}$ after Block 2, and temporal Block 3 produces the representation used for classification. (b) Two-stage fusion retains the same intermediate interaction while propagating $F_s^{(2)}$ through an independent spatial Block 3; the resulting $F_t^{(3)}$ and $F_s^{(3)}$ undergo a second interaction before pooling and classification. The complete model follows topology (b), uses addition at the intermediate point, and replaces the uniform final operator with the cross-dimensional fusion in Eq. 1.

3.4 Cross-Dimensional Spatial-to-Temporal Fusion

Uniform final addition applies the same interaction at every simulation step, channel, and location. We instead generate three axis-wise weights from $F_s^{(3)}$. Let $\mathrm{Avg}_{\mathcal{D}}$ and $\mathrm{Max}_{\mathcal{D}}$ denote pooling over the axes in $\mathcal{D}$:

$$A_T = \sigma\Big(g_T\Big([\mathrm{Avg}_{CHW}(F_s^{(3)}); \mathrm{Max}_{CHW}(F_s^{(3)})]\Big)\Big), \tag{10}$$

$$A_C = \sigma\Big(g_C\Big([\mathrm{Avg}_{THW}(F_s^{(3)}); \mathrm{Max}_{THW}(F_s^{(3)})]\Big)\Big), \tag{11}$$

$$A_S = \sigma\Big(g_S\Big([\mathrm{Avg}_{TC}(F_s^{(3)}); \mathrm{Max}_{TC}(F_s^{(3)})]\Big)\Big). \tag{12}$$

Here, $A_T \in \mathbb{R}^{T\times1\times1\times1}$, $A_C \in \mathbb{R}^{1\times C\times1\times1}$, and $A_S \in \mathbb{R}^{1\times1\times H\times W}$. The mappings g_T, g_C, and g_S operate on the corresponding pooled descriptors. All three weights are generated in parallel from $F_s^{(3)}$; $F_t^{(3)}$ is not used to compute them. Although the spatial input is repeated, its SNN feature can evolve over time because of recurrent membrane dynamics. Thus, A_T reflects the time-dependent reliability of the persistent encoding, not motion in the original clip.

The three branches have complementary scopes. A_T describes how strongly the persistent encoding should contribute at each simulation step, A_C emphasizes feature channels associated with the clip summary, and A_S localizes spatial support. Their factorized construction produces a full $T \times C \times H \times W$ modulation tensor by broadcasting, without forming a dense pairwise affinity matrix over all spatiotemporal elements.

This factorization also keeps the guidance computation compact. Before broadcasting, the three branches store only $T + C + HW$ coefficients rather than a dense affinity matrix over all $TCHW$ elements. Beyond the lightweight descriptor mappings, applying W_s requires only element-wise

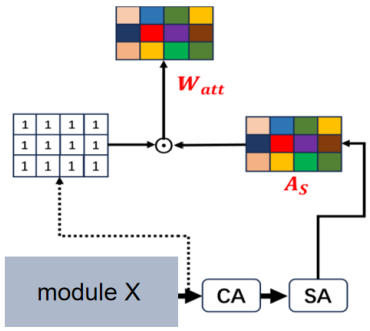

Figure 3: Construction of the channel–spatial term W_{att}. "Module X" denotes $F_s^{(3)}$; CA and SA generate A_C and A_S. After broadcasting to a common channel–spatial shape, their element-wise product is normalized to obtain W_{att}.

operations whose cost grows linearly with the feature size. The module therefore changes how the two streams interact without increasing the depth of either backbone.

After broadcasting to a common shape, the weights are multiplied and applied to the output of temporal Block 3:

$$W_s = A_T \odot A_C \odot A_S, \qquad \widetilde{F}_t^{(3)} = W_s \odot F_t^{(3)}. \tag{13}$$

The direct spatial path is then restored by residual addition:

$$F = \widetilde{F}_t^{(3)} + F_s^{(3)}. \tag{14}$$

Accordingly, $F_s^{(3)}$ has two distinct roles: it generates the modulation weights and contributes directly to the final representation. The residual path prevents the context stream from being reduced to a gate and preserves information that may be weakly represented in $F_t^{(3)}$.

3.5 Attention-Guided Threshold Regulation

Cross-dimensional weighting acts on the output of temporal Block 3, whereas threshold regulation acts within that block. The spatial feature sequence is therefore computed first and converted into a threshold map for its temporal neurons.

We first average the spatial feature over time and obtain an activity-dependent term:

$$\bar{F}_s = \mathrm{Avg}_T(F_s^{(3)}), \qquad k = \sigma(\bar{F}_s). \tag{15}$$

For each channel c, let Ω contain its $N = H \times W$ spatial responses. The normalized response distribution and its channel-wise spatial entropy are

$$p_{c,i} = \frac{\bar{F}_s(c,i) + \delta}{\sum_{j\in\Omega}(\bar{F}_s(c,j) + \delta)}, \tag{16}$$

$$\epsilon_c = -\frac{1}{\log N} \sum_{i\in\Omega} p_{c,i} \log(p_{c,i} + \delta), \tag{17}$$

where δ is a small numerical constant. This normalization gives $\epsilon_c \in [0,1]$: spatially concentrated responses have lower entropy and diffuse responses have higher entropy. The vector $\epsilon \in \mathbb{R}^{C\times 1\times 1}$ is broadcast over the spatial dimensions.

The attention term reuses the channel and spatial weights from Eqs. 11 and 12. After broadcasting A_C and A_S to $C \times H \times W$, we compute

$$Q = A_C \odot A_S, \qquad W_{att} = \frac{Q - \min(Q)}{\max(Q) - \min(Q) + \delta}, \tag{18}$$

where the extrema are taken over the broadcast channel–spatial map. Appendix Fig. 3 illustrates this branch. The regulated threshold is

$$V_{th} = V_{TH} \odot \mathrm{clip}(k + \epsilon - W_{att}, \tau_{\min}, \tau_{\max}), \tag{19}$$

Table 1: Single-stream and single-stage additive fusion accuracy (%).

Model	DVS128G.	UCF-50
Spatial stream	92.40	85.20
Temporal stream	96.90	86.30
Two-stream, single-stage add	**97.60**	**87.20**

where V_{TH} is the base threshold and $0 < \tau_{\min} < \tau_{\max}$ keeps the multiplier positive and bounded. After broadcasting, k, ϵ, and W_{att} share the same channel–spatial support. Their signs encode different regulation roles: W_{att} lowers the threshold at locations supported by the spatial representation, while k and ϵ raise it for persistently active or spatially diffuse responses. The resulting $C \times H \times W$ map is broadcast over the T simulation steps and applied only inside temporal Block 3.

Cross-dimensional fusion and threshold regulation act at complementary points in the computation. The former selects the output of the final temporal block, while the latter changes the spike decisions and membrane trajectory that generate that output. Classification is performed as

$$z = \mathrm{MLP}(\mathrm{GAP}_{HW}(\mathrm{Avg}_T(F))), \tag{20}$$

where z denotes the action logits. The network is trained end to end with cross-entropy loss and STBP.

4 Experiments

4.1 Datasets and Experimental Setting

DVS128Gesture contains event recordings from 11 gesture classes and emphasizes sparse motion changes. **UCF-50** contains 50 RGB action classes with variations in viewpoint, scale, background, and actor appearance. We use the two datasets to examine the same architecture under distinct input statistics: event windows expose brightness changes directly, whereas RGB clips retain appearance together with camera and background variation.

Controlled protocol. All variants within a dataset use the same input construction, preprocessing, SpikeResNet architecture, classifier, and optimization settings; only the fusion or regulation component under study is changed. The two pathways are optimized jointly with cross-entropy loss and STBP. Models are implemented in PyTorch and trained on one NVIDIA RTX 4090 GPU. All reported comparisons use $T = 20$ time steps and top-1 classification accuracy. Neither optical flow nor residual-frame preprocessing is used.

4.2 Single-Stream and Fusion Study

Table 1 compares the individual pathways with the single-stage additive routing in Fig. 2(a). In this configuration, $F_s^{(2)}$ is added to $F_t^{(2)}$ and no independent spatial Block 3 is retained. The temporal stream performs better on both datasets, indicating that the time-indexed sequence carries the dominant action signal under this backbone. The spatial stream remains discriminative on its own. Intermediate addition raises accuracy from 96.90% to 97.60% on DVS128Gesture and from 86.30% to 87.20% on UCF-50, showing that the repeated summary contributes complementary information before the final temporal stage.

Table 2 compares four intermediate-fusion operators under the same two-stream backbone. Addition gives the highest accuracy on both datasets and adds no fusion parameters. Maximum keeps only the larger aligned response, whereas concatenation changes the input dimensionality of temporal Block 3. A learned projection after concatenation does not recover the loss. We therefore use addition in the subsequent routing study.

4.3 Fusion Position and Repeated Interaction

The lower panels of Fig. 2 distinguish two effects. The single-stage layout in Fig. 2(a) tests whether the summary is useful when injected only before temporal Block 3. The two-stage layout in Fig. 2(b)

Table 2: Effect of the single-stage intermediate fusion operator (accuracy, %).

Fusion	Extra params.	DVS128G.	UCF-50
Add	No	**97.60**	**87.20**
Max	No	97.50	86.60
Concat.	Yes	96.50	86.90
Concat. + Conv	Yes	95.40	86.30

Table 3: Two-stage fusion combinations (accuracy, %).

Intermediate + Final	DVS128G.	UCF-50
Add + Add	**97.90**	**87.80**
Add + Max	96.50	87.20
Add + Cat	97.50	86.20
Max + Add	97.20	86.30
Max + Max	97.20	86.80
Max + Cat	97.00	87.10
Cat + Add	97.20	87.10
Cat + Max	97.20	87.20
Cat + Cat	97.50	86.80

keeps that injection and additionally preserves an independent spatial Block 3, allowing the summary to influence temporal feature formation and to contribute a high-level spatial representation at the final interaction.

Table 3 evaluates all nine intermediate–final operator combinations. Add+Add is the only configuration that ranks first on both datasets, reaching 97.90% on DVS128Gesture and 87.80% on UCF-50. Relative to single-stage addition, retaining the independent spatial output improves accuracy by 0.30 and 0.60 points, respectively. Performance also depends on the operator pair rather than either operator alone; for example, Add+Cat falls to 86.20% on UCF-50. These results motivate a fixed additive intermediate route and a more selective final interaction.

4.4 Cross-Dimensional Fusion and Threshold Ablations

Table 4 evaluates the proposed components after intermediate addition has been fixed. The first row retains threshold regulation but replaces cross-dimensional final fusion with uniform addition; it therefore differs from Add+Add in Table 3, which uses the base threshold. On UCF-50, this substitution reduces accuracy from 94.05% to 89.53%. Removing A_T, A_C, and A_S decreases accuracy by 0.29, 1.65, and 2.85 points, respectively. The ordering shows that spatial localization contributes most strongly, while channel and temporal selection provide additional refinement.

When cross-dimensional fusion is retained, using the base threshold reduces UCF-50 accuracy from 94.05% to 93.15%. Removing W_{att}, k, or ϵ yields 93.22%, 93.60%, and 93.68%, corresponding to drops of 0.83, 0.45, and 0.37 points. The attention term therefore provides the largest individual contribution within threshold regulation, while activity and entropy supply complementary control. On DVS128Gesture, the full configuration also attains the highest accuracy, although the component-wise margins are smaller than those observed on UCF-50.

4.5 Stage-Wise and Modality-Level Analysis

Table 5 summarizes the main configurations examined across the routing and component studies. Intermediate addition improves both datasets, and preserving the independent spatial output for a second interaction provides a further gain. Replacing that uniform final interaction with cross-dimensional weighting and enabling threshold regulation raises UCF-50 accuracy from 87.80% to 94.05%; the corresponding controlled variants in Table 4 separate the two effects. Relative to the temporal-only baseline, CF-TS-SNN reduces classification error by 40.3% on DVS128Gesture and 56.6% on UCF-50. Here, relative error reduction is computed as $(e_{\text{base}} - e_{\text{model}})/e_{\text{base}}$, where $e = 1 - \text{accuracy}$.

Table 4: Ablation results (accuracy, %). CDF denotes cross-dimensional fusion and TR denotes threshold regulation; all variants retain intermediate addition.

Variant	DVS128G.	UCF-50
Uniform final addition (TR retained)	97.90	89.53
Full w/o A_T	98.10	93.76
Full w/o A_C	97.90	92.40
Full w/o A_S	97.96	91.20
CDF with base threshold	98.01	93.15
Full w/o W_{att}	98.01	93.22
Full w/o k	97.92	93.60
Full w/o ϵ	97.99	93.68
Full CF-TS-SNN	**98.15**	**94.05**

Table 5: Stage-wise gain over the temporal-only baseline. Error reduction is relative.

	DVS128G.		UCF-50	
Stage	Acc.	Err. red.	Acc.	Err. red.
Temporal only	96.90	–	86.30	–
Single-stage add	97.60	22.6%	87.20	6.6%
Two-stage add	97.90	32.3%	87.80	10.9%
Full CF-TS-SNN	**98.15**	**40.3%**	**94.05**	**56.6%**

Table 6: Comparison with representative neuromorphic methods on DVS128Gesture.

Method	Steps	Acc. (%)
TrueNorth Amir et al. (2017)	60	91.77
SLAYER Shrestha & Orchard (2018)	60	93.40
STBP Wu et al. (2018)	60	93.42
tdBN Zheng et al. (2021)	40	96.90
PLIF Fang et al. (2021)	20	97.60
Single-stage two-stream (ours)	20	97.60
Full CF-TS-SNN	20	**98.15**

The larger UCF-50 gain is consistent with the different information carried by the two input types. Event representations emphasize brightness changes and contain less static image content, whereas RGB clips retain appearance together with camera motion and background variation. The persistent spatial pathway can therefore provide stronger disambiguating evidence in the RGB setting.

4.6 Comparison on DVS and Qualitative Analysis

Table 6 compares CF-TS-SNN with representative neuromorphic systems on DVS128Gesture. The complete model reaches 98.15% at 20 time steps, the highest accuracy reported in the table. Together with the controlled studies in Tables 1–4, the comparison supports the effectiveness of the complete architecture and its routing and regulation components.

5 Conclusion

We presented CF-TS-SNN, an optical-flow-free two-stream SNN for RGB and event-based action recognition. A temporal pathway encodes time-indexed observations, while an architecturally matched spatial pathway maintains a repeated clip summary. Their interaction combines intermediate additive injection, factorized cross-dimensional modulation, a direct spatial residual, and context-dependent regulation of the final temporal block. At 20 time steps, CF-TS-SNN achieves 98.15% on DVS128Gesture and 94.05% on UCF-50. The routing and component studies consistently identify the combined configuration as the most effective design, with particularly large gains on RGB video where persistent context complements motion-dependent evidence.

Reproducibility Statement

We provide detailed descriptions of the proposed architecture, training objectives, and experimental settings in the main paper. All datasets used in this work are publicly available, including DVS128Gesture and UCF-50. The implementation details, model configuration, and evaluation protocols are described to facilitate future reproduction and comparison.

AI use statement

In this work, generative AI tools were used only for limited language verification of the final manuscript, such as checking grammar and improving readability. Generative AI tools were not used for developing the proposed method, designing experiments, implementing algorithms, generating experimental results, analyzing data, or producing scientific claims.

References

A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7243–7252, 2017.

G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 813–824. PMLR, 2021.

Y. Cao, Y. Chen, and D. Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition, 2015. Volume 113, pages 54–66.

J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

S. Deng, Y. Li, S. Zhang, and S. Gu. Temporal efficient training of spiking neural network via gradient re-weighting. In *International Conference on Learning Representations*, 2022.

W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. 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.

C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1933–1941, 2016.

C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6202–6211, 2019.

G. Gallego, T. Delbrück, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022.

J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

S. Kim, S. Park, B. Na, and S. Yoon. Spiking-YOLO: Spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11270–11277, 2020.

Y. Li, Y. Guo, S. Zhang, S. Deng, Y. Hai, and S. Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

J. Lin, C. Gan, and S. Han. TSM: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7083–7093, 2019.

Q. Meng, M. Xiao, S. Yan, Y. Wang, Z. Lin, and Z.-Q. Luo. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12444–12453, 2022.

E. O. Neftci, H. Mostafa, and F. 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.

A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in Neuroscience*, 13:95, 2019.

S. B. Shrestha and G. Orchard. SLAYER: Spike layer error reassignment in time. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Z. Tong, Y. Song, J. Wang, and L. Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10078–10093, 2022.

A. Vicente-Sola, D. L. Manna, P. Kirkland, G. Di Caterina, and T. Bihl. Keys to accurate feature extraction using residual spiking neural networks. *Neuromorphic Computing and Engineering*, 2 (4):044001, 2022.

L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pp. 20–36, 2016.

S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. CBAM: Convolutional block attention module. In *European Conference on Computer Vision*, pp. 3–19, 2018.

Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12:331, 2018.

Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, and L. Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1311–1318, 2019.

S. Xiao, Y. Li, Y. Kim, D. Lee, and P. Panda. ReSpike: Residual frames-based hybrid spiking neural networks for efficient action recognition. *Neuromorphic Computing and Engineering*, 5(1), 2025.

Q. Xu, Y. Qi, H. Yu, J. Shen, H. Tang, and G. Pan. CSNN: An augmented spiking based framework with perceptron-inception. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 1646–1652, 2018.

M. Yao, J. Hu, Z. Zhou, L. Yuan, Y. Tian, B. Xu, and G. Li. Spike-driven transformer. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64043–64058, 2023a.

M. Yao, G. Zhao, H. Zhang, Y. Hu, L. Deng, Y. Tian, B. Xu, and G. Li. Attention spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023b.

L. Yu, L. Huang, C. Zhou, H. Zhang, Z. Ma, H. Zhou, and Y. Tian. SVFormer: A direct training spiking transformer for efficient video action recognition. In *Human Brain and Artificial Intelligence*, pp. 161–180. Springer, 2025.

H. Zheng, Y. Wu, L. Deng, Y. Hu, and G. 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.

Z. Zhou, Y. Zhu, C. He, Y. Wang, S. Yan, Y. Tian, and L. Yuan. Spikformer: When spiking neural network meets transformer. In *International Conference on Learning Representations*, 2023.

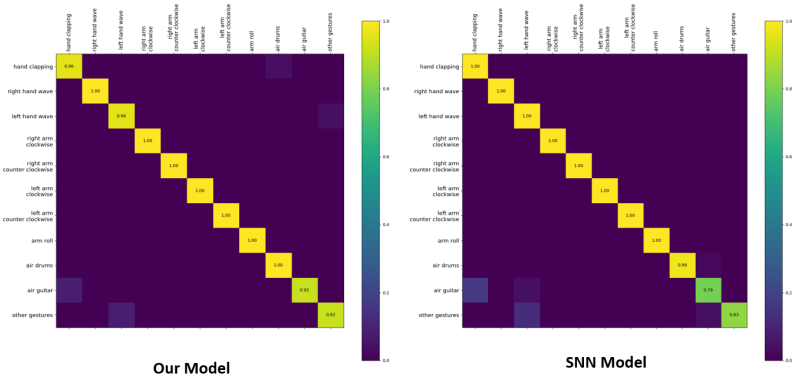

Figure 4: Confusion matrices of CF-TS-SNN and the temporal-only SNN on DVS128Gesture. The two-stream model shows fewer concentrated off-diagonal errors among motion-similar classes.

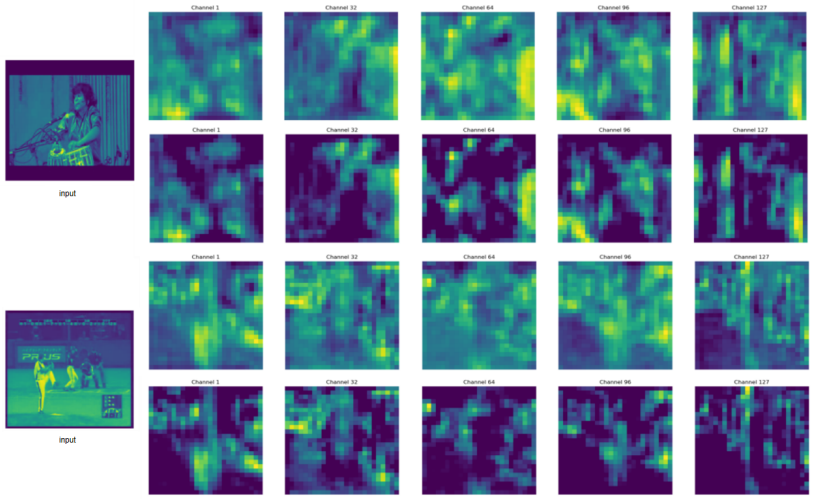

Figure 5: Representative temporal feature maps before and after cross-dimensional weighting. Rows 1 and 3 show unweighted features; rows 2 and 4 show the corresponding weighted features. The weighted responses are more concentrated in regions assigned higher spatial support.

A Additional Method and Qualitative Visualizations

The following visualizations provide supporting method details and qualitative analyses referenced from the main paper.

Qualitative analysis. The confusion matrices in Fig. 4 show fewer concentrated off-diagonal errors for CF-TS-SNN than for the temporal-only SNN. Several motion-similar gesture pairs exhibit clearer separation, consistent with the additional spatial description supplied by the repeated clip summary. Figure 5 visualizes temporal features before and after cross-dimensional weighting. The weighted maps suppress responses assigned low spatial support and retain more concentrated activations, consistent with the multiplicative operation in Eq. 13.