TDAP: TEMPORAL AND DEPENDENCY-AWARE PRUNING FOR SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking Neural Networks (SNNs) provide an energy-efficient alternative to traditional Artificial Neural Networks for event-based vision tasks. However, due to substantial parameter redundancy, their deployment on resource-constrained neuromorphic hardware remains challenging. Existing post-training pruning methods for SNNs primarily treat each channel as an independent feature extractor, overlooking two critical characteristics of spike dynamics: (1) the temporal informativeness of information propagation across time steps, and (2) the predictive dependencies reflecting inter-channel relationships. To address these limitations, we propose TDAP (Temporal Dependency-Aware Pruning), a post-training pruning framework that discovers channel dependencies directly from calibration data without requiring fine-tuning. TDAP comprises two modules: (i) a Temporal Firing Rate (TFR) module that encodes information arrival through learnable temporal weightings, and (ii) a Channel Cross-Correlation Dependency (CCD) module that captures predictive relationships between channels via lagged correlation analysis with spike-specific statistics. We further introduce a geometric mean fusion strategy to aggregate the temporal and dependency modules. On CIFAR-100, TDAP attains 54.23% at 60% sparsity, surpassing pure temporal methods by up to 8.63 percentage points. On event-based DVS-CIFAR10, TDAP reaches 37.00% at 70% sparsity, exceeding Wanda by 13.30 percentage points. These results validate that modeling temporal and channel dependencies is essential for aggressive SNN compression. Anonymous code is available at `https://anonymous.4open.science/r/project-code-2026-0271`.

1 INTRODUCTION

Spiking Neural Networks (SNNs) have emerged as a promising paradigm for energy-efficient neural computation, leveraging event-driven activation and temporal dynamics to process sparse asynchronous data streams (Maass, 1997; Davies et al., 2018). Unlike conventional artificial neural networks, which perform dense matrix multiplications at every forward pass, SNNs propagate information via discrete spike events, offering the potential for substantial reductions in energy consumption when deployed on neuromorphic hardware (Roy et al., 2019; Davies et al., 2021). However, the practical deployment of SNNs on resource-constrained edge devices is hindered by substantial parameter redundancy. Network pruning, the systematic removal of redundant parameters, has proven indispensable for compressing ANNs (Han et al., 2015), yet its adaptation to SNNs presents unique challenges that existing methods fail to address.

Existing post-training pruning methods for ANNs, such as magnitude pruning (Han et al., 2015) and Wanda (Sun et al., 2023), treat channels as independent feature extractors. When applied to SNNs, these methods uniformly aggregate spike activations over all time steps, thereby ignoring that early spikes often encode coarse-grained features while later spikes refine predictions (Kim et al., 2020; Rathi & Roy, 2021). This temporal blindness underestimates the importance of channels whose activity is concentrated at decision-critical moments. Moreover, current criteria evaluate each channel in isolation, failing to account for the predictive dependencies that naturally arise in spiking dynamics. In an SNN, the spike train of one channel can predict the firing of another after a temporal lag, forming a directed influence structure. Pruning a channel with high predictive influence severs the input to its dependent channels, causing disproportionate information loss analogous to an information bottleneck in network flow.

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.

To address these limitations, we propose Temporal Dependency-Aware Pruning (TDAP), a post-training framework that explicitly models both *when* information arrives and *which* channels drive others. TDAP incorporates a Temporal Firing Rate (TFR) module that learns time-step weights from calibration data through a tiny MLP with only 49 parameters per layer, replacing rigid hand-crafted decay schedules with data-driven importance profiles. A concurrent Channel Cross-Correlation Dependency (CCD) module discovers directed predictive dependencies among channels via lagged cross-correlation analysis, yielding a dependency graph in which the out-degree of a channel quantifies its collective influence. To enhance discrimination in the sparse binary regime, we augment the out-degree with channel-level firing density and temporal stability statistics. The two saliency vectors are combined through geometric mean fusion, producing a unified importance score for each weight. For layer l, the importance tensor is

$$\mathbf{I}^{(l)} = |\mathbf{W}^{(l)}| \odot \sqrt{\mathbf{M}_{\text{tfr}}^{(l)} \cdot \mathbf{M}_{\text{ccd}}^{(l)}}, \tag{1}$$

where the channel-wise TFR and CCD vectors are broadcast to the weight shape. We show that preserving channels with high out-degree in the dependency graph minimizes an information-bottleneck upper bound, providing formal motivation for dependency-aware pruning. We further establish a sufficient condition under which geometric mean fusion reduces variance relative to arithmetic fusion. The pruning pipeline uses 128 label-free calibration samples and requires no backbone fine-tuning.

Experiments on CIFAR-10, CIFAR-100, and the neuromorphic DVS-CIFAR10 dataset demonstrate that TDAP substantially outperforms magnitude pruning, Wanda, and recent SNN-specific compressors under aggressive sparsity. On CIFAR-10 with a Spiking ResNet-19 at 70% sparsity, TDAP attains 86.75% accuracy, exceeding magnitude pruning by 1.49 percentage points and the learnable temporal-only variant by 3.25 percentage points. On DVS-CIFAR10 at the same sparsity, TDAP exceeds Wanda by 13.30 percentage points. The framework adds only 0.001% parameter overhead and completes pruning in under 45 seconds on a single GPU.

Our contributions are as follows:

1. We introduce the TFR module, which quantifies channel importance through learnable temporal weighting. A lightweight MLP with 49 parameters per layer estimates time-step weights from calibration data, overcoming the rigidity of hand-crafted decay schedules.
2. We propose the CCD module, which discovers predictive dependencies between channels via lagged cross-correlation analysis. Augmented with firing density and temporal stability, CCD provides a channel-influence measure that respects the directed temporal flow in SNNs.
3. We derive an information-bottleneck upper bound for dependency-aware masking and a variance condition for geometric mean fusion, and validate the resulting criterion across frame-based and event-based benchmarks.

2 Related Work

Post-training pruning for ANNs and SNNs. Post-training pruning compresses pre-trained models without retraining. Magnitude pruning (Han et al., 2015; Wang et al., 2026) removes small weights but ignores the data distribution. Wanda (Sun et al., 2023) improves this criterion by using weight-activation products, whereas SparseGPT (Frantar & Alistarh, 2023) uses second-order updates at greater computational cost. These methods treat networks as static mappings and do not model SNN temporal dynamics. Recent SNN adaptations include Spiking Brain Compression (Shi et al., 2025); however, they still regard channels as independent and do not model temporal heterogeneity. TDAP instead uses lightweight first-order calibration statistics.

Temporal dynamics and channel dependencies. SNNs encode information in spike timing, not only in firing rates (Neftci et al., 2019; Roy et al., 2019). Early spikes may encode coarse features while later spikes refine predictions (Kim et al., 2020), motivating temporal pruning schedules (Chen et al., 2021). Differentiation on spike representations (Meng et al., 2022) and temporal-wise attention (Yao et al., 2021) further demonstrate that *when* information arrives is as important as *what* is transmitted. However, existing post-training pruning criteria do not learn temporal weights from calibration data alone.

Inter-channel relationships in SNNs also reflect functional specialization. SE blocks (Hu et al., 2018) and CBAM (Woo et al., 2018) measure simultaneous correlations in real-valued activations, whereas SNNs require lagged predictive dependencies between binary spike trains. Granger-causality tests (Granger, 1969) are statistically rigorous but computationally expensive for online pruning. Iterative autoregressive surrogates are likewise poorly suited to one-shot pruning. Existing SNN pruning and evolutionary structure-learning methods (Shi et al., 2025; Shen et al., 2023) evaluate channels in isolation or rely on expensive training. CCD uses lagged cross-correlation as an efficient proxy and augments it with firing density and temporal stability.

3 PRELIMINARIES

Spiking neural networks. Consider a feed-forward SNN with L spiking layers. At time step $t \in \{1, \dots, T\}$, layer l receives a spike tensor $\mathbf{S}_t^{(l-1)} \in \{0,1\}^{N \times C_{\text{in}}^{(l)} \times H \times W}$ from the preceding layer (or the encoded input), where N is the batch size. The membrane potential $\mathbf{U}_t^{(l)}$ and output spike $\mathbf{S}_t^{(l)}$ of a Leaky Integrate-and-Fire neuron are

$$\mathbf{U}_t^{(l)} = \tau \mathbf{U}_{t-1}^{(l)} \odot \left(1 - \mathbf{S}_{t-1}^{(l)}\right) + \mathbf{W}^{(l)} * \mathbf{S}_t^{(l-1)}, \tag{2}$$

$$\mathbf{S}_t^{(l)} = \Theta\Big(\mathbf{U}_t^{(l)} - \vartheta\Big), \tag{3}$$

where τ is the membrane time constant, $\mathbf{W}^{(l)} \in \mathbb{R}^{C_{\text{out}}^{(l)} \times C_{\text{in}}^{(l)} \times K \times K}$ is the convolutional weight, $\Theta(\cdot)$ is the Heaviside function, and ϑ is the firing threshold. Surrogate gradients (Neftci et al., 2019) overcome the non-differentiability of Θ during training.

Dynamic connectivity. Post-training pruning methods for ANNs generally treat each channel as an independent feature extractor. In an SNN, however, the spike train of one channel may predict spikes in another after a temporal lag, forming dependencies that reflect functional specialization. Pruning a channel with high out-degree can therefore cause disproportionate information loss because dependent channels lose their driving input. TDAP discovers these dependencies from calibration data and uses them as a pruning signal.

Problem definition. Given a pre-trained SNN with parameters $\{\mathbf{W}^{(l)}\}_{l=1}^{L}$ and calibration set $\mathcal{D}_{\text{cal}}$, we seek binary masks $\{\mathbf{M}^{(l)}\}_{l=1}^{L}$ such that the pruned weights are $\widehat{\mathbf{W}}^{(l)} = \mathbf{W}^{(l)} \odot \mathbf{M}^{(l)}$. No backbone weight is fine-tuned after pruning. Following the form of the Wanda criterion (Sun et al., 2023), each weight is scored by

$$\mathbf{I}^{(l)} = |\mathbf{W}^{(l)}| \odot \mathbf{\Phi}^{(l)}, \tag{4}$$

where $\mathbf{\Phi}^{(l)}$ is a channel-wise metric broadcast over the output and spatial dimensions. At target sparsity s, global thresholding produces

$$M_{c_{\text{out}}, c_{\text{in}}, :, :}^{(l)} = \mathbb{1}\Big[I_{c_{\text{out}}, c_{\text{in}}, :, :}^{(l)} \geq \gamma_s\Big], \quad \gamma_s = \text{Quantile}(\text{vec}(\mathbf{I}), s)\,. \tag{5}$$

We decompose the activation metric into complementary SNN-specific factors,

$$\mathbf{\Phi}^{(l)} = f\Big(\mathbf{M}_{\text{tfr}}^{(l)}, \mathbf{M}_{\text{ccd}}^{(l)}\Big), \tag{6}$$

where $\mathbf{M}_{\text{tfr}}^{(l)}$ captures temporal firing-rate dynamics and $\mathbf{M}_{\text{ccd}}^{(l)}$ captures channel cross-correlation dependencies.

4 METHOD

Figure 1 summarizes TDAP. A temporal module learns when information is informative, while a dependency module estimates which channels predict others. Their normalized scores are fused into a unified saliency criterion and multiplied by the weight magnitude.

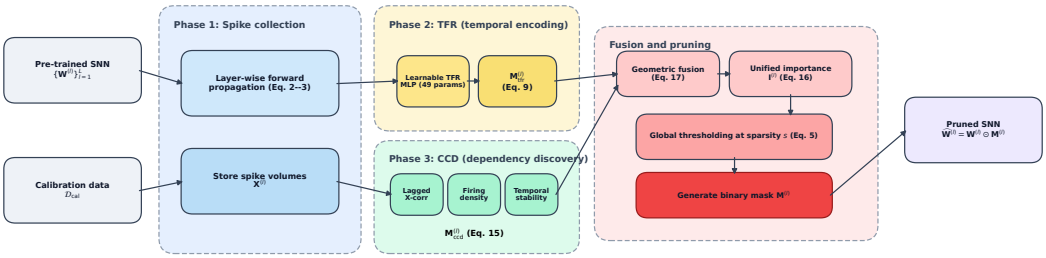


Figure 1: Overview of TDAP. Given a pre-trained SNN, TDAP collects spike volumes from calibration data, computes Temporal Firing Rate (TFR) and Channel Cross-Correlation Dependency (CCD) metrics, fuses them by geometric mean, and generates pruning masks in one shot without fine-tuning.

4.1 Temporal Firing Rate Module

For layer l, let $\mathbf{X}^{(l)} \in \{0,1\}^{T\times N\times C_{\text{in}}^{(l)}\times H\times W}$ denote the input spike volume collected by forwarding the calibration set. We first compute the per-time-step spatial firing rate

$$F_{t,n,c}^{(l)} = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} X_{t,n,c,h,w}^{(l)}, \tag{7}$$

which gives $\mathbf{F}^{(l)} \in [0,1]^{T\times N\times C_{\text{in}}^{(l)}}$.

Learnable temporal weighting. Each layer uses a lightweight MLP that maps the normalized time index $t/T \in [0,1]$ to a scalar logit. A softmax over time produces $\widetilde{w}_t^{(l)} \geq 0$ and $\sum_t \widetilde{w}_t^{(l)} = 1$. The last layer of the MLP is initialized to zero so that $\mathrm{Softmax}(\mathbf{0}) = 1/T$, allowing temporal preferences to emerge from the calibration data rather than from initialization bias. The MLP can optionally be optimized for K iterations using

$$\mathcal{L}_{\text{pre}} = -\lambda_1 \operatorname{Var}\left(\mathbf{M}_{\text{tfr}}^{(l)}\right) + \lambda_2 \mathcal{H}\left(\widetilde{\mathbf{w}}^{(l)}\right), \tag{8}$$

where the first term increases discrimination across channels and $\mathcal{H}(\mathbf{p}) = -\sum_i p_i \log p_i$ discourages extreme concentration. The learnable TFR score is

$$M_{\text{tfr},c}^{(l)} = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T} \widetilde{w}_t^{(l)} F_{t,n,c}^{(l)}. \tag{9}$$

For comparison, a fixed TFR baseline uses an exponential-decay schedule,

$$w_t = \frac{\exp(-t/\tau_{\text{decay}})}{\sum_{t'=1}^{T}\exp(-t'/\tau_{\text{decay}})}, \qquad t = 1,\ldots,T, \tag{10}$$

and $M_{\text{tfr},c}^{(l)} = N^{-1}\sum_{n,t} w_t F_{t,n,c}^{(l)}$. This baseline encodes the prior that early spikes are more important (Kim et al., 2020) but cannot adapt to dataset-specific temporal profiles.

4.2 Channel Cross-Correlation Dependency Module

TFR analyzes time within each channel but treats channels independently. CCD instead captures predictive dependencies: the spike train of channel i at time $t-\delta$ may predict the train of channel j at time t. We use "predictive dependency" to denote temporal predictability, not intervention-based causation (Granger, 1969). A full Granger-causality test would require repeated autoregressive model comparison and is too costly for one-shot pruning.

Lagged cross-correlation. For channel c, extract a temporal trace $\mathbf{z}_c \in \mathbb{R}^{T\times N}$ from $\mathbf{F}^{(l)}$. For lag $\delta \in \{1,\dots,\Delta\}$ (default $\Delta = 1$), the standardized cross-correlation is

$$\rho_{i,j}^{(\delta)} = \frac{\mathrm{Cov}\left(\mathbf{z}_i^{(\delta)}, \mathbf{z}_j^{(0)}\right)}{\sigma(\mathbf{z}_i^{(\delta)})\,\sigma(\mathbf{z}_j^{(0)}) + \epsilon}, \tag{11}$$

where $\epsilon = 10^{-10}$ ensures numerical stability. We average absolute correlations over lags,

$$\mathbf{D}^{(l)} = \frac{1}{\Delta}\sum_{\delta=1}^{\Delta}\left|\boldsymbol{\rho}^{(\delta)}\right|, \qquad D_{i,i}^{(l)} = 1. \tag{12}$$

Thus $D_{i,j}^{(l)}$ measures how strongly channel i predicts channel j after a temporal lag, regardless of whether the correlation is positive or negative.

Firing density and temporal stability. Correlation alone may not distinguish noisy-but-active from structured-but-sparse channels. We therefore compute the mean firing density

$$\rho_c = \frac{1}{TN}\sum_{t=1}^{T}\sum_{n=1}^{N} F_{t,n,c}, \tag{13}$$

and temporal stability, defined as the inverse mean coefficient of variation,

$$s_c = \left(1 + \frac{1}{T}\sum_{t=1}^{T}\frac{\sigma_t(F_{\cdot,c})}{\mu_t(F_{\cdot,c}) + \epsilon}\right)^{-1}. \tag{14}$$

The final CCD vector combines outgoing predictive influence, density, and stability:

$$\mathbf{M}_{\mathrm{ccd}}^{(l)} = \mathbf{D}^{(l)}\mathbf{1} + \alpha\boldsymbol{\rho} + \beta\mathbf{s}. \tag{15}$$

4.3 Unified Importance and Fusion

The vectors $\mathbf{M}_{\mathrm{tfr}}^{(l)}$ and $\mathbf{M}_{\mathrm{ccd}}^{(l)}$ are normalized and broadcast to the weight shape. The unified importance is

$$I_{c_{\mathrm{out}},c_{\mathrm{in}},k_1,k_2}^{(l)} = \left|W_{c_{\mathrm{out}},c_{\mathrm{in}},k_1,k_2}^{(l)}\right| f\left(M_{\mathrm{tfr},c_{\mathrm{in}}}^{(l)}, M_{\mathrm{ccd},c_{\mathrm{in}}}^{(l)}\right). \tag{16}$$

Our primary fusion is the geometric mean,

$$f_{\mathrm{geo}}(a,b) = \sqrt{ab}. \tag{17}$$

It dampens extreme values in either metric while still requiring both temporal and dependency cues to be non-negligible. This is useful when the two statistics have heterogeneous scales across layers. We also evaluate multiplication ab, arithmetic averaging $(a+b)/2$, and weighted fusion $\lambda_a a + \lambda_b b$ with $\lambda_a + \lambda_b = 1$.

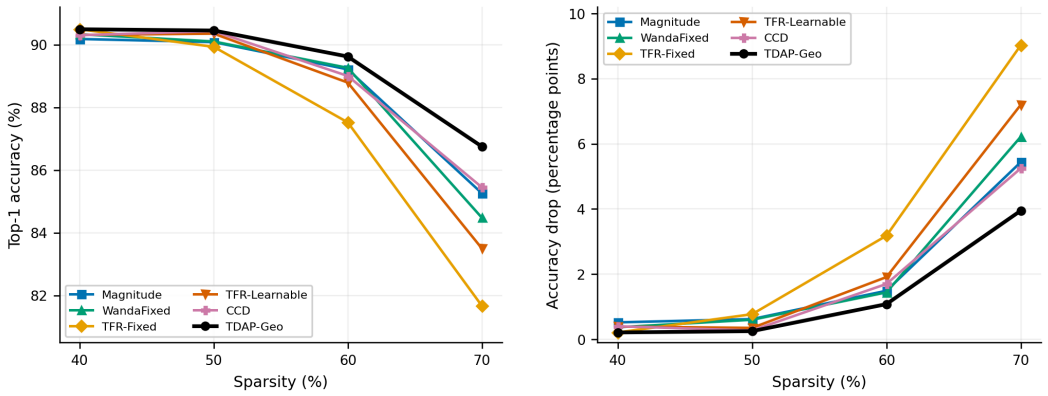
4.4 Theoretical Motivation

We formalize the information bottleneck induced by pruning and motivate retaining channels with large predictive out-degree. Complete statements and derivations appear in Appendix B.

Dependency-aware pruning. Let $d_c = \sum_j D_{c,j}^{(l)}$ be the out-degree of channel c, and let $\mathcal{R}_{\mathrm{ccd}}$ be the set removed by pure out-degree CCD pruning at sparsity s. The information-bottleneck coefficient defined in Appendix B satisfies

$$\mathcal{B}^{(l)}(\mathcal{R}_{\mathrm{ccd}}) \le \frac{s}{1-s}\cdot\frac{1}{C_{\mathrm{in}}^{(l)}}\sum_{c=1}^{C_{\mathrm{in}}^{(l)}} d_c\,\|\mathbf{D}^{(l)}\|_1. \tag{18}$$

The proof follows by showing that removing the lowest-out-degree channels minimizes an upper bound on the predictive influence severed between removed and retained sets.



(a) Top-1 accuracy (%) (b) Accuracy drop (percentage points)

Figure 2: Pruning behavior on CIFAR-10. Geometric mean fusion consistently yields the highest retained accuracy among the reported TDAP variants.

Geometric fusion. Let M_{tfr} and M_{ccd} be unbiased non-negative importance estimators with means $\mu_{\mathrm{tfr}}, \mu_{\mathrm{ccd}}$ and noise variances $\sigma^2_{\mathrm{tfr}}, \sigma^2_{\mathrm{ccd}}$. A Delta-method analysis gives a sufficient condition for geometric fusion to have lower asymptotic mean-squared error than arithmetic fusion:

$$\frac{\sigma^2_{\mathrm{tfr}}}{\mu^2_{\mathrm{tfr}}} + \frac{\sigma^2_{\mathrm{ccd}}}{\mu^2_{\mathrm{ccd}}} < 2\left(\frac{\mu_{\mathrm{tfr}} - \mu_{\mathrm{ccd}}}{\mu_{\mathrm{tfr}} + \mu_{\mathrm{ccd}}}\right)^2 + \frac{2\sigma_{\mathrm{tfr}}\sigma_{\mathrm{ccd}}}{\mu_{\mathrm{tfr}}\mu_{\mathrm{ccd}}}. \tag{19}$$

5 Experiments

We evaluate TDAP on CIFAR-10, CIFAR-100, and DVS-CIFAR10 using Spiking ResNet-19 with LIF neurons. All comparisons use post-training pruning without backbone fine-tuning.

5.1 Experimental Setup

Datasets and models. For CIFAR-10 and CIFAR-100, we use frame-based inputs with $T = 20$. For DVS-CIFAR10, we use raw event streams with $T = 10$ and hidden dimension $H_{\mathrm{hid}} = 16$ in the TFR MLP. The unpruned Spiking ResNet-19 (Fang et al., 2021a;b) achieves 90.70%, 65.76%, and 82.30% top-1 accuracy on CIFAR-10, CIFAR-100, and DVS-CIFAR10, respectively.

Baselines. Magnitude pruning (Han et al., 2015) retains weights solely by absolute value. WandaFixed (Sun et al., 2023) multiplies each weight magnitude by the norm of the corresponding activation and aggregates uniformly across time. TFR-Learnable uses Equation 9; TFR-Fixed uses Equation 10; and CCD isolates Equation 15. These baselines decouple magnitude, activation, temporal weighting, and channel dependency.

Implementation and metrics. We report top-1 classification accuracy under global weight sparsity. The calibration set contains 128 unlabeled samples. The TFR MLP is optionally optimized for $K = 100$ iterations with Equation 8; CCD uses lag $\Delta = 1$. Each pruning mask is generated from one pass over the calibration set, with no backbone fine-tuning.

5.2 Main Results

Figure 2 visualizes CIFAR-10 accuracy and accuracy drop from 40% to 70% sparsity using the values in Table 1. Temporal- and dependency-aware variants reduce degradation relative to magnitude and uniform-activation baselines. Geometric fusion is best or tied at every reported sparsity, indicating that it balances the complementary signals without the collapse of direct multiplication or domination by a single arithmetic term.

Table 1: Top-1 accuracy (%) of Spiking ResNet-19 on CIFAR-10. The unpruned model achieves 90.70%.

Sparsity	Magnitude	WandaFixed	TFR-Fixed	TFR-Learn.	CCD	TDAP-Geo
40%	90.18	90.33	90.49	90.31	90.30	**90.49**
50%	90.08	90.10	89.93	90.35	90.43	**90.45**
60%	89.21	89.26	87.52	88.79	89.00	**89.62**
70%	85.26	84.48	81.67	83.50	85.45	**86.75**

CIFAR-10. At low sparsity, all methods are competitive. Beyond 50%, TDAP's advantage increases: at 70% sparsity, TDAP-Geo obtains 86.75%, exceeding magnitude pruning by 1.49 points and TFR-Learnable by 3.25 points (Table 1).

Table 2: Top-1 accuracy (%) on CIFAR-100. The unpruned model achieves 65.76%.

Sparsity	Magnitude	WandaFixed	TFR-Fixed	TFR-Learn.	CCD	TDAP-Geo
40%	64.06	63.80	63.30	63.91	**64.90**	64.30
50%	**63.57**	60.36	58.66	58.85	63.39	62.41
60%	55.91	47.60	45.36	45.60	**56.53**	54.23
70%	36.71	24.54	22.29	22.64	**37.28**	31.08

CIFAR-100. TDAP-Geo remains competitive at low sparsity and strongly improves over pure temporal criteria as sparsity increases. At 60%, it exceeds TFR-Fixed by 8.87 points and TFR-Learnable by 8.63 points. The isolated CCD score is strongest on this benchmark (Table 2), indicating that channel dependency contributes most of the gain under severe compression.

Table 3: Top-1 accuracy (%) on DVS-CIFAR10. The unpruned model achieves 82.30%.

Sparsity	Magnitude	WandaFixed	TFR-Fixed	TFR-Learn.	CCD	TDAP-Geo
35%	**82.20**	79.70	80.20	79.80	81.60	81.70
45%	80.20	75.60	73.80	74.80	79.40	**80.80**
55%	69.10	63.30	59.50	60.10	**72.70**	72.60
65%	45.50	37.50	37.70	36.20	46.90	**48.80**
70%	30.60	23.70	27.40	24.50	**38.40**	37.00

DVS-CIFAR10. The event-based setting exhibits strong temporal heterogeneity. At 70% sparsity, TDAP-Geo exceeds magnitude pruning by 6.40 points and WandaFixed by 13.30 points. CCD alone is 1.40 points higher than the fused score at this operating point, while TDAP-Geo gives the strongest result at 65% sparsity.

5.3 Ablation Studies

Fusion strategy. Table 4 compares four fusion functions. Geometric fusion is most robust across the reported datasets and sparsity levels. On DVS-CIFAR10 at 60% sparsity, it reaches 62.70%, exceeding multiplication by 18.50 points and arithmetic averaging by 8.60 points.

Table 4: Fusion strategies at 50% and 60% sparsity (top-1 accuracy, %).

	50% sparsity				60% sparsity			
Dataset	Multiply	Average	Geometric	Weighted	Multiply	Average	Geometric	Weighted
CIFAR-10	90.01	90.32	**90.45**	90.08	88.49	89.45	**89.62**	89.41
CIFAR-100	56.67	60.73	**62.41**	59.59	41.74	49.10	**54.23**	47.15
DVS-CIFAR10	68.70	73.50	**77.70**	72.90	44.20	54.10	**62.70**	51.50

Temporal weighting. The learnable TFR increasingly outperforms its fixed counterpart as compression grows (Table 5). At 80% sparsity, the gain is 4.15 points, supporting dataset-specific rather than hand-crafted temporal profiles.

Table 5: TFR variants on CIFAR-10 (top-1 accuracy, %).

Sparsity	TFR-Fixed	TFR-Learnable	Improvement
50%	89.93	90.35	+0.42
60%	87.52	88.79	+1.27
70%	81.67	83.50	+1.83
80%	40.61	44.76	+4.15

CCD components. Table 6 ablates CCD on CIFAR-100. Lagged cross-correlation is the dominant component: CCD-Only-Lag is within 0.32 points of the full model at 70% sparsity, whereas removing it degrades performance substantially.

Table 6: CCD ablation on CIFAR-100 (top-1 accuracy, %). The reported unpruned accuracy for this ablation is 62.52%.

Sparsity	Magnitude	CCD-Full	CCD-w/o-Lag	CCD-Only-Lag
50%	36.91	**45.90**	23.66	45.85
60%	21.35	**33.11**	16.14	26.74
70%	4.95	**8.98**	5.35	8.66

Across datasets, temporal and dependency cues become most valuable beyond 50% sparsity, when pruning increasingly disrupts temporal information flow. The DVS-CIFAR10 results show the largest gain over WandaFixed, reaching 13.30 points at 70% sparsity.

6 Conclusion

We presented TDAP, a post-training pruning framework for SNNs that jointly models temporal heterogeneity through a learnable Temporal Firing Rate module and inter-channel predictive dependencies through a Channel Cross-Correlation Dependency module. Geometric mean fusion balances the two signals, supported by theoretical motivation and empirical results across CIFAR-10/100 and DVS-CIFAR10. TDAP provides an efficient approach to SNN compression for edge deployment without backbone fine-tuning.

AI use statement

Generative AI tools were used to assist with conceptual refinement, manuscript drafting and language editing, consistency checking, and conversion to the ICLR 2027 format. All AI-assisted material was reviewed by the authors against the experimental records and cited sources. The authors take responsibility for the final content of this work, including its text, claims, equations, code, and other artifacts.

Ethics statement

This work uses public image-classification benchmarks and involves no human subjects or personally identifying data. Its intended benefit is lower-cost, energy-efficient neural inference. As with other efficient vision technologies, downstream applications may include privacy-sensitive settings; deployment therefore remains subject to applicable consent, privacy, fairness, and security safeguards.

Reproducibility statement

Equations 7–17 specify the pruning criterion, and Appendix A gives the complete algorithm. Dataset settings, calibration size, time steps, baselines, and sparsity definitions are reported in Section 5;

complete theoretical derivations appear in Appendix B. An anonymous implementation is available at `https://anonymous.4open.science/r/project-code-2026-0271`.

REFERENCES

Yanqi Chen, Zhaofei Yu, Wei Fang, Tiejun Huang, and Yonghong Tian. Pruning of deep spiking neural networks through gradient rewiring. *arXiv preprint arXiv:2105.04916*, 2021.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A. Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R. Risbud. Advancing neuromorphic computing with Loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934, 2021.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21056–21069, 2021a.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2661–2671, 2021b.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.

Clive W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

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

Jinseok Kim, Kyungsu Kim, and Jae-Joon Kim. Unifying activation- and timing-based learning rules for spiking neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19534–19544, 2020.

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

Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan 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.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

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

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

Jiangrong Shen, Qi Xu, Jian K. Liu, Yueming Wang, Gang Pan, and Huajin Tang. ESL-SNNs: An evolutionary structure learning strategy for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 86–93, 2023.

Lianfeng Shi, Ao Li, and Benjamin Ward-Cherrier. Spiking brain compression: Exploring one-shot post-training pruning and quantization for spiking neural networks. *arXiv preprint arXiv:2506.03996*, 2025.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Junqiao Wang, Zhehang Ye, and Yuqi Ouyang. Spiking layer-adaptive magnitude-based pruning. *arXiv preprint arXiv:2603.14946*, 2026.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision*, pp. 3–19, 2018.

Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10221–10230, 2021.

A Pruning Algorithm

Algorithm 1: Post-Training SNN Pruning with TDAP

Input: Pre-trained SNN $\mathcal{M}$ with weights $\{\mathbf{W}^{(l)}\}$; calibration loader $\mathcal{D}_{\text{cal}}$; target sparsity s; fusion function f (default: geometric mean); lag Δ; hidden dimension H_{hid}.

Output: Pruned model $\widehat{\mathcal{M}}$ with masks $\{\mathbf{M}^{(l)}\}$.

// Phase 1: collect input spikes
for each batch $(\mathbf{X}, \cdot) \in \mathcal{D}_{\text{cal}}$ **do**
  Perform layer-wise forward propagation using Equations 2–3.
  Store input spike volumes $\mathbf{X}^{(l)}$ for all layers.
end for
// Phase 2: compute TFR
for $l = 1, \ldots, L$ **do**
  Initialize the layer's TFR MLP with $H_{\text{hid}} = 16$ (Section 4.1).
  Optionally optimize Equation 8; compute $\mathbf{M}_{\text{tfr}}^{(l)}$ using Equation 9.
end for
// Phase 3: compute CCD
for $l = 1, \ldots, L$ **do**
  Compute $\mathbf{D}^{(l)}$ using Equations 11–12.
  Compute density $\boldsymbol{\rho}$ and stability $\mathbf{s}$ using Equations 13–14.
  Assemble $\mathbf{M}_{\text{ccd}}^{(l)}$ using Equation 15.
end for
// Phase 4: fuse and prune
for $l = 1, \ldots, L$ **do**
  Compute $\mathbf{I}^{(l)}$ using Equation 16 with f_{geo}.
  Determine global threshold γ_s using Equation 5.
  Generate $\mathbf{M}^{(l)}$ and set $\widehat{\mathbf{W}}^{(l)} \leftarrow \mathbf{W}^{(l)} \odot \mathbf{M}^{(l)}$.
end for; **return** $\widehat{\mathcal{M}}$.

B Theoretical Analysis

B.1 Information-Bottleneck Bound

Definition B.1 (Information-bottleneck coefficient). Let $\mathcal{R} \subseteq \{1, \ldots, C_{\text{in}}^{(l)}\}$ be the removed channel set with $|\mathcal{R}| = sC_{\text{in}}^{(l)}$. Define

$$\mathcal{B}^{(l)}(\mathcal{R}) = \sum_{j \notin \mathcal{R}} \sum_{i \in \mathcal{R}} |D_{i,j}^{(l)}|. \tag{20}$$

This quantity measures the predictive influence that removed channels exert on retained channels.

Theorem B.2 (Dependency-aware pruning). Let $\mathcal{R}_{\text{mag}}$ and $\mathcal{R}_{\text{ccd}}$ be the channel sets removed by magnitude pruning and pure out-degree CCD pruning, respectively. With $d_c = \sum_j D^{(l)}_{c,j}$, CCD satisfies

$$\mathcal{B}^{(l)}(\mathcal{R}_{\text{ccd}}) \le \frac{s}{1-s} \cdot \frac{1}{C^{(l)}_{\text{in}}} \sum_{c=1}^{C^{(l)}_{\text{in}}} d_c \, \|\mathbf{D}^{(l)}\|_1. \tag{21}$$

Proof. When $\alpha = \beta = 0$, the CCD score is $M^{(l)}_{\text{ccd},c} = d_c = \sum_j D^{(l)}_{c,j}$, so retaining top-scoring channels is equivalent to retaining those with highest out-degree. Magnitude pruning instead ranks by $w_c = \|\mathbf{W}^{(l)}_{:,c}\|_F$, which need not preserve high-out-degree channels. For removal set $\mathcal{R}$ with complement $\mathcal{S}$,

$$\mathcal{B}^{(l)}(\mathcal{R}) = \mathbf{1}_{\mathcal{R}}^{\top} \mathbf{D}^{(l)} \mathbf{1}_{\mathcal{S}} = \sum_{i \in \mathcal{R}} d_i - \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{R}} D^{(l)}_{i,j} \le \sum_{i \in \mathcal{R}} d_i. \tag{22}$$

CCD minimizes the right-hand side by removing the lowest-out-degree channels. Let γ be the $(1-s)$-quantile of $\{d_c\}$. Then

$$\sum_{i \in \mathcal{R}_{\text{ccd}}} d_i \le s C^{(l)}_{\text{in}} \gamma, \qquad \gamma \le \frac{\|\mathbf{d}\|_1}{(1-s) C^{(l)}_{\text{in}}}. \tag{23}$$

Since $\|\mathbf{d}\|_1 = \|\mathbf{D}^{(l)}\|_1$, combining the inequalities yields Equation 21. □

B.2 Learnable TFR Objective

Theorem B.3 (Stationary-point characterization). Let $\mathbf{A} \in \mathbb{R}^{T \times T}$ be the temporal covariance matrix

$$A_{t,t'} = \frac{1}{NC} \sum_{n=1}^{N} \sum_{c=1}^{C} \left(F_{t,n,c} - \overline{F}_t\right) \left(F_{t',n,c} - \overline{F}_{t'}\right), \tag{24}$$

where $\overline{F}_t = (NC)^{-1} \sum_{n,c} F_{t,n,c}$. Then $\mathbf{A}$ is symmetric positive semidefinite. Any interior stationary point of Equation 8, expressed over the simplex $\Delta_T = \{\widetilde{\mathbf{w}} \ge 0 : \mathbf{1}^{\top} \widetilde{\mathbf{w}} = 1\}$, satisfies

$$\widetilde{w}^*_t = \frac{\exp\left(\frac{2\lambda_1}{\lambda_2} (\mathbf{A}\widetilde{\mathbf{w}}^*)_t\right)}{\sum_{t'=1}^{T} \exp\left(\frac{2\lambda_1}{\lambda_2} (\mathbf{A}\widetilde{\mathbf{w}}^*)_{t'}\right)}. \tag{25}$$

Derivation. Let centered data matrix $\mathbf{X} \in \mathbb{R}^{T \times (NC)}$ have columns $\mathbf{x}_{n,c} = \mathbf{F}_{:,n,c} - \overline{\mathbf{F}}$. Then $\mathbf{A} = (NC)^{-1} \mathbf{X}\mathbf{X}^{\top} \succeq 0$. The objective can be written

$$\mathcal{L}_{\text{pre}}(\widetilde{\mathbf{w}}) = -\lambda_1 \widetilde{\mathbf{w}}^{\top} \mathbf{A} \widetilde{\mathbf{w}} + \lambda_2 \sum_t \widetilde{w}_t \log \widetilde{w}_t. \tag{26}$$

For an interior stationary point, the Lagrangian first-order condition is

$$-2\lambda_1 (\mathbf{A}\widetilde{\mathbf{w}})_t + \lambda_2 (\log \widetilde{w}_t + 1) - \mu = 0, \tag{27}$$

which, after normalization, gives Equation 25.

B.3 Lagged Correlation Under a Binary Autoregressive Model

Theorem B.4 (Asymptotic lagged correlation). Suppose channels i and j follow a stationary binary autoregressive process with

$$\Pr(z_j[t] = 1 \mid \mathcal{H}_{t-1}) = \sigma\left(\sum_{\delta=1}^{\Delta} a_\delta z_i[t-\delta] + \sum_{\delta=1}^{\Delta} b_\delta z_j[t-\delta] + c\right), \tag{28}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$. Under the model assumptions and conditioning on all confounders, a nonzero a_δ induces nonzero population association at the corresponding lag; when all $a_\delta = 0$,

the conditional association vanishes. For stationary ergodic sequences, sample lagged correlations converge in probability to their population values as the effective sample size grows.

For binary series, the population lagged correlation is

$$\rho_{i,j}^{(\delta)} = \frac{\mathbb{E}[z_i[t-\delta]z_j[t]] - \mathbb{E}[z_i]\mathbb{E}[z_j]}{\sqrt{\mathrm{Var}(z_i)\,\mathrm{Var}(z_j)}}. \tag{29}$$

The monotonicity of σ gives the sign of the conditional association when $a_\delta \neq 0$, while the no-influence case gives conditional independence under the stated assumptions. Standard ergodic laws yield convergence of empirical moments and hence of the sample correlation.

Corollary B.5 (Density and estimation noise). For mean firing probability p_1 in the sparse regime, the null variance of a sample lagged correlation scales approximately as

$$\mathrm{Var}\left(\widehat{\rho}_{i,j}^{(\delta)} \mid \mathrm{null}\right) \approx \frac{1}{TNp_1}. \tag{30}$$

The density term in Equation 15 therefore up-weights channels whose dependency estimates are supported by more spike events.

B.4 Geometric Mean Fusion

Theorem B.6 (Delta-method comparison). Let $M_{\mathrm{tfr}} = \mu_{\mathrm{tfr}} + \epsilon_{\mathrm{tfr}}$ and $M_{\mathrm{ccd}} = \mu_{\mathrm{ccd}} + \epsilon_{\mathrm{ccd}}$ be non-negative estimators with zero-mean noises and variances σ_{tfr}^2 and σ_{ccd}^2. A second-order expansion gives

$$\sqrt{M_{\mathrm{tfr}}M_{\mathrm{ccd}}} \approx \sqrt{\mu_{\mathrm{tfr}}\mu_{\mathrm{ccd}}}\left(1 + \frac{\epsilon_{\mathrm{tfr}}}{2\mu_{\mathrm{tfr}}} + \frac{\epsilon_{\mathrm{ccd}}}{2\mu_{\mathrm{ccd}}} - \frac{\epsilon_{\mathrm{tfr}}^2}{8\mu_{\mathrm{tfr}}^2} - \frac{\epsilon_{\mathrm{ccd}}^2}{8\mu_{\mathrm{ccd}}^2} + \frac{\epsilon_{\mathrm{tfr}}\epsilon_{\mathrm{ccd}}}{4\mu_{\mathrm{tfr}}\mu_{\mathrm{ccd}}}\right). \tag{31}$$

Neglecting covariance terms, this implies

$$\mathbb{E}[M_{\mathrm{geo}}] \approx \sqrt{\mu_{\mathrm{tfr}}\mu_{\mathrm{ccd}}}\left(1 - \frac{\sigma_{\mathrm{tfr}}^2}{8\mu_{\mathrm{tfr}}^2} - \frac{\sigma_{\mathrm{ccd}}^2}{8\mu_{\mathrm{ccd}}^2}\right), \tag{32}$$

$$\mathrm{Var}(M_{\mathrm{geo}}) \approx \frac{\mu_{\mathrm{ccd}}}{4\mu_{\mathrm{tfr}}}\sigma_{\mathrm{tfr}}^2 + \frac{\mu_{\mathrm{tfr}}}{4\mu_{\mathrm{ccd}}}\sigma_{\mathrm{ccd}}^2. \tag{33}$$

For $M_{\mathrm{avg}} = (M_{\mathrm{tfr}} + M_{\mathrm{ccd}})/2$,

$$\mathbb{E}[M_{\mathrm{avg}}] = \frac{\mu_{\mathrm{tfr}} + \mu_{\mathrm{ccd}}}{2}, \qquad \mathrm{Var}(M_{\mathrm{avg}}) = \frac{\sigma_{\mathrm{tfr}}^2 + \sigma_{\mathrm{ccd}}^2}{4}. \tag{34}$$

Taking $M_{\mathrm{true}} = \sqrt{\mu_{\mathrm{tfr}}\mu_{\mathrm{ccd}}}$ and comparing the second-order MSE approximations yields the sufficient condition in Equation 19.

C Limitations

The TFR module operates at discrete time steps and does not capture exact within-step spike arrival times. CCD computes dependencies within each layer, omitting cross-layer temporal interactions that may arise in deep architectures. The asymptotic argument in Theorem B.4 relies on a stationary binary autoregressive model and adequate control of confounders; real spike trains may violate these assumptions. The empirical evaluation is limited to image classification on three benchmarks with ResNet-style architectures. TDAP's behavior on recurrent SNNs, larger datasets, and other tasks remains to be studied. Finally, geometric fusion treats the temporal and dependency scores symmetrically after normalization; adaptive or input-dependent fusion may yield additional gains.