HUS-SNN: HOMEOSTATIC UNBIASED SPARSITY FOR N:M SPARSE SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks offer high energy efficiency, but their real-world deployment is limited by the mismatch between theoretical sparsity and dense weight implementations. N:M structured sparsity can improve hardware efficiency, but applying it to SNNs leads to performance issues: gradient expectation bias from the non-commutativity of pruning masks and surrogate gradients, and irreversible neuronal silencing caused by reduced synaptic in-degree. We propose the Homeostatic Unbiased Sparse (HUS) framework to address these issues via a two-path optimization design. It combines gradient-aware mask updates with shadow gradient accumulation and multiplicative updates to enable unbiased topological search under N:M constraints. At the same time, a homeostatic weight update path regulates neuron excitability to prevent silent neurons. Experiments on multiple datasets show that our method maintains competitive accuracy while significantly reducing FLOPs, latency time and energy. Codes are available at `https://anonymous.4open.science/r/HUS-SNN-FDE0/`.

1 INTRODUCTION

Spiking neural networks (SNNs) achieve high energy efficiency via intrinsic spatiotemporal sparsity. However, there is a large gap between the theoretical advantages of event-driven computation and practical hardware acceleration, which limits real-world deployment(Wu et al., 2024; Rathi et al., 2023). Fine-grained $N : M$ structured sparsity can improve the efficiency of general-purpose accelerators. Yet applying such fixed constraints to the nonlinear temporal dynamics of spiking neurons hurts model performance(Muralidharan, 2023; Ahmed Zaid et al., 2024). Specifically, forcibly pruning synaptic connections disturbs membrane potential accumulation, causing problems such as gradient expectation bias and permanent neuronal inactivity.

These problems come from the inherent conflict between static structural pruning and the randomness of surrogate gradient learning(Aliyev et al., 2026; Sun et al., 2026). As identified in our theoretical analysis, the non-commutativity between the pruning mask and the surrogate gradient operator triggers a state of gradient starvation where masked parameters are deprived of essential learning signals. In addition, sharply reducing synaptic input connections under $N : M$ sparsity often makes neurons fail to fire, which severely impairs feature extraction ability. To address the two issues above, we need a dynamic method that satisfies hardware structural constraints while adapting to neural plasticity during training.

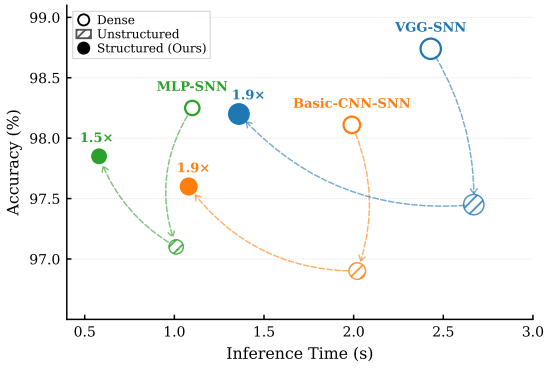

Figure 1: Energy-accuracy trade-off for dense, unstructured, and our structured $N : M$ sparsity. This figure shows 2:4 sparsity. Numbers denote energy reduction multiples relative to the dense baseline. Our method reduces latency time while maintaining high accuracy.

Existing SNN pruning methods mainly use static weight magnitude rules or heuristic probabilistic strategies, ignoring the dynamic characteristics of spike-based computation(Bu et al., 2025). These methods are easily affected by early training weights, which fix the final network structure and prevent the model from learning key features in later training stages. Besides, existing methods have no effective feedback mechanism to reactivate silent neurons during training, leading to irreversible information loss under high sparsity. There is an urgent need for an optimization framework that can concurrently refine sparse topology and maintain functional viability throughout the learning lifecycle.

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 propose the Homeostatic Unbiased Sparse (HUS) framework which treats $N : M$ sparsity as a regulated co-evolutionary process between topological masks and synaptic weights. Our framework operates through the iteration of the binary mask M and the iteration of the weight matrix W. The update of mask M combines gradient-guided dynamic masking and multiplicative updates, enabling unbiased structural search and avoiding gradient starvation. Meanwhile, the update of W adopts slow-scale firing rate regulation to globally adjust synaptic strength and keep all neurons active. This decoupled and cooperative design enables HUS to maintain high model capability while fully satisfying hardware-friendly sparse constraints. As shown in Fig.1, our structured N:M sparsity offers a substantially better energy-accuracy trade-off compared to conventional unstructured pruning, enabling high accuracy with significantly reduced energy consumption. Our contributions are as follows:

1. We identify gradient expectation bias induced by the non-commutativity between pruning masks and surrogate gradients in $N : M$ sparse SNNs, and propose an unbiased mask iteration mechanism with shadow gradient accumulation and multiplicative updates.
2. We reveal neuronal silencing caused by structural in-degree reduction, and introduce a slow-timescale homeostatic weight iteration that maintains neuronal activity under extreme sparsity.
3. We develop HUS-SNN, integrating hardware-friendly $N : M$ sparsity with spiking dynamics. Experiments demonstrate competitive accuracy with substantial reductions in FLOPs, latency, and energy consumption.

2 Related Works

Table 1: Comparison of representative sparse training methods. "Dynamic" denotes topology updates during training, while "Activity regulation" indicates an explicit mechanism for maintaining neuronal activity under sparsification.

Method	Target	Sparsity	Dynamic	Gradient-guided	Spike-aware	Activity Reg.
Grad R Chen et al. (2021)	SNN	Unstruct.	✓	✓	✗	✗
SR-STE Zhou et al. (2021)	ANN	N:M	✓	✓	✗	✗
STEP Lu et al. (2023)	ANN	N:M	✓	✓	✗	✗
Shi et al. Shi et al. (2024)	SNN	Unstruct.	✓	✓	✓	✗
S-STE Hu et al. (2024)	ANN	N:M	✓	✓	✗	✗
QP-SNN Wei et al. (2025)	SNN	Unstruct.	✗	✗	✓	✗
SpikeNM Ye et al. (2025)	SNN	N:M	✓	✓	✓	✗
HUS-SNN (Ours)	**SNN**	**N:M**	✓	✓	✓	✓

Efficiency Optimization While SNN theoretically offer energy efficiency through sparse activations, practical deployment faces challenges. Neuromorphic hardware like Loihi (Davies et al., 2018) and TrueNorth (Akopyan et al., 2015) exploit event-driven computation, but general-purpose accelerators underutilize SNN sparsity due to dense weight matrices. Recent works explore temporal pruning (Kim et al., 2023), membrane potential quantization (Hasssan et al., 2024; Lee et al., 2025; Guo et al., 2022), and early exit mechanisms (Li et al., 2023; KhademSohi et al., 2024). However, weight-level sparsity in SNN (Liu et al., 2024) remains underexplored compared to activation sparsity.

Neuron Pruning Channel pruning (He et al., 2017; Li et al., 2022b) and filter pruning (Meng et al., 2020) provide coarse-grained acceleration but sacrifice flexibility. N:M fine-grained structured sparsity (Zhang et al., 2022; Zhou et al., 2021) bridges this gap, enforcing exactly N non-zeros per M consecutive weights. This pattern enables 2× speedup on NVIDIA Ampere Sparse Tensor Cores while maintaining accuracy. However, N:M sparsity for SNN faces unique challenges, as shown in Section 3. Our comparison is shown in Table 1.

3 Problem Settings

3.1 Preliminaries

Leaky Integrate and Fire Neurons We adopt the discrete time LIF neuron model. The membrane potential v_t^i of neuron i at time step t evolves according to

$$v_t^i = \beta v_{t-1}^i (1 - s_{t-1}^i) + \sum_j W_{ij} s_t^j \tag{1}$$

where $\beta \in (0,1)$ is the decay factor and $s_t^i \in \{0,1\}$ is the binary spike output. The synaptic weight from neuron j to neuron i is denoted by W_{ij}.A spike is emitted when the membrane potential reaches the firing threshold:

$$s_t^i = H(v_t^i - \theta) \tag{2}$$

with H denoting the Heaviside step function. For gradient based training we employ the surrogate gradient method. The derivative is approximated via a sigmoid function

$$\frac{\partial s}{\partial v} \approx \sigma(kv)(1-\sigma(kv)). \tag{3}$$

N:M Structured Sparsity N:M sparsity enforces that within every consecutive block of M weights exactly N weights are nonzero and $N < M$. This pattern is directly supported by modern hardware accelerators. The sparse weight matrix is obtained through element wise multiplication with a binary mask

$$W_{\text{sparse}} = W \odot \mathcal{M}(W) \tag{4}$$

where $\mathcal{M}(W)$ selects the N weights with largest absolute magnitude within each group of M consecutive elements.

3.2 Limitations of Static N:M Sparse SNNs

We analyze the application of static N:M structured sparsity to spiking neural networks. The forward dynamics of a sparse layer compute the input current

$$I_t = (W \odot \mathcal{M}(W))\, s_t \tag{5}$$

where $s_t \in \{0,1\}^n$ is the binary spike vector and $\mathcal{M}(W)$ denotes a static binary mask generated periodically from weight magnitudes. While this formulation ensures hardware efficiency, it induces two fundamental pathologies that arise from the interaction of static pruning with the event driven nature of spiking neurons. Experiments of limitations are shown in Fig. 2.

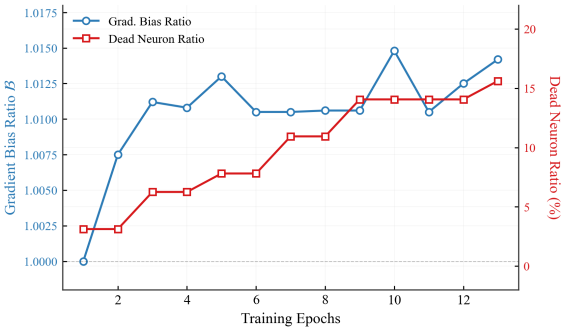

Figure 2: Gradient bias ratio and dead neuron ratio across training epochs.

Gradient Expectation Bias Let $\tilde{\nabla}_W \mathcal{L}$ be the true gradient of the loss and $\nabla_W \mathcal{L}$ the surrogate gradient employed for training. In a dense network, the surrogate gradient provides an approximate optimization signal. Under a static magnitude-based mask $\mathcal{M}(W)$, the operators do not commute and the masked surrogate gradient may deviate from the true gradient projected onto the sparse support.

Theorem 3.1 (Gradient expectation bias)**.** For a static $N{:}M$ sparse spiking neural network with magnitude-based mask $\mathcal{M}(W)$ and surrogate gradient ∇,

$$\mathbb{E}[\mathcal{M} \odot \nabla] \neq \mathcal{M} \odot \nabla_W L_{\text{dense}}(W) \tag{6}$$

holds in general, where L_{dense} is the loss of the dense network. This discrepancy can starve pruned parameters of meaningful learning signals.

Irreversible Neuronal Silencing A spiking neuron requires a sufficient number of active afferent synapses to reach its firing threshold ϑ. Static N:M pruning may reduce the effective in degree of a neuron below a critical value $\rho_c d_i$ where d_i is the original fan in and $\rho_c \in (0,1)$. Once a neuron fails to emit spikes the surrogate gradient for all its incoming weights becomes exactly zero. Because the mask $\mathcal{M}(W)$ is static the neuron remains permanently inactive.

Theorem 3.2 (Irreversible neuronal silencing)**.** For a neuron with threshold θ under a static $N{:}M$ mask $\mathcal{M}(W)$, there exists a critical fraction ρ_c such that if the number of retained afferent synapses satisfies $d_i^{\text{eff}} < \rho_c d_i$,

$$\lim_{t\to\infty} \Pr\big(r_i(t) = 0 \mid d_i^{\text{eff}} < \rho_c d_i\big) = 1, \tag{7}$$

where $r_i(t)$ denotes the firing rate of neuron i at time t. The condition $\nabla_{W_{\cdot,i}} \mathcal{L} = 0$ then follows for all afferent weights, rendering the failure irreversible.

These intrinsic limitations demonstrate that static N:M sparsity is fundamentally incompatible with the dynamical constraints of spiking neural computation and motivate the development of an adaptive homeostatic sparse learning framework.

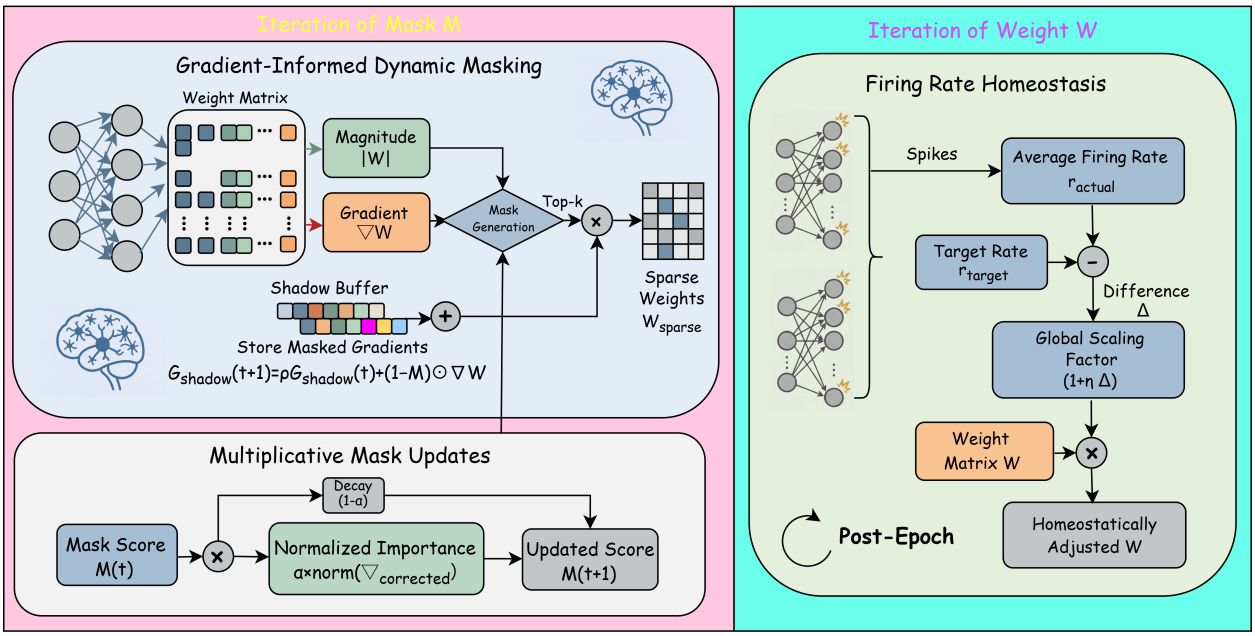

Figure 3: Pipeline of the proposed HUS framework. It includes gradient-guided dynamic masking, multiplicative mask updates, and firing-rate homeostasis to resolve core defects of static N:M sparse SNNs.

4 METHODS

It adopts a dual optimization pipeline consisting of binary mask updates $\mathcal{M}$ and weight updates W. As shown in Fig. 3, the mask path combines gradient-guided dynamic masking, shadow-gradient feedback, and multiplicative mask updates to alleviate gradient starvation. Meanwhile, W is adjusted via firing-rate homeostasis to stabilize neuronal activity under structural sparsity constraints.

4.1 GRADIENT-INFORMED DYNAMIC MASKING

The fundamental limitation of magnitude-based $N:M$ pruning lies in the non-commutativity between the pruning mask $\mathcal{M}$ and the surrogate gradient $\nabla_W \mathcal{L}$ as identified in Theorem 3.1. To bridge the gap between static structural constraints and the temporal evolution of SNNs, we redefine mask generation as an activity-dependent selection process where $\mathcal{M}$ becomes a dynamic function of both weight magnitudes and their instantaneous gradients. We distinguish the topology gradient G_{topo}, which is used to evaluate both active and inactive connections for topology exploration, from the optimizer gradient

$$G_{\text{opt}} = \mathcal{M} \odot G_{\text{topo}}, \tag{8}$$

which updates only active connections. As illustrated in the Iteration of Mask M block of Fig. 3, the selection criterion protects parameters actively participating in the minimization of the loss functional by following the criterion

$$\mathcal{M}(W, G_{\text{topo}}) = \text{TopN-in-M}\left(|W| + \lambda |G_{\text{topo}}|\right) \tag{9}$$

where the hyperparameter λ modulates the trade-off between current synaptic efficacy and latent learning potential, emulating biological activity-dependent pruning. To alleviate gradient starvation, we build a shadow gradient buffer to record historical gradient signals of masked parameters:

$$G_{\text{shadow}}^{(\tau+1)} = \rho G_{\text{shadow}}^{(\tau)} + (1 - \mathcal{M}^{(\tau)}) \odot G_{\text{topo}}^{(\tau)} \tag{10}$$

where the decay factor ρ defines the temporal consolidation window of historical gradients. The corrected gradient is defined as

$$G_{\text{corrected}} = G_{\text{topo}} + G_{\text{shadow}} \tag{11}$$

which preserves historical optimization signals from inactive connections for subsequent topology decisions. This formulation ensures that learning signals from dormant synapses can influence future topology decisions, mitigating the gradient bias inherent in static pruning schemes.

4.2 MULTIPLICATIVE MASK UPDATES

The binary mask update suffers from early training bias, where weights activated at the beginning dominate the mask scoring process. This bias restricts the learning of important late-stage features and causes the network to converge early. To address this issue, we redesign the mask score update to enable scale-robust competition among candidate synaptic connections.

As illustrated in the iteration of mask M within the framework architecture the proposed update uses a mean-normalized momentum rule to accumulate relative topology importance. The score S for each candidate connection evolves according to the relation

$$S^{(t+1)} = (1-\alpha)S^{(t)} + \alpha \frac{|G^{(t)}_{\text{corrected}}|}{\text{mean}(|G^{(t)}_{\text{corrected}}|) + \epsilon}, \tag{12}$$

where α denotes the momentum-based update rate and $\epsilon > 0$ is a small constant for numerical stability. The introduction of the mean-normalized gradient magnitude ensures that the topological importance is evaluated based on relative significance rather than absolute cumulative scale. This normalization makes the mask decision process less sensitive to variations in gradient scale across different training epochs.

Theorem 4.1 (Topology-Signal Preservation)**.** For an inactive connection j that remains masked for K consecutive topology updates,

$$G^{(t+K)}_{\text{shadow},j} = \rho^K G^{(t)}_{\text{shadow},j} + \sum_{k=0}^{K-1} \rho^{K-1-k} G^{(t+k)}_{\text{topo},j}.$$

Hence, any nonzero accumulated topology signal is retained in the shadow state, preventing inactive connections from being excluded from future topology competition solely due to masked optimizer gradients. Moreover, if the normalized corrected-gradient score of an inactive connection remains above the asymptotic N-th largest competing score within the same $N{:}M$ block, then the connection re-enters the active set after finitely many topology updates.

4.3 FIRING RATE HOMEOSTASIS

The imposition of strict $N:M$ sparsity constraints frequently precipitates the irreversible silencing of neurons whose remaining synaptic inputs fall below the firing threshold. This structural pathology creates a computational bottleneck where entire feature channels become dormant, effectively reducing the expressive capacity of the spiking network. To counteract this, we introduce a homeostatic plasticity mechanism that regulates neuronal excitability through the global scaling of synaptic strengths. As illustrated in the iteration of weight W in Fig. 3, this mechanism operates jointly with the weight updates, while topology updates are performed at a slower frequency.

The framework monitors the layer-wise average firing rate r_{actual} through an exponential moving average to capture the steady-state activity of the network

$$r^{(t+1)}_{\text{actual}} = C_1 \cdot r^{(t)}_{\text{actual}} + C_2 \cdot \text{mean}(\text{spikes}^{(t)}) \tag{13}$$

where the coefficients C_1 and C_2 satisfy the convex combination $C_1 + C_2 = 1$. The discrepancy between the observed activity and the target firing rate r_{target} drives the global weight adjustment

$$W \leftarrow W \cdot (1 + \eta_{\text{homeo}} \cdot (r_{\text{target}} - r_{\text{actual}})) \tag{14}$$

where η_{homeo} serves as the homeostatic coupling constant. Unlike additive biases which shift the firing threshold, this multiplicative scaling preserves the relative information encoded in the synaptic weight distribution while ensuring that the total input energy remains sufficient to sustain neuronal activation.

Theorem 4.2 (Survival guarantee for sparse neurons under HUS)**.** Assume that for a neuron i that remains silent after some finite time, the conditional expectations of the gradient correction satisfy

$$\mathbb{E}\big[-\eta_t\,(M'_{t,ij}\,g_{t,ij} + G_{t,ij}) \mid \mathcal{F}_{t-1}\big] \geq 0 \tag{15}$$

Then, every neuron in a HUS-trained SNN with homeostatic weight iteration fires infinitely often almost surely.

The following theorem establishes a generalization bound for HUS-SNN under the $N{:}M$ structured sparsity constraint.

Theorem 4.3. For any $\delta > 0$, with probability at least $1-\delta$ over the training set of size m, every $h \in \mathcal{H}_{\text{HUS}}$ satisfies

$$\mathcal{R}(h) \leq \hat{\mathcal{R}}_m(h) + \frac{2L_\ell\, C(\beta,T)\,(L_\sigma)^L B_W X_{\max}}{\sqrt{m}} \sqrt{\frac{N}{M}} + 3\sqrt{\frac{\log(2/\delta)}{2m}}, \tag{16}$$

where

$$C(\beta, T) = \sum_{k=0}^{T-1} \beta^k \leq \frac{1}{1-\beta}.$$

5 EXPERIMENTS

We validate the HUS framework by examining whether the dual-track optimization alleviates gradient expectation bias and irreversible neuronal silencing in static N:M sparse SNNs while maintaining competitive accuracy under hardware-friendly structured constraints. The evaluation spans six SNN backbones(MLP-SNN (Li et al., 2022a), Basic-CNN-SNN (Zhang et al., 2023), VGG-SNN (Li et al., 2022c), MobileNet-SNN (Cordone et al., 2022), ResNet-SNN (Fang et al., 2021), Spikingformer (Zhou et al., 2026)), four surrogate gradient functions (Pappas et al., 2023), and six datasets(MNIST (LeCun et al., 2010), Fashion-MNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky, 2009), CIFAR-100 (Krizhevsky, 2009), CIFAR-10-DVS (Li et al., 2017), ImageNet (Deng et al., 2009)) covering static images and neuromorphic event streams. We compare against dense baselines and static magnitude-based N:M pruning under identical training budgets. Ablation studies isolate the contribution of different modules.

In this section, we conduct extensive experiments to answer the following research questions (RQs):

- **(RQ1)** How does HUS perform across different datasets, architectures, and surrogate gradients?
- **(RQ2)** How does HUS compare with existing N:M structured sparse training methods?
- **(RQ3)** How does each component contribute to the effectiveness of HUS?
- **(RQ4)** Does HUS provide practical efficiency gains in latency and energy consumption?
- **(RQ5)** How robust is HUS to different hyperparameter configurations?

5.1 MAIN EXPERIMENTS

Table 2: Main experimental results with the default (ATan-like) surrogate gradient. Test accuracy (%) is mean±std or mean. FLOPs for structured N:M=1:4 sparsity are theoretical values. D/U FLOPs represents for FLOPs for dense and unstructured. Struct FLOPs represents for theoretical FLOPs for N:M=1:4 sparsity.

Dataset	Model	D/U FLOPs (G)	Dense (%)	Unstructured (%)	Structured (%)	Struct FLOPs (G)
MNIST	mlp_snn	0.001878	98.25±0.15	98.10±0.03	96.74±0.67	0.000470
	basic_cnn_snn	0.015605	98.11±0.07	97.97±0.02	96.90±0.12	0.003901
	vgg_snn	0.060111	98.74±0.09	98.45±0.22	98.03±0.27	0.015028
	mobilenet_snn	0.002807	96.30±0.28	95.31±0.10	88.19±1.65	0.000702
	resnet_snn	0.075013	92.09±7.45	92.34±6.27	87.27±0.36	0.018753
Fashion-MNIST	mlp_snn	0.001878	84.58±0.13	84.49±0.16	83.62±0.25	0.000470
	basic_cnn_snn	0.015605	82.96±0.10	82.00±0.31	80.16±0.26	0.003901
	vgg_snn	0.060111	84.27±0.03	84.23±0.18	83.03±0.16	0.015028
	mobilenet_snn	0.002807	78.14±0.16	75.88±1.12	69.24±0.13	0.000702
	resnet_snn	0.075013	82.44±0.40	75.55±3.63	73.51±5.64	0.018753
CIFAR-10	basic_cnn_snn	0.63	64.46	69.60	67.63	0.1575
	vgg_snn	0.63	65.43	63.48	64.51	0.1575
	mobilenet_snn	0.03	53.27	52.14	52.27	0.0075
	resnet_snn	2.20	70.71	68.57	68.48	0.55
CIFAR-100	resnet_snn	4.39	60.66	59.47	58.19	1.0975
CIFAR-10-DVS	basic_cnn_snn	0.067	47.38	45.54	45.63	0.017
	vgg_snn	0.63	55.94	48.51	53.93	0.1575
	mobilenet_snn	0.147	48.08	43.36	40.30	0.03675
	spikingformer	0.269	44.32	44.23	41.96	0.06725
	resnet_snn	0.03	52.01	51.75	50.26	0.0075
ImageNet	resnet_snn	18.14	60.23	56.82	60.05	4.54

As shown in Fig. 4,our N:M model matches the unstructured baseline in test accuracy. Table 2 reports test accuracy and theoretical FLOPs across all architectures under the default ATan surrogate gradient with dense, unstructured, and

structured 1:4 sparsity. For layers to which 1:4 N:M sparsity is applied, structured sparsity reduces the corresponding theoretical weight-computation FLOPs by approximately a factor of four. On MNIST the structured VGG-SNN reaches 98.03% accuracy compared to 98.74% for the dense model, a degradation of 0.71 percentage points, with FLOPs decreasing from 0.060 G to 0.015 G. The unstructured variant achieves 98.45%, indicating that the hardware-friendly structured constraint imposes only a small additional cost. The BasicCNN-SNN and MLP-SNN exhibit similar behavior, retaining structured accuracy within 1.2 to 1.5 percentage points of the dense upper bound. MobileNet-SNN drops from 96.30% to 88.19% under structured sparsity, a larger decline we attribute to the limited redundancy of depthwise separable convolutions. The Spikingformer performs weakly even in the dense regime, yet structured sparsity incurs only a 2.36 percentage-point drop, from 44.32% to 41.96%, while cutting FLOPs from 0.269 G to 0.067 G. On Fashion-MNIST the trends are consistent: VGG-SNN decreases from 84.27% to 83.03% and MLP-SNN from 84.58% to 83.62%, both within 1.5 percentage points of the dense reference.

On CIFAR-10 the ResNet-SNN achieves dense accuracy of 70.71% and structured accuracy of 68.48%, a loss of only 2.23 percentage points accompanied by a FLOPs reduction from 2.20 G to 0.55 G. The BasicCNN-SNN and VGG-SNN follow with structured accuracies of 67.63% and 64.51%, respectively. On CIFAR-100 the ResNet-SNN dense accuracy is 60.66% and its structured accuracy 58.19%, a modest gap of 2.47 percentage points while achieving the fourfold FLOPs reduction. Notably, this accuracy–efficiency trade-off remains consistent across both convolutional and residual architectures, suggesting that HUS is not restricted to a specific network topology. These results demonstrate that the dual-track optimization of HUS effectively preserves representational capacity under aggressive structured sparsity across diverse datasets and architectures. We further evaluate a more aggressive 1:8 sparsity setting to examine the behavior of HUS under stronger structural constraints, with the results reported in Appendix A.

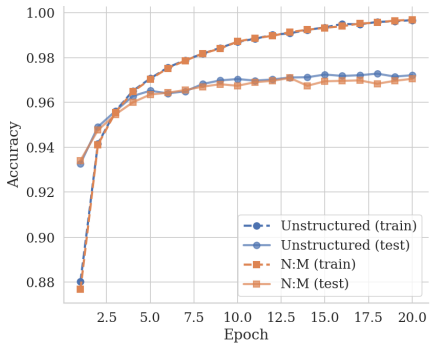

Figure 4: Accuracy curves.

Table 3 evaluates HUS with four surrogate gradient functions across different architectures on MNIST and Fashion-MNIST. HUS maintains stable performance across surrogate choices, with no surrogate causing a substantial performance collapse. Although minor variations exist across architectures, the overall results demonstrate that HUS is not tied to a particular gradient estimator and generalizes well across different surrogate gradients.

5.2 Comparison with Structured Sparse Training Methods

We conduct a comprehensive comparison with three representative $N{:}M$ sparse training methods—SR-STE, STEP, and SpikeNM—across five dataset-architecture settings under a unified 1:4 sparsity constraint. Beyond classification accuracy, we further evaluate neuronal activity to examine whether structured sparsity degrades spiking dynamics.

(a) Classification accuracy (%).

Dataset / Model	Dense	HUS	SR-STE	STEP	SpikeNM
C10 / Basic CNN	84.65	80.10	81.38	78.33	80.47
C10 / VGG	84.59	79.24	81.11	78.38	80.46
C10 / MobileNet	76.62	68.40	70.68	59.40	58.16
C100 / Basic CNN	56.78	53.21	54.15	51.19	49.67
C100 / VGG	56.85	52.53	54.02	50.45	48.99

(b) Neuronal activity.

Method	Dead Neuron	Firing Rate
Dense	1	–
HUS	**1**	**0.1472**
SR-STE	7	0.1032
STEP	7	0.0988
SpikeNM	8	0.1198

Table 4: Comparison with structured sparse training methods under 1:4 sparsity. Neuronal activity is evaluated on CIFAR-10 with Basic CNN (458 neurons). Seed = 42. For a fair comparison, we use a larger matched network configuration. Therefore, the absolute accuracies differ from those in Table 2.

As shown in Table 4a, HUS maintains competitive classification accuracy across all five settings. Although SR-STE achieves the strongest overall classification accuracy, HUS targets a complementary objective: maintaining competitive predictive performance while explicitly mitigating sparsity-induced neuronal inactivity. Table 4b reveals this distinction in internal spiking dynamics. HUS matches the dense network in dead-neuron count while maintaining substantially higher firing activity than the other sparse methods. These results suggest that classification accuracy alone does not fully characterize the effect of structured sparsity on SNN dynamics, motivating the explicit activity regulation introduced by HUS.

Table 3: Accuracy (%) of different surrogate gradients across models, datasets and sparsity modes. FLOPs for N:M=1:4 sparsity are theoretical values. Standard deviations omitted for brevity.

Model	Dataset	Sparse mode	sigmoid	fast_sigmoid	atan	piecewise	FLOPs (G)
mlp_snn	MNIST	dense	97.68	97.76	98.25	97.70	0.001878
		unstructured	97.72	97.76	98.10	97.58	0.001878
		structured	96.15	96.19	96.74	93.34	0.000470
basic_cnn_snn	MNIST	dense	97.82	91.63	98.11	91.63	0.015605
		unstructured	97.23	97.43	97.97	97.43	0.015605
		structured	95.01	95.84	96.90	95.84	0.003901
vgg_snn	MNIST	dense	98.49	95.27	98.74	95.27	0.060111
		unstructured	98.22	91.79	98.45	91.79	0.060111
		structured	97.54	94.43	98.03	94.43	0.015028
mobilenet_snn	MNIST	dense	94.84	95.59	96.30	95.59	0.002807
		unstructured	93.17	92.56	95.31	92.56	0.002807
		structured	71.08	82.06	88.19	82.06	0.000702
resnet_snn	MNIST	dense	87.38	88.14	92.09	88.14	0.075013
		unstructured	77.52	79.62	92.34	79.62	0.075013
		structured	85.36	83.93	87.27	83.93	0.018753
mlp_snn	Fashion-MNIST	dense	84.80	79.85	84.58	84.58	0.001878
		unstructured	84.57	78.50	84.49	84.63	0.001878
		structured	83.32	80.67	83.62	83.70	0.000470
basic_cnn_snn	Fashion-MNIST	dense	80.72	72.33	82.96	81.55	0.015605
		unstructured	80.93	74.38	82.00	81.33	0.015605
		structured	76.52	76.74	80.16	78.49	0.003901
vgg_snn	Fashion-MNIST	dense	84.56	81.72	84.27	84.53	0.060111
		unstructured	84.02	84.17	84.23	84.05	0.060111
		structured	81.66	82.08	83.03	81.65	0.015028
mobilenet_snn	Fashion-MNIST	dense	78.39	76.79	78.14	76.03	0.002807
		unstructured	73.76	73.17	75.88	74.24	0.002807
		structured	64.48	67.23	69.24	69.27	0.000702
resnet_snn	Fashion-MNIST	dense	70.69	80.90	82.44	73.05	0.075013
		unstructured	74.13	76.66	75.55	80.10	0.075013
		structured	58.67	68.92	73.51	64.83	0.018753

5.3 Ablation Study

Fig. 5 compares full HUS with variants removing gradient-informed masking (NoGM), multiplicative mask updates (NoMM), and firing-rate homeostasis (NoFH). Full HUS achieves 97.52% accuracy, compared with 97.26% for static pruning. Removing the multiplicative mask update causes the largest degradation to 96.17%, while removing gradient-informed masking and firing-rate homeostasis reduces accuracy to 97.34% and 97.36%, respectively. These results show that all three components contribute to HUS, with multiplicative mask updates having the largest impact in this setting.

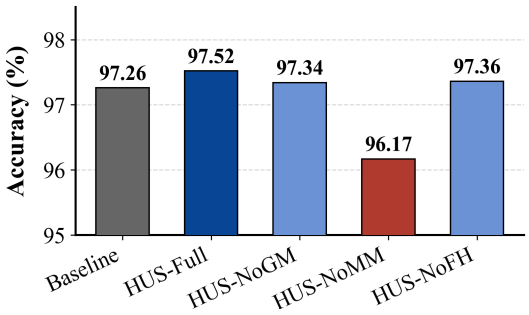

Figure 5: HUS ablation results.

5.4 Energy-Latency Analysis

Based on the results summarized in Table 5, structured N:M sparsity with a $2:4$ pattern demonstrates consistent and marked improvements in both inference latency and physical energy consumption across diverse SNN architectures on MNIST. For instance, in the MLP-SNN, structured sparsity reduces the average inference time from 1.10 s (dense) and 1.01 s (unstructured) to 0.58 s, while lowering the physical energy from 13.07 J and 14.68 J to 8.54 J. Similar trends hold for the convolutional and larger VGG-style models: the structured variant of VGG-SNN achieves 1.36 s and 30.37 J, compared with 2.43 s (57.84 J) for dense and 2.67 s (46.41 J) for unstructured. Even for resource-constrained designs such as MobileNet-SNN, structured sparsity delivers a latency of 0.71 s and energy of 92.97 J, outperforming both the dense (1.35 s, 177.21 J) and unstructured (1.35 s, 168.92 J) configurations. These gains are obtained while preserving

Table 5: Inference time, physical energy, and hardware speedup on MNIST. Mean±std over three runs. Using $2:4$ sparsity. HW Speedup is the end-to-end measured speedup relative to the respective dense baseline.

Model	Sparsity	Infer Time (s)	Physical Energy (J)	HW Speedup
mlp_snn	dense	1.10±0.04	13.07±3.63	1.00±0.00
	unstructured	1.01±0.01	14.68±1.09	1.00±0.00
	structured	0.58±0.00	8.54±1.03	1.81±0.01
basic_cnn_snn	dense	1.99±0.01	30.29±1.15	1.00±0.00
	unstructured	2.02±0.03	29.93±1.53	1.04±0.02
	structured	1.08±0.00	15.72±1.49	1.87±0.01
vgg_snn	dense	2.43±0.32	57.84±16.97	1.00±0.00
	unstructured	2.67±0.05	46.41±4.31	1.07±0.01
	structured	1.36±0.13	30.37±2.27	1.88±0.02
mobilenet_snn	dense	1.35±0.01	177.21±7.59	1.00±0.00
	unstructured	1.35±0.01	168.92±0.88	1.00±0.00
	structured	0.71±0.03	92.97±3.72	1.81±0.01
resnet_snn	dense	2.10±0.35	328.97±39.54	1.00±0.00
	unstructured	1.88±0.06	290.15±14.63	1.00±0.01
	structured	1.14±0.04	177.27±0.83	1.80±0.00

the end-to-end hardware speedup metrics reported in the table, underscoring the practical viability of structured sparsity. Overall, the results validate that $N:M$ structured sparsity is a highly effective design choice for spiking neural networks, offering substantial reductions in computational cost and energy footprint without sacrificing the benefits of sparsity. Measurement details are in Appendix. I.

5.5 Robustness to Hyperparameters

Table 6: Sensitivity of HUS to λ and α on MNIST with MLP-SNN under $2{:}4$ structured sparsity (fixed 50% sparsity). Accuracies are reported in %.

λ	α	Best	Final	Energy	λ	α	Best	Final	Energy
0.00	0.05	95.82	95.78	0.1285	0.50	0.20	96.27	96.27	0.1285
0.00	0.10	96.13	96.08	0.1378	0.50	0.50	96.51	96.51	0.1230
0.00	0.20	**96.57**	96.31	0.1301	0.75	0.05	96.35	96.35	0.1334
0.00	0.50	95.45	95.35	0.1350	0.75	0.10	96.50	96.26	0.1327
0.25	0.05	96.30	96.15	0.1317	0.75	0.20	96.29	96.29	0.1311
0.25	0.10	96.38	96.15	0.1313	0.75	0.50	96.06	96.06	0.1387
0.25	0.20	96.32	96.18	0.1315	1.00	0.05	96.51	96.51	0.1313
0.25	0.50	96.25	96.23	0.1293	1.00	0.10	96.47	96.36	0.1312
0.50	0.05	95.95	95.95	0.1302	1.00	0.20	96.43	96.43	0.1232
0.50	0.10	96.07	95.90	0.1352	1.00	0.50	96.13	96.13	0.1284

Table 6 shows that HUS is relatively insensitive to λ and α. Across all 20 configurations, the best accuracy ranges from 95.45% to 96.57% and the final accuracy from 95.35% to 96.51%, corresponding to variations of only 1.12 and 1.16 percentage points, respectively. The energy proxy also remains within a narrow range of 0.1230–0.1387. Overall, HUS maintains stable performance over a broad range of hyperparameter choices without requiring extensive tuning.

6 Conclusion

We propose the Homeostatic Unbiased Sparse (HUS) framework to tackle gradient expectation bias and irreversible neuronal silencing in N:M sparse spiking neural networks. Its mask and homeostatic weight iterations preserve topology signals and maintain neuronal activity under structural pruning. Evaluated on six architectures and six datasets, HUS preserves accuracy while reducing FLOPs, latency time and energy. Ablation studies validate its dedicated mechanisms, offering a principled solution for energy-efficient SNN deployment on N:M sparsity-enabled hardware.

AI USE DISCLOSURE

Generative AI tools were used to assist with the retrieval of knowledge. They were also used to support manuscript editing, including improving readability, organizing content, and formatting tables and figures.

ETHICS STATEMENT

This work studies structured sparsity for improving the computational and energy efficiency of spiking neural networks. The experiments use standard publicly available benchmark datasets and do not involve human subjects, personally identifiable information, or sensitive data. We are not aware of specific ethical risks introduced by the proposed method beyond those generally associated with the deployment of neural network models. All authors have reviewed and adhered to the ICLR Code of Ethics and take responsibility for the integrity and presentation of the reported results.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the HUS framework, including its gradient-informed masking, multiplicative mask updates, and firing-rate homeostasis mechanisms. Experimental settings, datasets, network architectures, sparsity configurations, surrogate gradients, and implementation details are reported in the main text and Appendix. Complete proofs and assumptions for the theoretical results are also provided in the Appendix. Additional training trajectories and robustness analyses are included to characterize optimization behavior beyond final accuracy. We will release the implementation and configuration files necessary to reproduce the reported experiments. Codes are available at `https://anonymous.4open.science/r/HUS-SNN-FDE0/`.

REFERENCES

Dehbia Ahmed Zaid, Badis Djamaa, and Mohamed Akrem Benatia. Learning N: M structured sparse neural networks from scratch: a comparative study. In *International Conference on Computing Systems and Applications*, pp. 57–69. Springer, 2024.

Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015.

Ilkin Aliyev, Jesus Lopez, and Tosiron Adegbija. Surrogates, Spikes, and Sparsity: Performance Analysis and Characterization of SNN Hyperparameters on Hardware. *arXiv preprint arXiv:2603.24891*, 2026.

Tong Bu, Xinyu Shi, and Zhaofei Yu. Activity Pruning for Efficient Spiking Neural Networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Biswadeep Chakraborty, Beomseok Kang, Harshit Kumar, and Saibal Mukhopadhyay. Sparse spiking neural network: Exploiting heterogeneity in timescales for pruning recurrent SNN. *arXiv preprint arXiv:2403.03409*, 2024.

Kaiwei Che, Luziwei Leng, Kaixuan Zhang, Jianguo Zhang, Qinghu Meng, Jie Cheng, Qinghai Guo, and Jianxing Liao. Differentiable hierarchical and surrogate gradient search for spiking neural networks. *Advances in Neural Information Processing Systems*, 35:24975–24990, 2022.

Ruizhi Chen, Hong Ma, Shaolin Xie, Peng Guo, Pin Li, and Donglin Wang. Fast and efficient deep sparse multi-strength spiking neural networks with dynamic pruning. In *2018 international joint conference on neural networks (ijcnn)*, pp. 1–8. IEEE, 2018.

Xi Chen, Chang Gao, Zuowen Wang, Longbiao Cheng, Sheng Zhou, Shih-Chii Liu, and Tobi Delbruck. Exploiting symmetric temporally sparse BPTT for efficient RNN training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11399–11406, 2024.

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.

Hao Cheng, Jiahang Cao, Erjia Xiao, Mengshu Sun, and Renjing Xu. Gaining the Sparse Rewards by Exploring Lottery Tickets in Spiking Neural Networks. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 442–449. IEEE, 2024.

Loïc Cordone, Benoît Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2022.

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.

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

Lei Deng, Yujie Wu, Yifan Hu, Ling Liang, Guoqi Li, Xing Hu, Yufei Ding, Peng Li, and Yuan Xie. Comprehensive snn compression using admm optimization and activity regularization. *IEEE transactions on neural networks and learning systems*, 34(6):2791–2805, 2021.

Chaoming Fang, Fengshi Tian, Jie Yang, and Mohamad Sawan. Accelerating BPTT-Based SNN Training with Sparsity-Aware and Pipelined Architecture. In *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. IEEE, 2024.

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

Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 326–335, 2022.

Jianhui Han, Zhaolin Li, Weimin Zheng, and Youhui Zhang. Hardware implementation of spiking neural networks on FPGA. *Tsinghua Science and Technology*, 25(4):479–486, 2020.

Zecheng Hao, Tong Bu, Jianhao Ding, Tiejun Huang, and Zhaofei Yu. Reducing ann-snn conversion error through residual membrane potential. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11–21, 2023.

Ahmed Hasssan, Jian Meng, Anupreetham Anupreetham, and Jae-sun Seo. SpQuant-SNN: ultra-low precision membrane potential with sparse activations unlock the potential of on-device spiking neural networks applications. *Frontiers in Neuroscience*, 18:1440000, 2024.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1389–1397, 2017.

Yuezhou Hu, Jun Zhu, and Jianfei Chen. S-ste: Continuous pruning function for efficient 2: 4 sparse pre-training. *Advances in Neural Information Processing Systems*, 37:33756–33778, 2024.

Hossein KhademSohi, Mohammadamin Abedi, Yani Ioannou, Steve Drew, Pooyan Jamshidi, and Hadi Hemmati. Selfxit: An unsupervised early exit mechanism for deep neural networks. *Transactions on machine learning research*, 2024.

Youngeun Kim, Yuhang Li, Hyoungseob Park, Yeshwanth Venkatesha, Ruokai Yin, and Priyadarshini Panda. Exploring lottery ticket hypothesis in spiking neural networks. In *European Conference on Computer Vision*, pp. 102–120. Springer, 2022.

Youngeun Kim, Yuhang Li, Hyoungseob Park, Yeshwanth Venkatesha, Anna Hambitzer, and Priyadarshini Panda. Exploring temporal information dynamics in spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 8308–8316, 2023.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Donghyun Lee, Abhishek Moitra, Youngeun Kim, Ruokai Yin, and Priyadarshini Panda. MD-SNN: Membrane Potential-aware Distillation on Quantized Spiking Neural Network. *arXiv preprint arXiv:2512.04443*, 2025.

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Frontiers in Neuroscience*, Volume 11 - 2017, 2017. ISSN 1662-453X. doi: 10.3389/fnins.2017.00309. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2017.00309.

Wenshuo Li, Hanting Chen, Jianyuan Guo, Ziyang Zhang, and Yunhe Wang. Brain-inspired multilayer perceptron with spiking neurons. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 783–793, 2022a.

Yawei Li, Kamil Adamczewski, Wen Li, Shuhang Gu, Radu Timofte, and Luc Van Gool. Revisiting random channel pruning for neural network compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 191–201, 2022b.

Yuhang Li, Shikuang Deng, Xin Dong, and Shi Gu. Converting artificial neural networks to spiking neural networks via parameter calibration. *arXiv preprint arXiv:2205.10121*, 2022c.

Yuhang Li, Tamar Geller, Youngeun Kim, and Priyadarshini Panda. Seenn: Towards temporal spiking early exit neural networks. *Advances in Neural Information Processing Systems*, 36:63327–63342, 2023.

Shuang Lian, Jiangrong Shen, Qianhui Liu, Ziming Wang, Rui Yan, and Huajin Tang. Learnable Surrogate Gradient for Direct Training Spiking Neural Networks. In *Ijcai*, pp. 3002–3010, 2023.

Timothy P Lillicrap and Adam Santoro. Backpropagation through time and the brain. *Current opinion in neurobiology*, 55:82–89, 2019.

Kefei Liu, Xiaoxin Cui, Yi Zhong, Yisong Kuang, Yuan Wang, Huajin Tang, and Ru Huang. A hardware implementation of SNN-based spatio-temporal memory model. *Frontiers in Neuroscience*, 13:835, 2019.

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

Yucheng Lu, Shivani Agrawal, Suvinay Subramanian, Oleg Rybakov, Christopher De Sa, and Amir Yazdanbakhsh. STEP: learning N: M structured sparsity masks from scratch with precondition. In *International Conference on Machine Learning*, pp. 22812–22824. PMLR, 2023.

Fanxu Meng, Hao Cheng, Ke Li, Huixiang Luo, Xiaowei Guo, Guangming Lu, and Xing Sun. Pruning filter in filter. *Advances in Neural Information Processing Systems*, 33:17629–17640, 2020.

Saurav Muralidharan. Uniform sparsity in deep neural networks. *Proceedings of Machine Learning and Systems*, 5: 743–753, 2023.

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

Christos Pappas, Stefanos Kovaios, Miltiadis Moralis-Pegios, Apostolos Tsakyridis, George Giamougiannis, Manos Kirtas, Joris Van Kerrebrouck, Gertjan Coudyzer, Xin Yin, Nikolaos Passalis, et al. Programmable tanh-, elu-, sigmoid-, and sin-based nonlinear activation functions for neuromorphic photonics. *IEEE Journal of Selected Topics in Quantum Electronics*, 29(6: Photonic Signal Processing):1–10, 2023.

Nitin Rathi, Indranil Chakraborty, Adarsh Kosta, Abhronil Sengupta, Aayush Ankit, Priyadarshini Panda, and Kaushik Roy. Exploring neuromorphic computing based on spiking neural networks: Algorithms to hardware. *ACM Computing Surveys*, 55(12):1–49, 2023.

Clemens JS Schaefer, Pooria Taheri, Mark Horeni, and Siddharth Joshi. The hardware impact of quantization and pruning for weights in spiking neural networks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 70(5): 1789–1793, 2023.

Xinyu Shi, Jianhao Ding, Zecheng Hao, and Zhaofei Yu. Towards energy efficient spiking neural networks: An unstructured pruning framework. In *The Twelfth International Conference on Learning Representations*, 2024.

Qi Sun, Zhen Cao, Kaige Geng, Ziyi Zhang, and Biao Hou. Optimization Method for Surrogate Function in Spiking Neural Networks Based on Membrane Potential Distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25718–25726, 2026.

Bingsen Wang, Jian Cao, Jue Chen, Shuo Feng, and Yuan Wang. A New ANN-SNN Conversion Method with High Accuracy, Low Latency and Good Robustness. In *IJCAI*, pp. 3067–3075, 2023.

Yuchen Wang, Malu Zhang, Yi Chen, and Hong Qu. Signed Neuron with Memory: Towards Simple, Accurate and High-Efficient ANN-SNN Conversion. In *IJCAI*, pp. 2501–2508, 2022.

Wenjie Wei, Malu Zhang, Zijian Zhou, Ammar Belatreche, Yimeng Shan, Yu Liang, Honglin Cao, Jieyuan Zhang, and Yang Yang. Qp-snn: Quantized and pruned spiking neural networks. In *International Conference on Learning Representations*, volume 2025, pp. 96470–96495, 2025.

Jiadong Wu, Yinan Wang, Zhiwei Li, Lun Lu, and Qingjiang Li. A Review of Computing with Spiking Neural Networks. *Computers, Materials & Continua*, 78(3), 2024.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Man Yao, Yuhong Chou, Guangshe Zhao, Xiawu Zheng, Yonghong Tian, Bo Xu, and Guoqi Li. Probabilistic modeling: Proving the lottery ticket hypothesis in spiking neural network. *arXiv preprint arXiv:2305.12148*, 2023.

Shuhan Ye, Yi Yu, Qixin Zhang, Chenqi Kong, Qiangqiang Wu, Xudong Jiang, and Dacheng Tao. Sparse by Rule: Probability-Based N: M Pruning for Spiking Neural Networks. *arXiv preprint arXiv:2511.12097*, 2025.

Yufei Zhang, Hui Xu, Lixing Huang, and Changlin Chen. A storage-efficient SNN–CNN hybrid network with RRAM-implemented weights for traffic signs recognition. *Engineering Applications of Artificial Intelligence*, 123:106232, 2023.

Yuxin Zhang, Mingbao Lin, Zhihang Lin, Yiting Luo, Ke Li, Fei Chao, Yongjian Wu, and Rongrong Ji. Learning best combination for efficient n: M sparsity. *Advances in Neural Information Processing Systems*, 35:941–953, 2022.

Aojun Zhou, Yukun Ma, Junnan Zhu, Jianbo Liu, Zhijie Zhang, Kun Yuan, Wenxiu Sun, and Hongsheng Li. Learning n: m fine-grained structured sparse neural networks from scratch. *arXiv preprint arXiv:2102.04010*, 2021.

Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Zhengyu Ma, Han Zhang, Huihui Zhou, and Yonghong Tian. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954*, 2023.

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

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

A Supplementary Results on MNIST with 1:8 Structured Sparsity

As shown in Table 7, the one to eight structured sparsity removes 87.5% of weights yet HUS preserves accuracy with negligible degradation. VGG-SNN achieves $97.56 \pm 0.31\%$ merely 0.54% below its dense baseline while Basic-CNN-SNN and MLP-SNN show gaps of 0.90% and 0.66% respectively. This resilience stems from two non trivial synergies. Gradient aware dynamic masking prevents the permanent elimination of weights that transiently have small gradients but later become critical. Firing rate homeostasis counteracts the tendency of structured sparsity to silence entire neuron groups. The hardware friendly $N : M$ pattern itself introduces no extra accuracy loss compared to unstructured sparsity indicating that the regular mask structure does not interfere with the homeostatic dynamics.

For ResNet-SNN the $1 : 8$ sparsity yields $95.46 \pm 0.93\%$ close to the dense 96.97% while achieving the full $8\times$ FLOPs reduction. The low variance of 0.93% confirms robust convergence under extreme pruning. The insight is that deep residual SNNs face a unique risk of neuron silence due to gated identity connections. Homeostatic weight update directly mitigates this by adaptively elevating the activity of silent neurons. Multiplicative mask update preserves long range global pathways that residual blocks rely on. Consequently HUS enables aggressive hardware friendly sparsity without re entrant tuning making it practical for next generation accelerators.

Table 7: Main experimental results with the default (ATan-like) surrogate gradient on MNIST dataset. Test accuracy (%) is mean±std or mean. FLOPs for structured N:M=1:8 sparsity are theoretical values. D/U FLOPs represents for FLOPs for dense and unstructured. Struct FLOPs represents for theoretical FLOPs for N:M=1:8 sparsity.

Model	D/U FLOPs (G)	Dense (%)	Unstructured (%)	Structured (%)	Struct FLOPs (G)
mlp_snn	0.000939	97.15±0.13	96.69±0.59	96.49±0.16	0.000117
basic_cnn_snn	0.007802	96.95±0.20	96.29±0.31	96.05±0.40	0.000975
vgg_snn	0.030055	98.10±0.19	97.87±0.06	97.56±0.31	0.003757
mobilenet_snn	0.001403	92.65±0.85	91.67±0.64	91.93±0.80	0.000175
resnet_snn	0.037507	96.97±0.52	96.16±1.04	95.46±0.93	0.004688

B Algorithm

Algorithm 1 HUS-SNN System Update

Require: $\alpha \in (0, 1]$, $\eta > 0$, $\gamma = M/N$, clipping bound B_ξ, Iteration T.
Ensure: Final weights $W^{(T)}$.

Initialize $W^{(0)}$ randomly, $S_{\text{hus}}^{(0)} = \mathbf{0} \in \mathbb{R}^{d\times d}$, $\xi^{(0)} = \mathbf{0} \in \mathbb{R}^{d\times d}$.
Compute initial mask $M'^{(1)} = \mathcal{T}_{N:M}(S_{\text{hus}}^{(0)})$.
for $t = 1$ to T **do**
  Compute true gradient $\tilde{\nabla}^{(t)}$ w.r.t. $W^{(t-1)}$.
  Corrected gradient: $\nabla_{\text{corr}}^{(t)} = \tilde{\nabla}^{(t)} + \xi^{(t-1)}$.
  for each group G_k **do**
    $\mu_G(|\nabla_{\text{corr}}^{(t)}|)_{i\in G_k} = \frac{1}{M}\sum_{j\in G_k} |\nabla_{\text{corr},j}^{(t)}|$.
    if $\mu_G = 0$ **then**
      set $\mu_G = 1$.
    end if
  end for
  $S_{\text{hus}}^{(t)} = (1-\alpha)S_{\text{hus}}^{(t-1)} + \alpha \cdot \frac{|\nabla_{\text{corr}}^{(t)}|}{\mu_G(|\nabla_{\text{corr}}^{(t)}|)}$.
  $M'^{(t+1)} = \mathcal{T}_{N:M}(S_{\text{hus}}^{(t)})$.
  $\xi^{(t)} = \text{clip}\left(\xi^{(t-1)} + (\mathbf{1} - M'^{(t)}) \odot \nabla_{\text{corr}}^{(t)}, -B_\xi, B_\xi\right)$.
  $W^{(t)} = W^{(t-1)} - \eta \cdot M'^{(t)} \odot \gamma \odot \nabla_{\text{corr}}^{(t)}$.
end for
return $W^{(T)}$.

C For Supplementary Details on Related Works

SNN Training Methods Spiking Neural Networks (SNN) communicate through discrete spike events, offering inherent spatiotemporal sparsity that enables significant energy efficiency on neuromorphic hardware (Liu et al., 2019; Han et al., 2020). Training approaches include ANN-SNN conversion (Hao et al., 2023; Wang et al., 2022; 2023) which achieves high accuracy but requires long inference timesteps, and direct training via surrogate gradients (Neftci et al., 2019; Che et al., 2022; Lian et al., 2023) or BPTT variants (Lillicrap & Santoro, 2019; Chen et al., 2024; Fang et al., 2024). Recent architectures like Spikformer (Zhou et al., 2022), Spikingformer (Zhou et al., 2023), and SEW-ResNet (Fang et al., 2021) achieve competitive ImageNet accuracy.

Unstructured Sparsity Magnitude-based pruning (Chakraborty et al., 2024; Schaefer et al., 2023) achieves high compression ratios but requires specialized hardware for acceleration. Lottery ticket hypothesis (Kim et al., 2022; Yao et al., 2023) and dynamic sparse training (Chen et al., 2018) demonstrate that sparse networks can match dense performance. For SNN specifically, Chen et al. (Chen et al., 2021) and Shi et al. (Shi et al., 2024) apply unstructured pruning but face irregular memory access patterns limiting practical speedup.

Sparsity in SNN Limited prior work addresses SNN weight sparsity. ADMM-based methods (Deng et al., 2021) and lottery tickets for SNN (Kim et al., 2022; Cheng et al., 2024) focus on unstructured pruning. Ye et al. (Ye et al., 2025) probability-based N:M pruning for SNN. To our knowledge, no prior work systematically investigates hardware-friendly N:M structured sparsity for SNN or addresses the interplay between temporal dynamics and structured weight pruning.

D Notations and Assumptions

Let $L : \mathbb{R}^d \to \mathbb{R}$ be the loss function. For a stochastic gradient $g(w, \xi)$ we denote by ξ the random noise. The mask $\mathcal{M}_t \in \{0,1\}^d$ follows an N:M pattern, i.e., it is partitioned into groups of size M and each group contains exactly N non-zero entries. The set of all admissible masks is denoted $\mathcal{S}$. The training uses two step sizes $\eta_t, \zeta_t > 0$ satisfying

$$\sum_t \eta_t = \sum_t \zeta_t = \infty, \qquad \sum_t (\eta_t^2 + \zeta_t^2) < \infty, \qquad \lim_{t\to\infty} \eta_t/\zeta_t = 0. \tag{17}$$

The fast variable $G_t \in \mathbb{R}^d$ evolves according to

$$G_{\text{shadow}}^{(\tau+1)} = \rho G_{\text{shadow}}^{(\tau)} + (1 - \mathcal{M}^{(\tau)}) \odot \nabla W^{(\tau)}, \tag{18}$$

and the slow variable w_t follows

$$w_{t+1} = w_t - \eta_t\big(\mathcal{M}_t \odot g(w_t, \xi_t) + G_t\big). \tag{19}$$

For the spiking part, each neuron i obeys the discrete LIF dynamics

$$v_{t+1}^i = \beta v_t^i + \sum_j W_{t,ij}\mathcal{M}_{t,ij}s_t^j - \theta s_t^i, \tag{20}$$

with spike $s_t^i = \mathbf{1}_{\{v_t^i \geq \theta\}}$, leakage $\beta \in (0,1)$, threshold $\theta > 0$, and homeostatic weight update

$$W_{t+1,ij} = W_{t,ij}\big(1 + \eta_{\text{homeo}}(r_{\text{target}} - r_{t,i})\big). \tag{21}$$

The surrogate activation used for gradient computation is σ, which is L_σ-Lipschitz.

We adopt the following assumptions throughout.

Assumption D.1 (Stability of iterates)**.** The sequences $\{w_t\}$ and $\{G_t\}$ remain in a compact set with probability one: $\sup_t \|w_t\| < \infty$ and $\sup_t \|G_t\| < \infty$ almost surely.

Assumption D.2 (Unbiased gradient with bounded variance)**.** $\mathbb{E}_\xi[g(w,\xi)] = \nabla L(w)$ and $\mathbb{E}_\xi[\|g(w,\xi) - \nabla L(w)\|^2] \leq \sigma^2$ for all w.

Assumption D.3 (Instrumental mask condition)**.** Let $\mathcal{F}_{t-1}$ be the σ-field generated by all randomness up to time $t-1$. The mask $\mathcal{M}_t$ is a measurable function of a stale gradient statistic $h_{t-1} \in \mathcal{F}_{t-1}$ such that

$$\mathbb{E}[\mathcal{M}_t \odot g(w_t, \xi_t) \mid \mathcal{F}_{t-1}] = \bar{\mathcal{M}}_t \odot \nabla L(w_t), \tag{22}$$

with $\bar{\mathcal{M}}_t$ bounded away from zero in expectation (relevance), and

$$\mathbb{E}[\mathcal{M}_t \odot g(w_t, \xi_t) - \bar{\mathcal{M}}_t \odot \nabla L(w_t) \mid \mathcal{F}_{t-1}, \mathcal{M}_t] = 0 \tag{23}$$

(exclusion). The second condition says that conditional on the mask and the past, the noise component of the masked stochastic gradient has zero mean.

E GLOBAL CONVERGENCE AND TRAJECTORY EQUIVALENCE

Lemma E.1 (Gradient Flow Convergence)**.** The coupled recursions satisfy $\lim_{t\to\infty} \|\nabla L(w_t)\| = 0$ with probability one. Moreover, the interpolated trajectory of $\{w_t\}$ converges weakly to the solution of the dense gradient flow $\dot{w} = -\nabla L(w)$.

Proof. The system is a two-time-scale stochastic approximation because $\eta_t/\zeta_t \to 0$. By Assumption D.1, the iterates are bounded with probability one, so the ODE method applies. For a frozen w, consider the fast ODE

$$\dot{G} = \mathbb{E}_{\xi,\mathcal{M}}[(I - \mathcal{M}) \odot g(w, \xi) \mid \mathcal{F}_t] - G. \tag{24}$$

Using the exclusion property of Assumption D.3, the conditional expectation simplifies to $(I - \bar{\mathcal{M}}_w) \odot \nabla L(w)$, where $\bar{\mathcal{M}}_w = \mathbb{E}[\mathcal{M} \mid w]$. Hence the fast ODE has a unique globally asymptotically stable equilibrium

$$G^*(w) = (I - \bar{\mathcal{M}}_w) \odot \nabla L(w). \tag{25}$$

Standard two-time-scale results imply that the tracking error $e_t^G := G_t - G^*(w_t)$ satisfies $e_t^G \to 0$ almost surely.

Define the effective slow-scale direction

$$F(w_t, G_t) = \mathbb{E}[\mathcal{M}_t \odot g(w_t, \xi_t) + G_t \mid \mathcal{F}_t]. \tag{26}$$

By relevance, $\mathbb{E}[\mathcal{M}_t \odot g(w_t, \xi_t) \mid \mathcal{F}_t] = \bar{\mathcal{M}}_t \odot \nabla L(w_t)$. Using $G_t = G^*(w_t) + e_t^G$ gives

$$F(w_t, G_t) = \bar{\mathcal{M}}_t \odot \nabla L(w_t) + (I - \bar{\mathcal{M}}_t) \odot \nabla L(w_t) + e_t^G = \nabla L(w_t) + e_t^G. \tag{27}$$

Thus the shadow variable G_t exactly cancels the mask's distortion, up to the vanishing tracking error.

Now consider the Lyapunov function $V(w) = L(w)$. From L-smoothness,

$$\mathbb{E}[L(w_{t+1}) \mid \mathcal{F}_t] \le L(w_t) - \eta_t \langle \nabla L(w_t), F(w_t, G_t) \rangle + \frac{L\eta_t^2}{2} \mathbb{E}[\|\hat{g}_t\|^2 \mid \mathcal{F}_t], \tag{28}$$

where $\hat{g}_t = \mathcal{M}_t \odot g(w_t, \xi_t) + G_t$. Substituting $F = \nabla L(w_t) + e_t^G$ and using boundedness yields

$$\mathbb{E}[L(w_{t+1}) \mid \mathcal{F}_t] \le L(w_t) - \eta_t \|\nabla L(w_t)\|^2 + \eta_t \|\nabla L(w_t)\| \, \|e_t^G\| + \mathcal{O}(\eta_t^2). \tag{29}$$

Since $e_t^G \to 0$, for any $\varepsilon > 0$ eventually $\|e_t^G\| \le \varepsilon$. The Robbins–Siegmund lemma then gives

$$\sum_t \eta_t \|\nabla L(w_t)\|^2 < \infty \tag{30}$$

with probability one. Because $\sum \eta_t = \infty$, we have $\liminf_{t\to\infty} \|\nabla L(w_t)\| = 0$ almost surely. Stability and the ODE method imply that every limit point is stationary, hence $\|\nabla L(w_t)\| \to 0$ with probability one.

For trajectory equivalence, rewrite the slow update as

$$w_{t+1} = w_t - \eta_t(\nabla L(w_t) + e_t^G + \delta_t) \tag{31}$$

with δ_t a martingale difference. Because $e_t^G \to 0$ and $\sum \eta_t^2 < \infty$, standard results on stochastic approximation give weak convergence to the ODE $\dot{w} = -\nabla L(w)$. □

Corollary E.2 (Finite-Time Tracking Bound)**.** Assume in addition that $\eta_t = \eta_0 t^{-\alpha}$, $\zeta_t = \zeta_0 t^{-\beta}$ with $0 < \alpha < \beta \le 1$, $\alpha + \beta > 1$, and that $w \mapsto \bar{\mathcal{M}}_w$, $w \mapsto \nabla L(w)$ are Lipschitz, and the noise has finite fourth moment. Then there exist constants $C_1, C_2 > 0$ and T_0 such that for all $t \ge T_0$,

$$\mathbb{E}[\|e_t^G\|^2] \le C_1 \zeta_t^2 + C_2 \frac{\eta_t^2}{\zeta_t}. \tag{32}$$

Consequently, for the piecewise-constant interpolation $w_\eta(t) = w_{\lfloor t/\eta_t \rfloor}$ and the dense gradient flow solution $w^{\text{dense}}(t)$ starting from the same initial condition, for any fixed $T > 0$,

$$\sup_{s\in[0,T]} \|w_\eta(s) - w^{\text{dense}}(s)\| = \mathcal{O}\big(\eta^{\min(\alpha,\beta-\alpha)/2}\big). \tag{33}$$

Proof. From the fast-scale update,

$$e_{t+1}^G = (1 - \zeta_t) e_t^G + \zeta_t\big((I - \mathcal{M}_t) \odot g_t - G^*(w_t)\big) + \zeta_t\big(G^*(w_t) - G^*(w_{t+1})\big). \tag{34}$$

The second term is a martingale difference with variance $\mathcal{O}(\zeta_t^2)$ by the exclusion property. The third term is bounded by $L_G\|w_{t+1} - w_t\| = \mathcal{O}(\zeta_t \eta_t)$. Taking expectations and solving the recursion gives the stated bound. Plugging this into the slow-scale Euler discretization error yields the convergence rate. □

F GRADIENT EXPECTATION BIAS

Theorem F.1 (Gradient expectation bias)**.** For a static $N{:}M$ sparse spiking neural network with magnitude-based mask $\mathcal{M}(W)$ and surrogate gradient ∇,

$$\mathbb{E}[\mathcal{M} \odot \nabla] \neq \mathcal{M} \odot \nabla_W L_{\text{dense}}(W) \tag{35}$$

holds in general, where L_{dense} is the loss of the dense network. This discrepancy can starve pruned parameters of meaningful learning signals.

Proof. The sparse network is obtained by replacing W with $W \odot \mathcal{M}$, so its loss is $L_{\text{sparse}}(W) = L_{\text{dense}}(W \odot \mathcal{M})$. The stochastic surrogate gradient used for training is $\mathcal{M} \odot g(W \odot \mathcal{M}, \xi)$, where g satisfies $\mathbb{E}_\xi[g(u, \xi)] = \nabla L_{\text{dense}}(u)$ by Assumption D.2. Taking expectation over the noise yields

$$\mathbb{E}_\xi[\mathcal{M} \odot g(W \odot \mathcal{M}, \xi)] = \mathcal{M} \odot \nabla L_{\text{dense}}(W \odot \mathcal{M}). \tag{36}$$

The dense reference vector at the same weight W is $\mathcal{M} \odot \nabla L_{\text{dense}}(W)$. Because $\mathcal{M}$ is not the identity and the gradient ∇L_{dense} is not constant along rays, the two vectors differ in general. The pruned connections receive identically zero gradient signal, while the unpruned ones follow a direction that deviates from the dense counterpart. This constitutes the gradient expectation bias. □

Theorem F.2 (Topology-Signal Preservation)**.** For an inactive connection j that remains masked for K consecutive topology updates,

$$G^{(t+K)}_{\text{shadow},j} = \rho^K G^{(t)}_{\text{shadow},j} + \sum_{k=0}^{K-1} \rho^{K-1-k} G^{(t+k)}_{\text{topo},j}.$$

Hence, any nonzero accumulated topology signal is retained in the shadow state, preventing inactive connections from being excluded from future topology competition solely due to masked optimizer gradients.

Proof. The HUS mask M'_t is computed from scores updated via the multiplicative rule

$$S_{t+1} = (1-\alpha) S_t + \alpha \frac{|\nabla_{\text{corr},t}|}{\text{mean}(|\nabla_{\text{corr},t}|) + \epsilon}, \tag{37}$$

and M'_t selects the top N scores in each M-sized block. The shadow gradient evolves as

$$G^{(t+1)}_{\text{shadow}} = \rho G^{(t)}_{\text{shadow}} + (I - M'_t) \odot G^{(t)}_{\text{topo}}. \tag{38}$$

Consider a connection j that remains inactive for K consecutive topology updates, i.e., $M'_{t+k,j} = 0$ for $k = 0, \ldots, K-1$. For this connection, the shadow update reduces to

$$G^{(t+k+1)}_{\text{shadow},j} = \rho G^{(t+k)}_{\text{shadow},j} + G^{(t+k)}_{\text{topo},j}. \tag{39}$$

Recursively expanding the above relation gives

$$G^{(t+K)}_{\text{shadow},j} = \rho^K G^{(t)}_{\text{shadow},j} + \sum_{k=0}^{K-1} \rho^{K-1-k} G^{(t+k)}_{\text{topo},j}. \tag{40}$$

Thus, topology signals generated while connection j is inactive are retained as an exponentially weighted history rather than being immediately discarded.

In contrast, the optimizer gradient satisfies

$$G^{(t)}_{\text{opt},j} = M'_{t,j} G^{(t)}_{\text{topo},j} = 0 \tag{41}$$

whenever $M'_{t,j} = 0$. Therefore, using the masked optimizer gradient alone would remove the instantaneous topology information associated with an inactive connection.

In HUS, the retained shadow state contributes to ∇_{corr} and hence to the score S_t used for subsequent top-N selection. Consequently, a nonzero accumulated topology signal can continue to affect the score of an inactive connection, preventing it from being excluded from future topology competition solely because its optimizer gradient is masked. This establishes topology-signal preservation and re-entry eligibility, without requiring the masked weight update itself to be an unbiased estimator of the dense gradient. □

G NEURONAL SILENCING

Theorem G.1 (Irreversible neuronal silencing)**.** For a neuron with threshold θ under a static $N{:}M$ mask $\mathcal{M}(W)$, there exists a critical fraction ρ_c such that if the number of retained afferent synapses satisfies $d_i^{\text{eff}} < \rho_c d_i$,

$$\lim_{t\to\infty} \Pr\big(r_i(t) = 0 \mid d_i^{\text{eff}} < \rho_c d_i\big) = 1, \tag{42}$$

where $r_i(t)$ denotes the firing rate of neuron i at time t. The condition $\nabla_{W_{\cdot,i}} \mathcal{L} = 0$ then follows for all afferent weights, rendering the failure irreversible.

Proof. Let $\mathcal{M}$ be static and fix $d_i^{\text{eff}} = \sum_j \mathcal{M}_{ij}$. Assumption 8 gives $|W_{ij}| \le W_{\max}$ uniformly, so $|I_t^i| \le W_{\max} d_i^{\text{eff}}$. During a silent phase $v_t^i < \theta$, the membrane evolves as $v_{t+1}^i = \beta v_t^i + I_t^i$, implying

$$|v_t^i| \le \beta^t |v_0^i| + \frac{W_{\max} d_i^{\text{eff}}}{1-\beta}. \tag{43}$$

Thus $\limsup_{t\to\infty} |v_t^i| \le W_{\max} d_i^{\text{eff}}/(1-\beta)$. If

$$\frac{W_{\max} d_i^{\text{eff}}}{1-\beta} < \theta, \tag{44}$$

the membrane potential can never reach θ. Define $\rho_c = (1-\beta)\theta/(W_{\max} d_i)$; then $d_i^{\text{eff}} < \rho_c d_i$ causes this inequality. Even if the neuron initially spikes, after each reset the same bound holds, so spikes must eventually cease. Once silent, the surrogate gradient for all afferent weights vanishes, and the static mask prevents any weight updates. Hence the silence is permanent with probability 1. □

Theorem G.2 (Survival guarantee for sparse neurons under HUS)**.** Assume that for a neuron i that remains silent after some finite time, the conditional expectations of the gradient correction satisfy

$$\mathbb{E}\big[-\eta_t \,(M'_{t,ij}\, g_{t,ij} + G_{t,ij}) \mid \mathcal{F}_{t-1}\big] \ge 0 \tag{45}$$

Then, every neuron in a HUS-trained SNN with homeostatic weight iteration fires infinitely often almost surely.

Proof. Fix a neuron i and define the permanent-silence event

$$\Omega_\infty = \big\{\exists\, T : s_t^i = 0 \text{ for all } t \ge T\big\}. \tag{46}$$

We prove $\Pr(\Omega_\infty) = 0$. Suppose $\Pr(\Omega_\infty) > 0$. On Ω_∞ the firing-rate estimate $r_{t,i}$ decreases to zero, hence for all sufficiently large t,

$$\gamma_t = 1 + \eta_{\text{homeo}}(r_{\text{target}} - r_{t,i}) \ge 1 + \delta, \qquad \delta = \tfrac{1}{2}\eta_{\text{homeo}} r_{\text{target}} > 0. \tag{47}$$

The weight update for an afferent synapse j reads

$$W_{t+1,ij} = \gamma_t W_{t,ij} - \eta_t\big(M'_{t,ij} g_{t,ij} + G_{t,ij}\big). \tag{48}$$

Taking conditional expectation given $\mathcal{F}_{t-1}$ and using the fact that $W_{t,ij}$ and γ_t are $\mathcal{F}_{t-1}$-measurable, together with equation 47 and equation 45, gives

$$\mathbb{E}[W_{t+1,ij} \mid \mathcal{F}_{t-1}] \ge (1+\delta) W_{t,ij} \tag{49}$$

for all sufficiently large t on Ω_∞.

Summing over all afferent synapses and defining

$$S_t := \sum_j W_{t,ij}, \tag{50}$$

we obtain

$$\mathbb{E}[S_{t+1} \mid \mathcal{F}_{t-1}] \ge (1+\delta) S_t. \tag{51}$$

Thus, permanent silence induces a persistent positive multiplicative drift in the total afferent weight mass.

Since the weights are non-negative and initialized non-degenerately, the afferent weight mass is positive. Under Assumption D.1, however, the weight iterates remain bounded, with

$$0 \le S_t \le d_i W_{\max}. \tag{52}$$

A persistent multiplicative drift with factor strictly larger than one is incompatible with such bounded dynamics. Hence permanent silence cannot persist under the stated assumptions, and therefore

$$\Pr(\Omega_\infty) = 0. \tag{53}$$

Since neuron i was arbitrary, every neuron fires infinitely often with probability one. □

H GENERALIZATION BOUND OF HUS-SNN

Let $\mathcal{H}_{\mathrm{HUS}}$ denote the hypothesis class of HUS-trained SNNs with L layers, T time steps, $N{:}M$ structured sparsity, and surrogate activation σ that is L_σ-Lipschitz. Assume in addition the weight constraints $\|W_l\|_F \le B_W$, bounded inputs $\|x_t\|_2 \le X_{\max}$, and the Lipschitz loss condition.

Theorem H.1. For any $\delta > 0$, with probability at least $1-\delta$ over the training set of size m, every $h \in \mathcal{H}_{\mathrm{HUS}}$ satisfies

$$\mathcal{R}(h) \le \hat{\mathcal{R}}_m(h) + 2L_\ell C(\beta,T)(L_\sigma)^L \left[\frac{B_W X_{\max}}{\sqrt{m}} \sqrt{\frac{N}{M}} + \sqrt{\frac{2\log|\mathcal{S}|}{m}} \right] + 3\sqrt{\frac{\log(2/\delta)}{2m}}, \tag{54}$$

where

$$C(\beta,T) = \frac{1}{T}\sum_{t=1}^{T}\sum_{\tau=1}^{t} \beta^{t-\tau} \le \frac{1}{1-\beta}, \tag{55}$$

and $\mathcal{S}$ denotes the set of admissible $N{:}M$ support patterns.

Proof. We first bound the Rademacher complexity of the weight class. Let

$$\mathcal{W}_{N:M} = \left\{ W \in \mathbb{R}^{d_1 \times d_2} : \|W\|_F \le B_W,\ \mathrm{supp}(W) \in \mathcal{S} \right\}. \tag{56}$$

For a fixed $N{:}M$ support pattern, the active fraction of weights is N/M. The Rademacher complexity of the corresponding restricted weight class is bounded by

$$\frac{B_W X_{\max}}{\sqrt{m}} \sqrt{\frac{N}{M}}. \tag{57}$$

Taking the union over the admissible support patterns gives

$$\hat{\mathfrak{R}}_m(\mathcal{W}_{N:M}) \le \frac{B_W X_{\max}}{\sqrt{m}} \sqrt{\frac{N}{M}} + \sqrt{\frac{2\log|\mathcal{S}|}{m}}. \tag{58}$$

Next we analyze the SNN as a Lipschitz mapping. Unfolding the LIF recurrence, the membrane potential at layer l and time t satisfies

$$v_{l,t} = \sum_{\tau=1}^{t} \beta^{t-\tau} W_l \sigma(v_{l-1,\tau}), \tag{59}$$

where indices not relevant to the argument are suppressed for clarity. Using the Lipschitz property of σ and the bounded weight norms, the temporal accumulation contributes the factor

$$C(\beta,T) = \frac{1}{T}\sum_{t=1}^{T}\sum_{\tau=1}^{t} \beta^{t-\tau} \le \frac{1}{1-\beta}. \tag{60}$$

Applying the contraction argument through the L surrogate-activation layers gives

$$\hat{\mathfrak{R}}_m(\mathcal{H}_{\mathrm{HUS}}) \le C(\beta,T)(L_\sigma)^L \left[\frac{B_W X_{\max}}{\sqrt{m}} \sqrt{\frac{N}{M}} + \sqrt{\frac{2\log|\mathcal{S}|}{m}} \right]. \tag{61}$$

For the L_ℓ-Lipschitz loss ℓ, Talagrand's contraction lemma gives

$$\hat{\mathfrak{R}}_m(\ell \circ \mathcal{H}_{\mathrm{HUS}}) \le L_\ell \hat{\mathfrak{R}}_m(\mathcal{H}_{\mathrm{HUS}}). \tag{62}$$

The standard Rademacher generalization bound states that, with probability at least $1-\delta$,

$$\mathcal{R}(h) \le \hat{\mathcal{R}}_m(h) + 2\hat{\mathfrak{R}}_m(\ell \circ \mathcal{H}_{\mathrm{HUS}}) + 3\sqrt{\frac{\log(2/\delta)}{2m}}. \tag{63}$$

Substituting the complexity bound above yields the stated result. □

I HARDWARE AND MEASUREMENT DETAILS

All experiments were conducted on a dedicated GPU server. The ImageNet experiments were performed on a single NVIDIA GeForce RTX 5090 GPU with 32 GB of GDDR7 memory, a base clock of 2.4 GHz, a boost clock of 2.8 GHz, and a thermal design power of 575 W. All other experiments covering MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100, and CIFAR-10-DVS were carried out on a secondary NVIDIA vGPU with 48 GB of memory, a maximum power of 350 W, and a core clock of 1.59 GHz. This vGPU is virtualized from a physical NVIDIA Tesla T4 or equivalent architecture. Both GPUs were deployed on the same cloud instance with shared network bandwidth to ensure consistent data transfer conditions. Physical energy consumption was acquired via the NVIDIA Management Library. At the beginning of each inference pass, the total GPU energy consumption in millijoules was read using `nvmlDeviceGetTotalEnergyConsumption`. After the inference completed and a GPU synchronization point, the final energy reading was taken. The energy consumed during inference was computed as the difference between the two readings and converted to joules. For every reported energy value, three independent runs were performed with different random seeds, and the mean is presented.

Latency measurements used the standard Python `time.perf_counter` with GPU synchronization via `torch.cuda.synchronize` before and after the inference loop to ensure accurate timing. To isolate core computational speedup, we designed a dedicated microbenchmark that measures the execution time of each linear or convolutional layer in isolation. For a layer with weight matrix W and binary mask $\mathcal{M}$, the dense forward pass computes $y = Wx$ while the sparse pass computes $y = (W \odot \mathcal{M})x$. Each operation is repeated 200 times per layer, with 20 warmup iterations to eliminate initialization overhead. The hardware speedup for a model is defined as the ratio of total dense layer execution time to total sparse layer execution time

$$\text{Speedup} = \frac{\sum_l t_l^{\text{dense}}}{\sum_l t_l^{\text{sparse}}}, \tag{64}$$

where the summation runs over all linear and convolutional layers in the network. For dense and unstructured sparse models which do not employ structured masks, the denominator uses the same dense execution time by construction, yielding a speedup of 1.0. This approach strictly isolates the computational benefit of structured sparsity from memory hierarchy effects and is consistent across all model architectures and sparsity patterns.

J TRAINING DYNAMICS UNDER STRUCTURED SPARSITY

To complement the final accuracy comparison in Section 5.2, we further examine the complete training trajectories of HUS and representative structured sparse training methods. We report test accuracy over 400 epochs on CIFAR-10 and CIFAR-100 with Basic CNN, VGG, and MobileNet architectures under the same $1{:}4$ sparsity setting.

This analysis provides a more detailed view of optimization behavior than the best or final accuracy alone. In particular, we examine whether sparse training reaches a stable solution, whether high peak accuracy is preserved throughout training, and how the behavior changes across architectures and datasets.

J.1 COMPARISON WITH DENSE TRAINING

Figure 6 compares HUS with the corresponding dense models across all six dataset-architecture configurations. As expected, dense training generally provides a higher accuracy ceiling because no weights are removed. Nevertheless, HUS follows a similar overall optimization trajectory and converges to a stable sparse solution across all evaluated configurations.

The gap depends on architecture. For Basic CNN and VGG, HUS approaches the dense trajectory after the early training stage and maintains a relatively stable plateau. The gap is larger for MobileNet, indicating that the more compact architecture is more sensitive to the aggressive $1{:}4$ sparsity constraint. This behavior is observed on both CIFAR-10 and CIFAR-100.

The HUS trajectories also exhibit occasional sharp one-epoch drops in test accuracy. Importantly, these drops are transient: accuracy recovers rapidly and the long-term trajectory remains stable. This behavior is consistent with dynamic topology updates temporarily perturbing the active sparse subnetwork while allowing subsequent optimization to recover. Thus, despite retaining only one active weight in every group of four, HUS consistently reaches a stable solution across datasets and architectures.

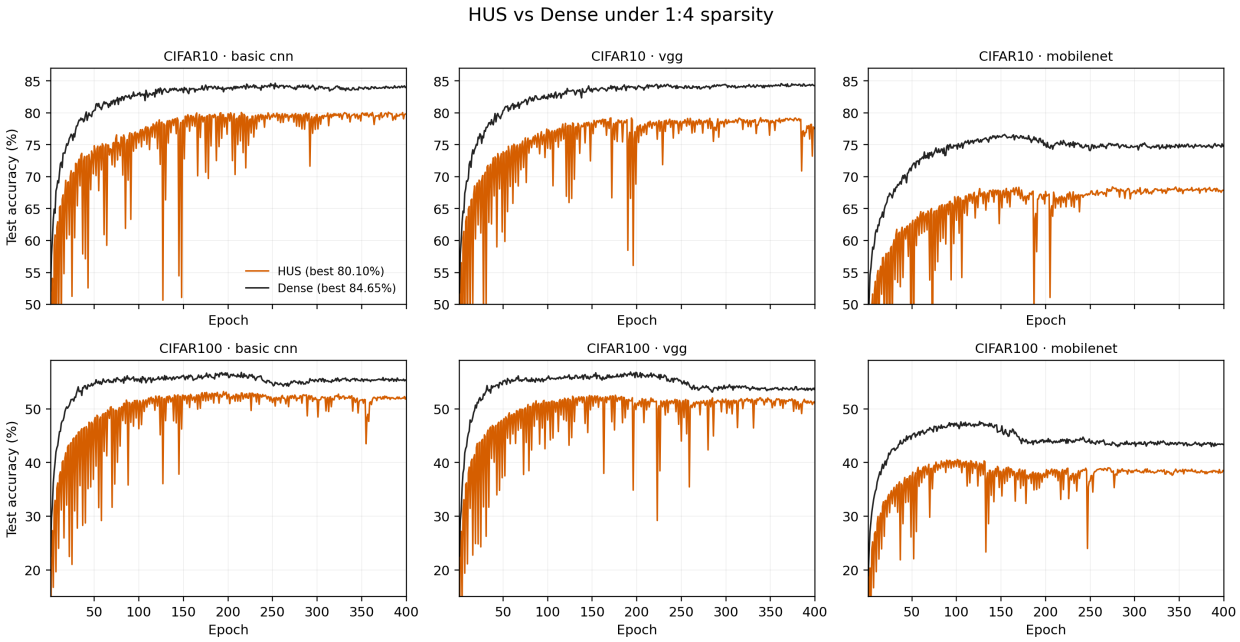

Figure 6: Training dynamics of HUS and dense training under 1:4 structured sparsity. Results are shown on CIFAR-10 and CIFAR-100 using Basic CNN, VGG, and MobileNet architectures. HUS denotes the sparse model, while Dense denotes the corresponding dense baseline.

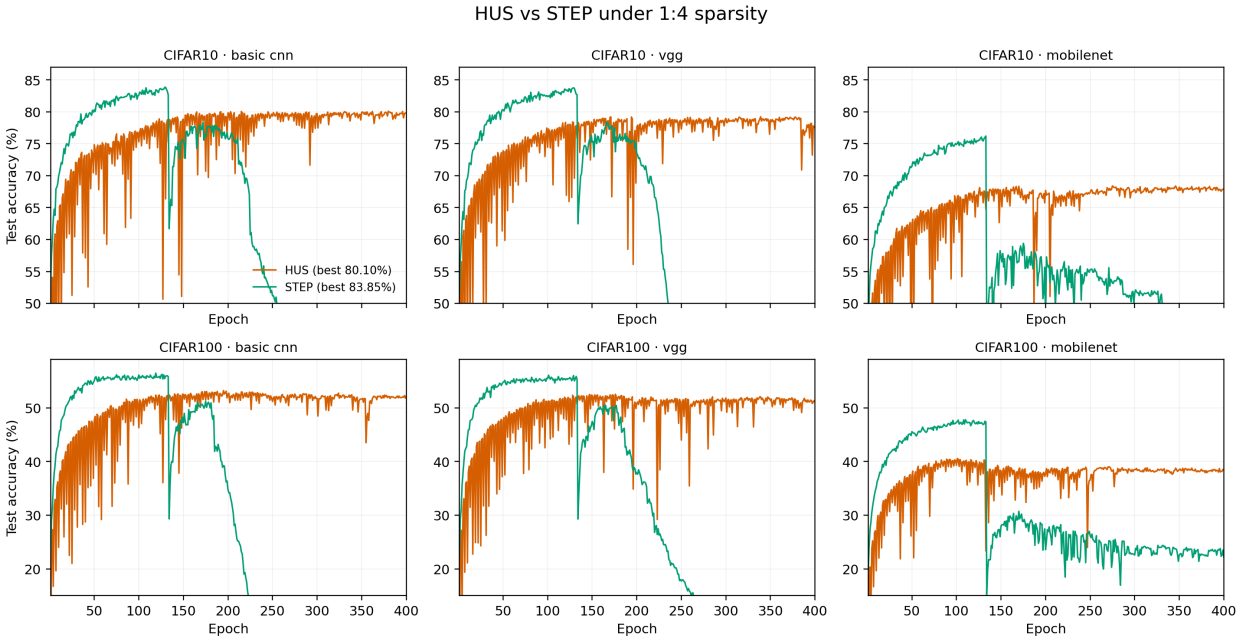

Figure 7: Training dynamics of HUS and STEP under 1:4 structured sparsity on CIFAR-10 and CIFAR-100. The same six dataset-architecture configurations are evaluated over 400 epochs.

J.2 Comparison with STEP

Figure 7 reveals a substantial difference between the optimization dynamics of HUS and STEP. STEP often reaches high accuracy during the early stage of training and can achieve a higher peak accuracy than HUS. However, these high-accuracy solutions are not consistently preserved throughout training.

For Basic CNN and VGG, STEP exhibits pronounced accuracy degradation after approximately the middle stage of training. On CIFAR-10, its accuracy initially rises above the HUS trajectory but subsequently drops sharply. A similar pattern occurs on CIFAR-100. The degradation is particularly severe for Basic CNN and VGG, where the late-stage trajectory falls far below the earlier peak.

MobileNet shows a related but less abrupt failure mode. STEP initially reaches substantially higher accuracy, followed by a sharp transition to a much lower and highly fluctuating regime. In contrast, HUS maintains a substantially more stable trajectory after reaching its accuracy plateau.

These results illustrate an important distinction between peak performance and training stability. A sparse method may discover a high-performing subnetwork at an intermediate epoch without preserving that solution during subsequent optimization. HUS does not always obtain the highest transient peak, but its training dynamics are considerably more stable over the full training horizon in these experiments. This observation also explains why comparisons based only on the best checkpoint can obscure substantial differences in sparse optimization behavior.

J.3 Comparison with SpikeNM

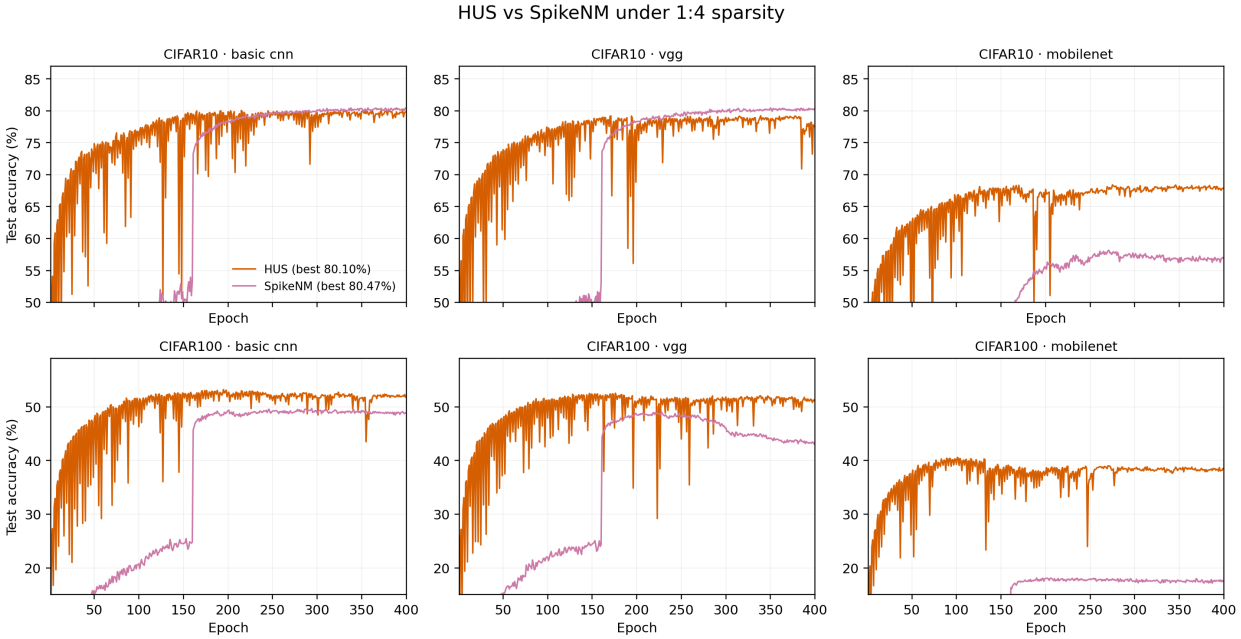

Figure 8: Training dynamics of HUS and SpikeNM under 1:4 structured sparsity on CIFAR-10 and CIFAR-100 using Basic CNN, VGG, and MobileNet architectures.

Figure 8 compares HUS with SpikeNM, an SNN-specific structured pruning method. The two approaches exhibit markedly different optimization trajectories. HUS improves progressively from the beginning of training, whereas SpikeNM shows a delayed transition before reaching its high-accuracy regime for Basic CNN and VGG.

For these two architectures, SpikeNM eventually approaches the performance of HUS, particularly on CIFAR-10. On CIFAR-100, HUS maintains a modest advantage after convergence. The difference becomes more pronounced for MobileNet. SpikeNM remains substantially below HUS throughout the later stage of training on both CIFAR-10 and CIFAR-100, suggesting greater sensitivity to the interaction between aggressive structured sparsity and compact network architectures.

Unlike STEP, SpikeNM does not exhibit the same pronounced late-stage collapse for Basic CNN and VGG. Instead, its primary characteristic in these experiments is delayed optimization followed by a relatively stable plateau. HUS

therefore differs from the two baselines in different ways: compared with STEP, it provides more stable late-stage training, while compared with SpikeNM, it reaches effective sparse solutions earlier and behaves more robustly on MobileNet.

J.4 Summary of Training Dynamics

Across the six evaluated dataset-architecture configurations, the training curves reveal information that is not captured by final accuracy alone. Dense training provides the expected upper reference, while the structured sparse methods exhibit substantially different optimization dynamics. STEP can achieve strong intermediate performance but suffers from pronounced late-stage degradation in several settings, whereas SpikeNM exhibits delayed optimization and weaker performance on MobileNet. HUS maintains a more stable long-term accuracy regime across the evaluated configurations, although transient accuracy drops can still occur during training.

Together with the final-accuracy and neuronal-activity results in Section 5.2, these observations provide a more complete characterization of HUS beyond a single best checkpoint. In practice, because dynamic sparse training may exhibit transient fluctuations, we recommend using a short validation window near the end of training for checkpoint selection rather than relying solely on the last epoch. Further improving the stability of topology evolution and reducing such transient perturbations represents an important direction for future work.

K Limitations

The current empirical study evaluates HUS primarily under moderate $N{:}M$ sparsity ratios and standard image classification benchmarks. Extending the evaluation to more aggressive compression regimes, large-scale datasets, and broader SNN architectures would provide a more comprehensive understanding of HUS under stringent constraints. Although HUS remains stable across the tested hyperparameter configurations, its dynamic topology updates can introduce transient accuracy fluctuations during training. In practice, a short validation window near the end of training can be used for checkpoint selection; developing more principled adaptive scheduling strategies for topology updates and homeostatic regulation remains an important direction.

The theoretical guarantees established in this work rely on assumptions that separate the evolution of sparse topology from weight optimization. Relaxing these assumptions and characterizing HUS under more general training dynamics would broaden its theoretical scope. On the systems side, integrating HUS with neuromorphic hardware supporting on-chip plasticity could enable closed-loop learning with physically instantiated sparse topologies. Finally, extending HUS to continual learning and spiking transformers would help evaluate whether its topology exploration and homeostatic mechanisms remain effective in more complex and dynamic settings.