Constructing Brain-Inspired Sparse Topologies for Energy-Efficient ANN-to-SNN Conversion via Cannistraci-Hebb Training

Kangyou Bao †, Wenqi Gu †, Jiaqing Lyu, Carlo Vittorio Cannistraci *

Posted Date: 14 February 2026

doi: 10.20944/preprints202602.1162.v1

Keywords: spiking neural networks; ANN-to-SNN conversion; dynamic sparse training; CannistraciHebb Training; energy-efficient architectures; brain-inspired topology

 doi:10.20944/preprints202602.1162.v1

Article

Constructing Brain-Inspired Sparse Topologies for Energy-Efficient ANN-to-SNN Conversion via Cannistraci-Hebb Training

Kangyou Bao [1,2,†], Wenqi Gu [1,3,†], Jiaqing Lyu [1,4] and Carlo Vittorio Cannistraci [1,3,4,*]

[1] Center for Complex Network Intelligence (CCNI), Tsinghua Laboratory of Brain and Intelligence (THBI), Department of Psychological and Cognitive Sciences, Tsinghua University, Beijing, China
[2] Department of Physics, Tsinghua University, Beijing, China
[3] School of Biomedical Engineering, Tsinghua University, Beijing, China
[4] Department of Computer Science and Technology, Tsinghua University, Beijing, China
* Correspondence: kalokagathos.agon@gmail.com
† Equal contribution.

Abstract

While ANN-to-SNN conversion is a pivotal approach to obtain SNNs, current methods mostly focus on dense architectures, disregarding the structural sparsity fundamental to brain neural networks. To bridge this gap, we propose a novel framework that integrates Cannistraci-Hebb Training (CHT)—a brain-inspired Dynamic Sparse Training algorithm—to instill biologically plausible topologies into SNNs. Through our framework, the converted SNNs directly inherit emergent brain-like properties, such as meta-depth and small-worldness, from the sparse ANNs. We confirm the brain-like topology trained by CHT and then investigate our framework across different conversion approaches. Our approach achieves comparable or superior accuracy to dense counterparts on both convolutional neural networks (CNNs) and Vision Transformer (ViT), while reducing theoretical energy consumption by over 60%. Empirically, we validate the framework's superiority over pruning baselines and direct SNN sparse training in terms of the accuracy-energy trade-off.

Keywords: spiking neural networks; ANN-to-SNN conversion; dynamic sparse training; Cannistraci-Hebb Training; energy-efficient architectures; brain-inspired topology

1. Introduction

Spiking neural network (SNN) is a type of biologically plausible neural network inspired by the brain. Unlike neurons in ANNs, spiking neurons process information in temporal dimension with event-driven spikes (Eshraghian et al. 2023). Because of the event-driven nature of SNNs (Modaresi et al. 2023) and temporal sparsity of activations, SNNs offer the potential to be far more energy-efficient than conventional ANNs on neuromorphic hardware (Davies et al. 2018; DeBole et al. 2019; Huo et al. 2023; Merolla et al. 2014; Zhang et al. 2024a).

Currently, methods to obtain high performance SNN can be categorized into two streams: direct training and ANN-to-SNN Conversion (Roy et al. 2019). In practice, however, directly training SNNs remains challenging due to the non-differentiability of spiking neurons (Li et al. 2024), often resulting in a persistent accuracy gap between SNNs and their ANN counterparts.

Consequently, ANN-to-SNN conversion has gained prominence as an alternative. Conversion circumvents the hurdles of direct training by utilizing well-established ANN training methods, often yielding close-to-ANN accuracy with low latency (Huang et al. 2025; Yang et al. 2025; You et al. 2024).

However, to the best of our knowledge, prior ANN-to-SNN conversion methodologies have predominantly centered on dense architectures (Huang et al. 2025; Yang et al. 2025; You et al. 2024). This focus presents a significant yet under-explored opportunity: introducing structural connection

 doi:10.20944/preprints202602.1162.v1

sparsity into SNN conversion could combine benefits from both sides. The event-driven nature of SNNs reduces computation at temporal level, and introducing sparsity into connection structure can further reduce computation at structural level (Tmamna et al. 2024). Furthermore, from a biological perspective, brain networks—the very inspiration for SNNs—are inherently sparse, with network properties of hyperbolic meta-deep structure, scale-freeness and small-worldness simultaneously (Cacciola et al. 2019; Cannistraci et al. 2013; Narula et al. 2017).

To bridge this gap, we propose a novel framework that integrates network-science-driven dynamic sparse training (DST) (Evci et al. 2020; Jayakumar et al. 2020; Mocanu et al. 2018; Yuan et al. 2021; Zhang et al. 2024c) into the SNN conversion pipeline which effectively instills brain-inspired topology into the network before conversion. Specifically, we leverage Cannistraci-Hebb Training (CHT), a state-of-the-art (SOTA) brain-inspired DST family recognized for consistently outperforming traditional DST baselines while delivering performance that rivals or even surpasses dense counterparts (Hanming et al. 2025; Zhang et al. 2024c, 2025). Crucially, inspired by the notion from Hebbian learning (Hebb 2005) that "neurons that fire together, wire together", CHT naturally induces self-organizing topology characterized by fundamental brain-network properties, such as scale-freeness and small-worldness (Zhang et al. 2024c). Our framework converts these CHT-trained ANNs to SNNs, ensuring that the resulting sparse SNNs directly inherit these brain-inspired topologies. As a consequence, the SNNs produced by our framework are not simply sparse in parameter count, but possess brain-like topologies that are intrinsically aligned with neuromorphic dynamics.

To empirically validate our framework, we conduct extensive experiments across diverse architectures and scales. First, from a network science perspective, our topological analysis confirms that CHT-trained networks successfully evolve into structures exhibiting brain-like properties. Building on this structural foundation, we evaluate the conversion performance on Convolutional Neural Networks (CNNs) using two distinct conversion methods: CS-QCFS (Yang et al. 2025) and DCGS (Huang et al. 2025). Our results demonstrate that the proposed framework achieves competitive accuracy compared to dense counterparts while yielding substantial energy savings. Notably, through rigorous ablation studies, we find that leveraging CHT instead of static pruning for ANN-to-SNN conversion usually leads to an impressive increase of both accuracy and energy efficiency. Furthermore, our method consistently outperforms SOTA direct SNN sparse training baselines: DPAP (Han et al. 2024) and SD-SNN (Han et al. 2025) , highlighting the efficacy of our strategy for sparse SNNs. Finally, to demonstrate scalability, we apply our framework to Vision Transformer (Dosovitskiy 2020) on ImageNet-1K (Deng et al. 2009). Our method achieves a remarkable 60% reduction in theoretical energy consumption compared to dense counterparts with a trivial accuracy drop (approx. 1%), also demonstrating superior accuracy-efficiency trade-offs compared to SOTA static pruning baselines.

Contributions of our work are summarized as follows:

1. We propose the first ANN-to-SNN conversion framework that incorporates brain-inspired Dynamic Sparse Training. By leveraging CHT, we enable the biologically plausible topologies that are inherited by the converted SNNs.
2. We empirically demonstrate the superiority of our framework over static pruning and direct training baselines on CNNs. On Vision Transformer, our framework reduces theoretical energy consumption by over 60% while maintaining comparable accuracy.

2. Related Works

ANN-to-SNN conversion. Methodologies can be primarily distinguished by their neural coding schemes. The prevalent paradigm relies on rate coding (Adrian 1926), where the firing rate of SNN neurons approximates the activation values of ANNs. A foundational work on CNNs is QCFS, which replaces ReLU with quantized function and trains the firing thresholds of IF neuron on ANNs (Bu et al. 2023). Subsequent studies have further refined this paradigm (Hao et al. 2023; Wang et al. 2024; Yang et al. 2025); for instance, CS-QCFS (Yang et al. 2025) introduces a channel-wise mechanism to adapt to varying activation across channels and incorporates a softplus function to ensure positive thresholds.

 doi:10.20944/preprints202602.1162.v1

To further enhance the expressivity of traditional IF neuron, methods utilizing multi-threshold neurons are proposed (Hu et al. 2023; Wang et al. 2022).

These principles also extend to Transformer-based ANN-to-SNN conversion. One recent work SpikeZIP-TF (You et al. 2024) adapts concept of bi-threshold neuron and quantized activation to achieve high performance conversion. This method requires extensive training, incurring rather high computational costs for Transformers.

To address the limitation of rate coding that if suffers from information decay due to time-averaging, differential coding is proposed in DCGS (Huang et al. 2025), which is a training-free method can be applied on both CNN and Transformer where spikes are treated as corrections to the encoded value. Also, DCGS leverages analytical derivation to identify optimal thresholds, which allows a rapid conversion of high-precision, low-latency SNNs.

Sparse Artificial Neural Networks. Sparsification methods for ANNs can be broadly categorized into pruning and Dynamic Sparse Training. Existing literature identifies Pruning After Training as the most effective paradigm, generally outperforming pruning before or during training (Cheng et al. 2024). A representative SOTA structural pruning (after training) method is DepGraph (Fang et al. 2023), which enforces simultaneous pruning across layers while ensuring removed parameters are consistently unimportant. In the context of Transformers, One-Shot Post-Training Pruning (PTP) has been extensively developed to mitigate computational constraints (Cheng et al. 2024). Leading methods include SparseGPT (Frantar & Alistarh 2023), RIA (Zhang et al. 2024b), and Wanda (Sun et al. 2023). However, pruning operates by exclusively removing connections without a mechanism to regrow them, thereby forfeiting the opportunity to explore optimal topologies.

Instead of pruning, another way to obtain sparse ANNs with close-to-dense performance is Dynamic Sparse Training. Pioneering works in this domain include Sparse Evolutionary Training (SET) (Mocanu et al. 2018), which introduces random connection regrowth, and RigL (Evci et al. 2020), which utilizes gradient information from non-existing links to guide regrowth. While effective, these methods typically rely on random or gradient-based heuristics that lack explicit topological grounding. Addressing this limitation, Zhang et al. 2024c propose CHT that adopts a gradient-free regrowth strategy rooted in network science. CHT has been demonstrated to outperform standard DST baselines (e.g. SET and RigL) across various tasks (Zhang et al. 2024c, 2025)(elaborated in Section 3.2), and has recently been extended to convolutional architectures as CHT-Conv (Hanming et al. 2025).

Network science informed deep learning. The theoretical foundation of CHT traces back to the Local Community Paradigm (LCP) proposed by (Cannistraci et al. 2013). This work revealed that biological brain networks significantly adhere to the LCP organization. Building on this, subsequent studies applied this concept to study brain functionalities (Cacciola et al. 2019; Narula et al. 2017), leading to the conceptualization of "Epitopological Learning" (EL) (Cannistraci 2018). Rooted in the Hebbian learning (Hebb 2005), EL means to learn via changing the shape of connectivity structures. This theoretical framework was further formalized into the Cannistraci-Hebb (CH) rule, developed as a link prediction automaton (Zhao et al. 2025). Recently, Zhang et al. 2024c adapted this rule to the domain of deep learning, employing it as the regrowth criterion in Dynamic Sparse Training that surpasses fully connected networks in some tasks.

3. Methods

In this section, we begin by introducing the Cannistraci-Hebb Training (CHT) mechanism and the proposed framework for topological inheritance. While the core framework is detailed here, specific descriptions of the used ANN-to-SNN conversion methods are provided in Appendix A. Subsequently, we formulate the theoretical energy estimation model for SNNs to quantify the efficiency gains from sparsity.

 doi:10.20944/preprints202602.1162.v1

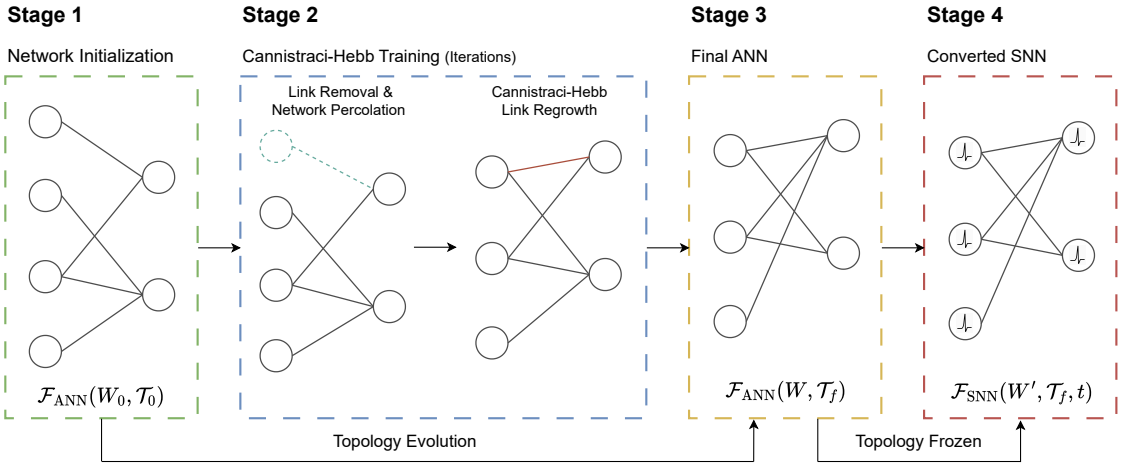


Figure 1. Illustration of our framework. The process consists of two main phases: Cannistraci-Hebb Training and SNN conversion. (Stages 1–3) The sparse network topology and weights are initialized (Stage 1) as $\mathcal{F}_{\text{ANN}}(W_0, \mathcal{T}_0)$. CHT is applied iteratively during training(Stage 2), where both weights and topology are learned and results in final ANN as $\mathcal{F}_{\text{ANN}}(W, \mathcal{T}_f)$ (Stage 3). (Stage 4) The trained sparse ANN is converted into a SNN as $\mathcal{F}_{\text{SNN}}(W', \mathcal{T}_f, t)$. Crucially, the topology remains frozen during this conversion, meaning the SNN inherits the sparse topology $\mathcal{T}_f$ learned by CHT.

3.1. Cannistraci-Hebb Training

Cannistraci-Hebb Training (CHT) (Hanming et al. 2025; Zhang et al. 2024c, 2025) is a family of network-science-driven and brain-inspired DST algorithms. It implements Epitopological Learning—a mechanism that aims to learn not only by optimizing weights but by evolving the network topology (Cannistraci 2018). During topology evolution, a fraction of connections are canceled and new connections are grown using Cannistraci-Hebb link prediction.

At the core of the link prediction lies the Cannistraci-Hebb rule. As illustrated in Figure 2, the rule leverages the concept of Local Community Paradigm (Cannistraci et al. 2013). Specifically, the L3 Common Neighbors (CNs) of a node pair are defined as connected nodes on any length-3 path of the node pair, which form the local community of the node pair. Among the links of CNs, links are categorized into internal Local Community Links (iLCL)—connections between CNs—and external Local Community Links (eLCL)—connections to the rest of the network. While specific variants are applied in this work(CH2 for linear layers (Zhang et al. 2025) and CH3 for convolutional layers (Hanming et al. 2025)), the shared principle of CH family is to minimize the external Local Community Link by penalizing eLCLs. This minimization effort aligns with the Hebb rule: "neurons that wire together fire together". Specifically, the CH rule drives the network to form relatively isolated local communities, which is equivalent to forming a barrier for local communities and effectively configuring the 'wiring' necessary for coordinated activity. Therefore this method enables neurons to 'fire together'-maintaining and reinforcing the signaling-within their respective communities, thereby leveraging local communities to facilitate functional and representational specialization.

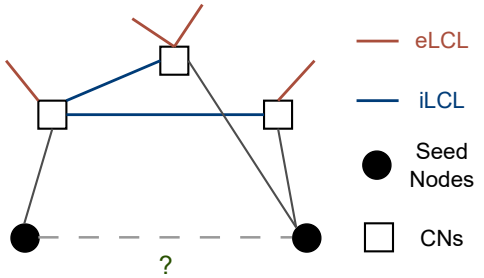


Figure 2. Local Community. The local community of two seed nodes is formed by the Common Neighbours(CNs) on length-3 path. Links of common neighbours can be categorized as internal Local Community Link(iLCL)(in blue) and external Local Community Link(eLCL)(in red).

 doi:10.20944/preprints202602.1162.v1

Stages 1–3 in Figure 1 illustrate the operational procedure of CHT. Initially, a sparse network topology and weights are initialized. For each topology evolution iteration during training, CHT first prunes weights according to weight magnitude, and then percolate the inactive neurons. After removal and percolation, new links are sampled and regrow according to scores calculated by Cannistraci-Hebb link prediction rule. Please refer to Appendix B for more details.

3.2. Proposed Framework

The key motivation of this work is to equip SNNs with biologically plausible topologies. Given that CHT induces self-organizing brain-like properties in ANNs which is verified in Section 4.2, we propose a framework that ensures these properties are preserved during conversion.

As shown in Figure 1, the framework proceeds in four stages. Stages 1–3 execute CHT to evolve the topology. In the final transition (Stage 3 $\rightarrow$ Stage 4), we freeze the learned topology $\mathcal{T}_f$ during the SNN conversion process, which ensures that the structural advantages of the sparse ANN are directly transferred to the SNN:

$$\mathcal{F}_{\text{ANN}}(W, \mathcal{T}_f) \xrightarrow[\text{Topology Invariant}]{\text{SNN Conversion}} \mathcal{F}_{\text{SNN}}(W', \mathcal{T}_f, t) \tag{1}$$

where W and W' represents the weight of ANN and SNN respectively (which is not necessarily equal).

This structural motivation is further bolstered by CHT's established performance superiority over conventional DST counterparts in the ANN domain. Prior evaluations on CNNs and MLP structures have shown that CHT consistently outperforms DST baselines including RigL and SET, particularly at sparsity as high as 99% (Zhang et al. 2024c). Furthermore, recent benchmarks in LLMs indicate that CHT achieves lower perplexity on language modeling tasks and superior zero-shot performance on the GLUE benchmark compared to other DST baselines (Zhang et al. 2025). Critical for our work, CHT provides a foundation of high-performance sparse ANN for the subsequent conversion process.

3.3. SNN Energy Consumption Estimation

Leveraging event-driven neuromorphic hardware, the forward computation of an SNN layer is triggered exclusively upon the emission of spikes (Eshraghian et al. 2023; Modaresi et al. 2023). Importantly, synaptic operations (SOPs) occur only at existing synaptic links on neuromorphic hardware (Davies et al. 2018; DeBole et al. 2019; Huo et al. 2023; Merolla et al. 2014; Zhang et al. 2024a), thereby reducing computation through not only temporal activation but also structural sparsity.

Beyond sparsity, a critical factor contributing to SNN efficiency is the distinct nature of SOP. Due to the binary spikes (0 or 1), the dominant operation shifts from the energy-intensive Multiply-and-Accumulate (MAC) to the simpler Accumulation (AC) (Yao et al. 2023). It is important to note that if input data is not specifically encoded into spikes, the operation type of first layer remains MAC.

To quantify theoretical energy efficiency, we adopt standard energy constants from 45nm technology (Yao et al. 2023), where $E_{\text{MAC}} = 4.6\,\text{pJ}$ and $E_{\text{AC}} = 0.9\,\text{pJ}$. Consequently, the total inference energy of the network is formulated as:

$$E_{\text{total}} = N_{\text{MAC}} \cdot E_{\text{MAC}} + N_{\text{SOP}} \cdot E_{\text{AC}} \tag{2}$$

where N_{MAC} denotes the operations in input layer, and N_{SOP} represents the total number of spike-based accumulation operations (Synaptic Operations) across all subsequent spiking layers. Unless otherwise specified, all energy metrics reported in the Experiments and Appendix refer to the average inference energy per image on the test set.

4. Experiments

4.1. Experimental Setup

Datasets and Architectures. We evaluate our framework on three standard benchmarks: CIFAR-10, CIFAR-100 (Krizhevsky et al. 2009), and ImageNet-1K (Deng et al. 2009). For CNN experiments on

 doi:10.20944/preprints202602.1162.v1

CIFAR datasets, we utilize VGG-16 and ResNet-18 (He et al. 2016). Following standard practices for CIFAR datasets, we adapt VGG-16 by reducing the dense layer dimension from 4096 to 512 (Liu & Deng 2015). We also scale our experiments to ImageNet-1K using base size Vision Transformer (ViT-B) (Dosovitskiy 2020).

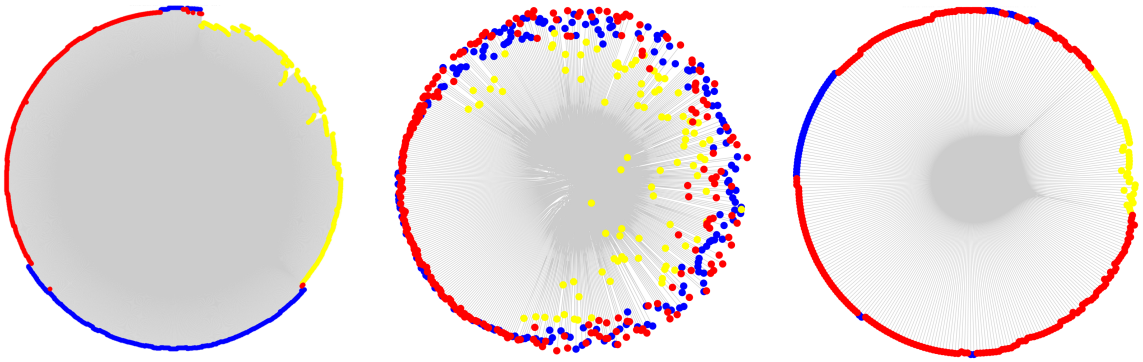
Implementation Details. Due to the page limit, we provide training details in Appendix C for reproducibility.

4.2. Network Science Analysis

To rigorously validate the "brain-inspired" claim of our proposed framework, we move beyond standard performance metrics and investigate the underlying structural organization of the evolved networks. For detailed network analysis methods, please refer to Appendix D.

We focused our analysis on the classifier segment of the VGG-16 architecture trained on CIFAR-10, comparing our model with Depgraph-pruned model (the ANNs presented in Section 4.3). The last two fully connected layers are used for analysis. This network encompasses three layers of neurons, where the sparsity level reaches 99%. We projected the connectivity patterns of these layers into a 2D hyperbolic disk. In Figure 3, the nodes are color-coded by layer depth: Red (input of the first linear layer), Yellow (hidden layer), and Blue (output layer).

Emergence of Meta-depth. As observed in the visualization of our topology (Figure 3b), the network manifests a distinct structural organization. The nodes self-organize into concentric layers, exhibiting a phenomenon known as "Meta-depth" (Zhang et al. 2024c). Meta-depth means that the network learns a topological hierarchical hub-centric organization where some nodes are deeper (with higher connections) than others in the connectivity hierarchy and are more important than the others to control the construction of meta-features (Zhang et al. 2024c). Indeed, the visual stratification in Figure 3b indicates that our method generates a hyperbolic and hierarchical flow of information that support performance even at extreme sparsity (Zhang et al. 2024c). The deeper nodes in the sparse network are the nodes in the middle layer (yellow nodes), which control the generation flow of meta-features in hidden layers. In stark contrast, the topology resulting from the DepGraph-pruned topology (Figure 3c) does not display hyperbolic meta-depth organization with no discernible hierarchy between the layers of neurons after projection, closely resembling the random initialization (Figure 3a). The meta-deep hyperbolic structure is a typical feature of brain networks (Allard & Serrano 2020; Cacciola et al. 2017).



(a) Init. ($\gamma = 6.91$) **(b) Ours** ($\gamma = 2.19$) **(c)** prune ($\gamma = 9.46$)

Figure 3. Hyperbolic topological embedding of VGG-16 linear layers at 99% sparsity. The nodes are colored by depth (Red→Yellow→Blue). (a) Topology at Randon Initialization. (b) Topology of our network. (c) Topology of Depgraph-pruned model.

Scale-Free Properties. Quantitatively, we measured the degree distribution exponent, γ. The topology of our method yields a γ value of **2.19**. In network science, a γ value between 2 and 3 is the signature of a *scale-free network*—a vital characteristic of biological neural systems (Lynn et al. 2024) with hierarchical organization that ensures efficient communication via hub nodes. Conversely, the

 doi:10.20944/preprints202602.1162.v1

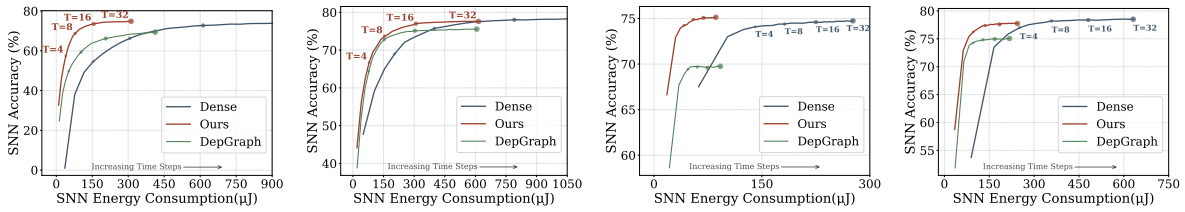
measured γ value for DepGraph baseline is **9.46**. A γ value significantly larger than 3 indicates a deviation from the scale-free regime, pointing towards a topological structure that lacks the efficient hierarchy associated to scale-freeness.

This analysis provides compelling empirical evidence that our method does not merely sparsify the network; it fundamentally reshapes the connectivity into a hyperbolic, meta-deep and scale-free, brain-like topology. This structural superiority potentially underlies the energy-efficiency gains observed in our SNN conversion, as the topology is intrinsically aligned with the event-driven nature of neuromorphic dynamics.

4.3. Performance Evaluation over Dense and Pruning Baseline

Table 1. SNN performance evaluation on CIFAR-10/100 using VGG-16 and ResNet via CS-QCFS and DCGS conversion methods. Metrics format: For Dense Baselines, absolute values are reported. For sparse models (DepGraph and Ours), Accuracy is reported as the deviation from Dense (Δ, where ↓ indicates loss), and Energy is reported as the **Reduction Percentage** compared to Dense (higher is better). **Bold** indicates better accuracy / energy reduction between Depgraph and Ours.

Dataset	Model	Method	ANN acc(%)	DCGS ($T = 8$) SNN acc(%) Abs. / Δ	DCGS ($T = 8$) Energy (μJ) Abs. / Red.	CS-QCFS ($T = 16$) SNN acc(%) Abs. / Δ	CS-QCFS ($T = 16$) Energy (μJ) Abs. / Red.
CIFAR10	VGG-16	Dense	94.45	94.21	139.19	93.71	356.74
		Depgraph	92.73	↓1.54	64.00%	↓2.76	55.66%
		Ours	93.98	**↓0.26**	**66.76%**	**↓0.86**	**66.63%**
	ResNet-18	Dense	95.29	95.14	289.44	94.60	648.40
		Depgraph	94.50	↓0.84	**63.77%**	↓1.27	65.48%
		Ours	95.19	**↑0.12**	59.04%	**↓0.67**	**71.72%**
CIFAR100	VGG-16	Dense	74.82	74.60	225.05	72.73	612.29
		Depgraph	69.85	↓5.01	66.89%	↓9.19	70.15%
		Ours	75.13	**↑0.47**	**69.45%**	**↑0.79**	**74.46%**
	ResNet-18	Dense	78.41	78.37	480.98	78.03	790.95
		Depgraph	75.08	↓3.42	**65.03%**	↓3.87	**66.34%**
		Ours	77.83	**↓0.77**	61.66%	**↓1.04**	60.98%



(a) VGG-16 via CS-QCFS **(b)** ResNet-18 via CS-QCFS **(c)** VGG-16 via DCGS **(d)** ResNet-18 via DCGS

Figure 4. Pareto Frontiers of Accuracy vs. Energy on CIFAR100. We plot the energy-accuracy trajectory across time steps $T \in [1, 32]$. Each figure contains trajectories of CHT-sparse SNN, dense counterpart and static pruning baseline. *Note: The curves of Dense SNNs converted via CS-QCFS are truncated due to excessive energy consumption.*

To validate the effectiveness of our framework, we first benchmark SNNs converted by our framework(denoted as CHT-sparse SNNs) against their dense counterparts and conduct an ablation study on the sparsifying method. Specifically, we replace the dynamic CHT in our framework with a SOTA pruning method. We select DepGraph (Fang et al. 2023) as a representative and formidable baseline because Pruning After Training (PAT) has been identified in existing literature as the most effective paradigm (Cheng et al. 2024). By keeping the rest of the pipeline identical, this comparison isolates the impact of CHT's dynamic topological evolution versus static pruning. We evaluate these methods on CIFAR-10/100 using VGG-16 and ResNet-18 architectures across two conversion methods: CS-QCFS and DCGS. For a fair assessment, the layer-wise sparsity, ANN training iterations and entire SNN conversion procedure of Dense and DepGraph baselines are aligned with our CHT models. For

 doi:10.20944/preprints202602.1162.v1

all sparse models, the parameter sparsity for convolutional layers is 78% and parameter sparsity of linear layers is 99%.[1] More implementation details can be found in Appendix C.

For conciseness, we hereafter refer to SNNs converted from pruned ANNs simply as 'the pruning baseline' throughout the remainder of this paper. Also, CHT-sparse SNN can be interpreted as sparse SNN converted from proposed framework.

Table 1 summarizes the quantitative results. We report the performance of SNNs converted via DCGS at $T = 8$ time steps and SNNs converted via CS-QCFS at $T = 16$ time steps. The reason for choosing of T based on analysis of accuracy saturation can be found in E.1.

Generally, our CHT-SNNs achieve substantial efficiency gains, delivering over 60% energy reduction compared to Dense counterparts in most cases (7 out of 8 scenarios), with negligible accuracy loss ($< 1\%$) or even slight improvements (+0.79%). Crucially, when comparing the proposed CHT against static pruning (DepGraph): (1) In 5 out of 8 scenarios, CHT-SNNs surpass the pruning baseline in both accuracy and energy efficiency. (2) In the remaining cases (primarily ResNet-18), while static pruning yields marginally higher energy reduction, it suffers from severe accuracy degradation. For instance, on CIFAR-100, CHT-SNNs recover approximately 2.6% $\sim$ 2.8% of the Top-1 accuracy lost by DepGraph, with only a small trade-off in energy reduction.

To visualize the accuracy - energy trade-off, we plot the Pareto frontiers of SNNs on CIFAR100 in Figure A1. We observe that the trajectories of CHT-sparse SNNs (Ours, red curves) predominantly occupy the optimal top-left region relative to the baselines. This positioning indicates that our framework generally achieves a more favorable accuracy-energy balance than pruning baseline and dense SNN. Due to the page limit, pareto plots on CIFAR10 are provided in Appendix E.2.

Finally, comparing the two conversion methods, CHT-sparse SNN converted via DCGS generally exhibits better accuracy compared to CS-QCFS, particularly considering lower time steps $T = 8$ compared with $T = 16$ for CS-QCFS.

4.4. Comparison with STBP-Based Direct SNN Sparse Training

In this section, we compare our framework against SOTA STBP-based (Wu et al. 2018) direct SNN sparse training methods: DPAP (Han et al. 2024) and SD-SNN (Han et al. 2025), which represent competitive baselines for obtaining high-performance sparse SNNs. We employ a widely-used CNN architecture (6 convolutional layers and 2 fully-connected layers) on CIFAR-10/100 (Han et al. 2024, 2025) in these experiments. To ensure a fair comparison, we align the sparsity of our CHT-trained models with the uncontrollable sparsity levels produced by the baseline methods for convolutional layers and linear layers separately. All models are trained equally for total 600 epochs, including ANN training and possible training in SNN conversion stage. All SNNs' performance are evaluated with $T = 8$ time steps. More implementation details can be found in Appendix C.

Table 2 summarizes the SNN performance comparison. The results demonstrate that our CHT-sparse SNN (via DCGS conversion method) consistently outperforms both DPAP and SD-SNN baselines across all datasets and metrics. Specifically, on the challenging CIFAR-100 dataset, our method achieves a 1.16% accuracy gain over SD-SNN while simultaneously reducing energy consumption by 42.13% compared to DPAP under similar sparsity.

To further explore the potential of CHT in the high-sparsity regime, we extend our evaluation to network sparsity levels around 70% (10% for convolutional layers and 90% for linear layers) and around 90%(60% for convolutional layers and 99% for linear layers).

Table 3 presents the sensitivity analysis. Remarkably, on CIFAR-10, we observe an "inverse" trend where accuracy improves as sparsity increases (sparsity up to 90.62%). Compared to the moderate sparsity case ($\sim$35%), the ultra sparse model ($\sim$90%) not only reduces energy consumption by 29% but also achieves a higher top-1 accuracy (94.48%). Similarly, on CIFAR-100, the model at over 90% sparsity

[1] The final linear classification layer remains dense. Consequently, for the ResNet-18 architecture, which contains only a single linear layer for classification, the sparsification is applied exclusively to its convolutional layers.

 doi:10.20944/preprints202602.1162.v1

maintains a competitive accuracy of 75.25%, which still outperforms all direct training baselines operating at much lower sparsity levels.

Table 2. Comparison against direct SNN sparse training on CIFAR-10/100. We compare our framework using DCGS(denoted as Ours(DCGS)) and CS-QCFS(denoted as Ours(CS-QCFS)) conversion method against SOTA direct training baselines DPAP and SD-SNN with $T = 8$ time steps. **Bold** indicates best performance.

Dataset	Method	Network sparsity(%)	SNN acc(%)	Energy(μJ)
CIFAR10	DPAP	57.87	94.13	488.54
	Ours(DCGS)		**94.39**	**347.92**
	Ours(CS-QCFS)		93.91	394.94
	SD-SNN	35.26	94.32	368.92
	Ours(DCGS)		**94.34**	**356.02**
	Ours(CS-QCFS)		94.01	384.82
CIFAR100	DPAP	57.75	74.50	601.61
	Ours(DCGS)		**75.35**	**348.16**
	Ours(CS-QCFS)		75.32	542.25
	SD-SNN	37.02	74.86	426.31
	Ours(DCGS)		**76.02**	**344.12**
	Ours(CS-QCFS)		74.83	543.85

Table 3. Sparsity Sensitivity Analysis on CIFAR-10/100. We evaluate the performance of CHT-sparse SNN (DCGS) across varying sparsity regimes on CIFAR-10/100 (with $T = 8$ time steps). **Bold** indicates best accuracy and least energy consumption. Note that the ultra sparse models maintain or even improve accuracy while reducing energy consumption.

Dataset	Network sparsity(%)	SNN acc(%)	Energy(μJ)
CIFAR10	35.26	94.34	356.02
	57.87	94.39	347.92
	72.84	94.44	329.45
	90.62	**94.48**	**252.07**
CIFAR100	37.02	**76.02**	344.12
	57.75	75.35	348.16
	72.68	75.82	331.57
	90.42	75.25	**309.24**

4.5. Scalability to Vision Transformer

To evaluate the scalability of our framework, we extend our experiments to the large-scale ImageNet-1K dataset using base size Vision Transformer (ViT-B), employing DCGS as the conversion method. We compare SNN converted from CHT-trained sparse ANN (CHT-sparse SNN) with dense counterpart and SNNs converted from pruned ANNs using a series of SOTA static pruning baselines. Pruning baselines include SparseGPT (Frantar & Alistarh 2023), RIA with / without reconstruction (Zhang et al. 2024b)(denoted as RIA_{rec} and RIA respectively), Wanda with / without reconstruction (Sun et al. 2023)(denoted as $\text{Wanda}_{\text{rec}}$ and Wanda respectively), RI (Zhang et al. 2024b) and (Weight) Magnitude (Zhu & Gupta 2017). For sparse models, the sparsity is uniformly set to 70% for linear layers. More implementation details can be found in Appendix C. We choose to report SNNs' performance at $T = 8$ time steps, and its rationale can be found in Appendix E.1.

Table 4 summarizes the quantitative results. Remarkably, the SNN derived from our framework achieves 80.53% top-1 accuracy. Compared to the Dense SNN, this represents a massive 60.2% reduction in energy consumption (29.07 mJ vs. 73.06 mJ) with a negligible accuracy drop ($< 1\%$). Furthermore, our method establishes a significant performance gap against static pruning baselines. As shown in

 doi:10.20944/preprints202602.1162.v1

Table 4, our framework outperforms the nearest competitor (SparseGPT) by over 15% in accuracy while maintaining comparable inference energy.

Table 4. Performance comparison of SNNs on ImageNet-1K with ViT-B. All SNN models are evaluated with $T = 8$ time steps. **Metrics format:** For Dense Baselines, absolute values are reported. For sparse models, Accuracy is reported as the deviation from Dense (Δ, where ↓ indicates loss), and Energy is reported as the **Reduction Percentage** compared to Dense (higher is better).

Method	ANN acc(%)	SNN acc(%) Abs. / Δ	Energy(mJ) Abs. / Red.
Dense	81.8	81.45	73.06
Ours	80.82	↓0.92	60.21%
SparseGPT	66.46	↓16.18	60.69%
RIA_{rec}	66.24	↓15.95	61.00%
$\text{Wanda}_{\text{rec}}$	62.68	↓20.40	60.55%
RIA	57.70	↓24.88	62.07%
Wanda	51.00	↓32.86	61.57%
RI	36.18	↓45.41	64.15%
Magnitude	33.47	↓48.21	63.96%

This advantage is visualized in the accuracy-energy Pareto frontier (Figure 5). Our method (red curve) resides in the optimal top-left region, closely tracking the Dense baseline's accuracy trajectory but significantly shifted towards the low-energy regime. By contrast, all static pruning baselines suffer from a significant drop in accuracy (below 67%).

In this section, we exclude direct SNN training methods. Direct SNN training for Transformers demands substantial computation resources (Yao et al. 2025) while suffering from poor performance compared to its ANN counterpart (Li et al. 2022; You et al. 2024).

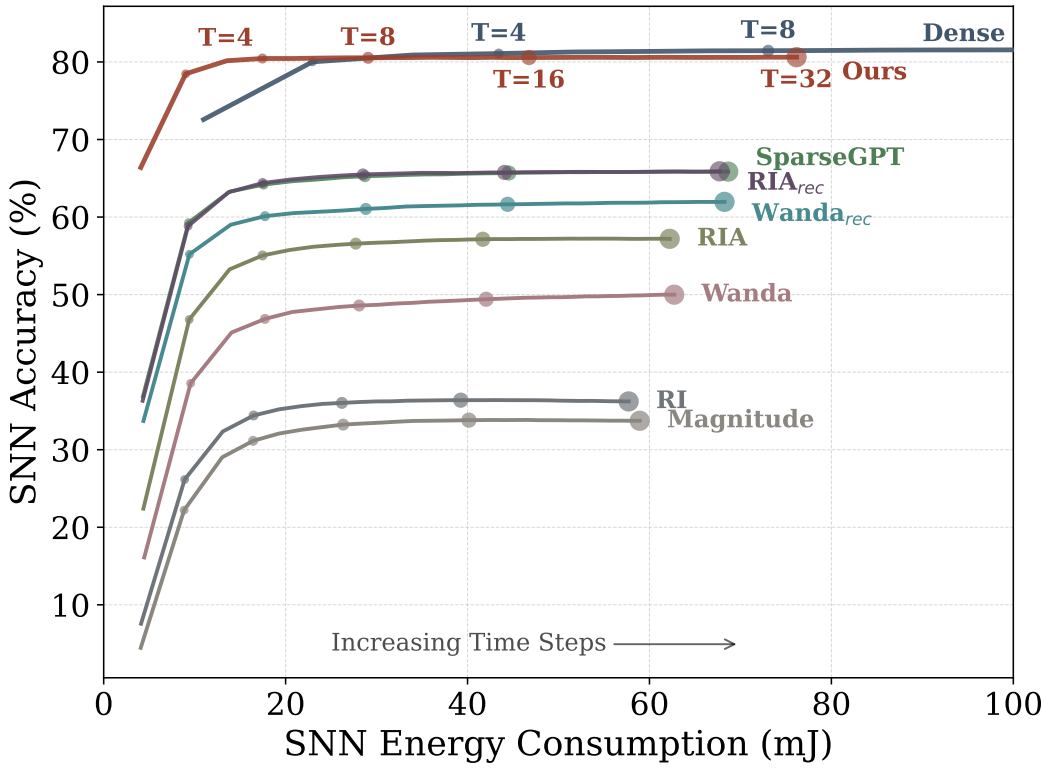


Figure 5. Pareto Frontier of Accuracy vs. Energy on ImageNet-1K. We plot the energy-accuracy trajectory across time steps $T \in [1, 32]$ for each SNN. *Note: The dense SNN's curve is truncated due to its excessively high energy consumption.*

 doi:10.20944/preprints202602.1162.v1

5. Conclusion

In this work, we proposed a framework that bridges network science and neuromorphic computing by integrating Cannistraci-Hebb Training (CHT) into the ANN-to-SNN conversion pipeline. Within this framework, the converted SNN directly inherits the brain-inspired topology evolved by CHT. Empirical results suggest that our approach offers superior or competitive accuracy-energy trade-offs relative to SOTA static pruning and direct SNN sparse training baselines across various CNN and Vision Transformer architectures.

While promising, our energy efficiency gains are currently based on theoretical estimations; future work can focus on deployment on physical neuromorphic hardware to validate real-world energy consumption. Additionally, the current scope of this work is primarily focused on static computer vision tasks; extending this methodology to other domains, such as dynamic vision tasks and Natural Language Processing, remains a subject for future investigation.

Impact Statement

This paper presents work whose goal is to advance the spiking neural network in the community of neuromorphic community for promoting Green AI. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Appendix A. Used ANN to SNN Conversion Methods

CS-QCFS(Yang et al. 2025). CS-QCFS converts ANNs to SNNs by first replacing ReLU with a Softplus Quantization Clip-Floor-Shift (S-QCFS) activation function on pre-trained ANNs for further training to learn learn layer-wise thresholds. Then, S-QCFS function is replaced with channel-wise S-QCFS(CS-QCFS) function to finetune channel-wise thresholds in convolution layers, which aims to capture heterogeneous activation distribution across channels. Finally, the finetuned ANN weights are directly transferred to SNN. Channel-wise thresholds of CS-QCFS are used as SNN spiking threshold, after applying a soft-plus function to ensure positivity of SNN's threshold.

DCGS(Huang et al. 2025). The DCGS method provides a training-free approach to convert pre-trained ANNs into high-precision, low-latency SNNs. Differential coding scheme is implemented where spikes are treated as temporal corrections to the encoded values. For ReLU conversions, optimal firing thresholds for multi-threshold neurons are identified through mathematical derivation to minimize quantization errors between ANN activations and their spiking representations. Various non-linear modules in the network are converted to equivalent counterpart in SNNs.

Appendix B. Cannistraci-Hebb Rule

As illustrated in Figure 2, the local community of a node pair is formed by Common Neighbors on length-3 paths (CNs). Within this structure, links can be categorized to iLCL as internal Local Community Link and eLCL as external Local Community Link. Further, we denote:

$$di_n^* = 1 + \#iLCL_n \tag{A1}$$

$$de_n^* = 1 + \#eLCL_n \tag{A2}$$

where n represents any node in local community, and # means to count total number.

To mitigate the computational complexity associated with path-based rule, we adopt the node-based Cannistraci-Hebb rule. (Zhang et al. 2025).

The Cannistraci-Hebb score between two seed nodes u and v can be calculated via two primary variants:

CH2-L3n.

$$\textbf{CH2-L3n}(u,v) = \sum_{z \in CNs} \frac{di_z^*}{de_z^*} \tag{A3}$$

 doi:10.20944/preprints202602.1162.v1

CH3-L3n.

$$\textbf{CH3-L3n}(u,v) = \sum_{z \in CNs} \frac{1}{de_z^*} \tag{A4}$$

The shared property of CH2 and CH3 is to penalizethe external Local Community Link.

For convolutional layers, CHT-Conv can be generally interpreted by unfolding the convolutional layers and applying the CH3-L3n rule.

Cannistraci-Hebb soft rule (CHTs). While a deterministic top-K selection strategy can be used for link regrowth, it often risks becoming trapped in epitopological local minima(Zhang et al. 2025). To address this, the soft rule (CHTs) is proposed, where the probability $p(u,v)$ of regrowing a link between nodes u and v is sampled based on their normalized CH-score:.

If use soft rule, the probability is sampled from the score:

$$p(u,v) = \frac{\textbf{CH-score}(u,v)}{\sum_{u,v} \textbf{CH-score}(u,v)} \tag{A5}$$

Cannistraci-Hebb Training soft rule with sigmoid density decay (CHTss). CHT and CHTs typically maintains a constant sparsity level throughout the training process. To further enhance performance, CHTss incorporates a gradual pruning strategy where the sparsity s_t increases from an initial value s_i to a target value s_f following a sigmoid decay function over iteration t between t_0 and t_f:

$$s_t = s_i + (s_f - s_i)\left(\frac{1}{1 + e^{-k(t - \frac{t_0 + t_f}{2})}}\right) \tag{A6}$$

Appendix C. Implementation Details

Appendix C.1. ANN Training

Appendix C.1.1. VGG-16 and ResNet-18

Basic configurations.

We train all CNNs using an SGD optimizer with a weight decay of 5×10^{-4}. Detailed hyperparameters are listed in Table A1a,b. We perform a grid search for the learning rate over $\{0.1, 0.01, 0.001\}$ and batch size over $\{32, 64, 128\}$. The best-performing ANN is converted to SNN. Dense models, CHT-trained models and pruning(Depgraph) baseline models share this same basic configurations.

Table A1. Training hyper parameter list of VGG-16 and ResNet-18 and (6 Conv 2 FC) architectures. Grid search is performed on hyper parameters labeled 'search'.

(a) VGG-16

hyper parameter	value
epochs	300
learning rate	search
batch size	search
scheduler	cosine annealing
milestones	/
warmup	5

(b) ResNet-18

hyper parameter	value
epochs	200
learning rate	search
batch size	search
scheduler	step decay
milestones	60,120,160
warmup	/

(c) (6 Conv 2 FC) architecture

hyper parameter	value
epochs	200 / 600
learning rate	search
batch size	search
scheduler	cosine annealing
milestones	/
warmup	5

Sparse Training configurations. We apply CHTss (CH3-L3n) to convolutional layers and CHTs (CH2-L3n) to linear layers. Topology evolution occurs every epoch during the first 75% of training.

For convolutional layers in VGG-16 and ResNet-18, the sparsity increases from 20% to 78% ($k = 1$), while linear layer sparsity is fixed at 99%.

For DepGraph, we align the total epochs with CHT: the first 2/3 of epochs are used for dense pre-training, followed by pruning and fine-tuning. Learning rate during finetuning phase is 0.1 times learning rate in pre-training phase(Fang et al. 2023).

 doi:10.20944/preprints202602.1162.v1

Regarding regularization, we apply dropout to the VGG-16 fully-connected layers: 0.5 for dense and 0.2 for CHT because that sparse models require less regularization (Lei et al. 2023). Dropout is set to 0.0 for ResNet-18.

Appendix C.1.2. ViT-B

Basic configurations. We adopt the standard hyperparameter configurations from (Touvron et al. 2021) for both dense and CHT-trained ViT models.

Sparse Training configurations. We use CHTs with BRF initialization (Zhang et al. 2025). Topology evolution is conducted from epoch 10 to 60% of the total epochs with a frequency of 5 epochs. This conservative strategy balances representation preservation with topological exploration.

Appendix C.2. Comparison Between Direct SNN Sprase Training

SNNs converted by our framework and baseline SNNs share a same architecture with 6 convolutional layers and 2 fully connected layers (6 Conv 2 FC).

For DPAP and SD-SNN, we utilize the hyperparameters provided by articles (Han et al. 2024, 2025) to directly train SNNs. These baseline SNNs require 600 epochs and they are trained with $T = 8$ time steps.

To ensure a fair comparison with baselines, we align the total training epochs of SNNs converted by our framework with baseline SNNs. Specifically, models using the CS-QCFS conversion (which requires 400 additional epochs) are trained for 200 ANN epochs, while models using the training-free DCGS method are trained for 600 ANN epochs.

In the experiments summarized in Table 2, we separately align the convolutional and linear layer sparsity of our models with those of the SNN baselines. Given that CHT enforces uniform sparsity within each layer type, we calculate specific sparsity levels for both convolutional and linear components to ensure their respective total parameter counts match those of the baselines.

Table A2 presents the detailed parameter sparsity of convolutional layers and linear layers.

Table A2. Sparsity for convolutional layers and linear layers.

Dataset	method	parameter sparsity(%)	
		all convolutional layers	all linear layers
CIFAR10	DPAP	3.35	72.74
	SD-SNN	6.33	43.15
CIFAR100	DPAP	1.56	73.02
	SD-SNN	6.34	45.37

Other hyper parameters of our ANN training are listed in Table A1c. Except specified sparsity, the topology evolution configurations are the same with configurations described in section C.1.1.

Appendix C.3. CS-QCFS Conversion

CS-QCFS conversion method contains a training stage and an additional finetuning stage for channel-wise threhshold. We provide the hyper parameters in Table A3. We perform a grid search over learning rate in $\{0.05, 0.01, 0.005, 0.001\}$, batch size in $\{32, 64, 128\}$, and L in $\{4, 8\}$, as suggested in (Yang et al. 2025). We report the SNN with best accuracy at $T = 16$ time steps. Reason for this is explained in E.1.

 doi:10.20944/preprints202602.1162.v1

Table A3. Hyper parameter list of CS-QCFS conversion method.

hyper parameter	value
train epochs	300
finetune epochs	100
learning rate	search
batch size	search
L	search
scheduler	cosine annealing
milestones	/
warmup	0
dropout	0

Appendix C.4. DCGS Conversion

In DCGS method, let $2n$ denotes the number of thresholds of multi-threshold neuron.

For CNNs, we use $n = 4$ since it yields better performance (Huang et al. 2025). Threshold iteration is applied to determine optimal threshold.

For challenging ViT, we use $n = 8$. Threshold is set to top 99.9% of activation values of ANN and scaled by $c = 4$ (Huang et al. 2025).

Appendix D. Network Science Analysis Method

Appendix D.1. Scale-Free Network

A scale-free network, as originally defined by Barabási et al.(Barabási & Albert 1999), is a complex network characterized by a highly heterogeneous node degree distribution. In such networks, a small number of nodes (known as *hubs*) possess a disproportionately large number of connections, while the vast majority of nodes have only a few links. A defining property of scale-free networks is that their node degree distribution follows a power law: $P(k) \sim k^{-\gamma}$ where γ is a scaling exponent typically ranging between 2 and 3. This stands in stark contrast to random networks, whose node degree distributions follow a binomial or Poisson distribution (Barabási 2013).

Appendix D.2. Coalescent Embedding

Coalescent embedding, a framework introduced by Muscoloni et al. (Muscoloni et al. 2017), is a topology-aware machine learning techniques designed for unsupervised, nonlinear dimensionality reduction. Its core purpose is to embed complex networks into a geometric space, with a particular focus on hyperbolic geometry, which has proven to be a natural fit for representing many real-world networks. In this work, we employ coalescent embedding to map networks with an underlying latent hyperbolic geometry into a two-dimensional hyperbolic space.

Appendix E. Supplementary Information for Experiments

Appendix E.1. SNN Accuracy Saturation Analysis

While SNN accuracy generally improves with T, it eventually saturates; conversely, energy consumption grows linearly with T. Identifying the saturation point T_0 is thus critical for optimizing the accuracy-efficiency trade-off.

The determination of T_0 is formally described in Algorithm 1.

Using window size $w = 5$ and threshold $\epsilon = 0.01$, we report the saturation times in Table A4a for CNNs and Table A4b for ViT-B.

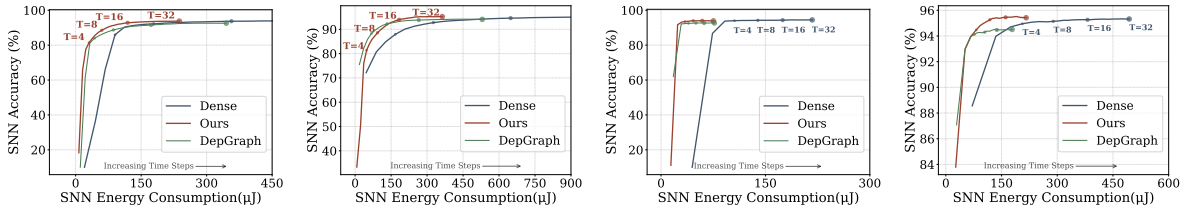
These results provide the rationale for the inference steps reported in the main text: Setting $T = 8$ for DCGS and $T = 16$ for CS-QCFS ensures convergence across all evaluated models.

 doi:10.20944/preprints202602.1162.v1

Algorithm 1: Determination of Saturation Time T_0

Input: Time steps **T**, Accuracies **A**, window size $w = 5$, threshold $\epsilon = 0.01$
Output: Saturation time T_0 or None
Calculate relative improvements: $\Delta_t = (A_t - A_{t-1})/A_{t-1}$
for $i = w$ **to** length(**T**) **do**
  {Check if max improvement in the past w steps is below threshold}
  if $\max(\Delta_{i-w+1}, \ldots, \Delta_i) \leq \epsilon$ **then**
    return $\mathbf{T}[i]$
  end if
end for
return None

Appendix E.2. Pareto Frontier Figures for VGG-16 and ResNet-18 on CIFAR10



(a) VGG-16 via CS-QCFS **(b)** ResNet-18 via CS-QCFS **(c)** VGG-16 via DCGS **(d)** ResNet-18 via DCGS

Figure A1. Pareto Frontiers of Accuracy vs. Energy on CIFAR10. We plot the energy-accuracy trajectory across time steps $T \in [1, 32]$. Each figure contains trajectories of CHT-sparse SNN, dense counterpart and static pruning baseline. *Note: The curves of Dense SNNs converted via CS-QCFS are truncated due to excessive energy consumption.*

Table A4. Saturation time T_0 for SNN accuracy.

(a) VGG-16 and ResNet-18 on CIFAR10 and CIFAR100 via CS-QCFS and DCGS.

Dataset	Model	Method	DCGS T_0	CS-QCFS T_0
CIFAR10	VGG-16	Dense	7	10
		DepGraph	6	11
		Ours	7	13
	ResNet-18	Dense	6	10
		DepGraph	6	10
		Ours	7	13
CIFAR10	VGG-16	Dense	7	16
		DepGraph	7	16
		Ours	7	14
	ResNet-18	Dense	8	12
		DepGraph	7	12
		Ours	8	14

(b) ViT-B on ImageNet-1K via DCGS.

Method	T_0
Dense	4
Ours	4
SparseGPT	5
RIA_{rec}	5
$\text{Wanda}_{\text{rec}}$	5
RIA	6
Wanda	6
RI	7
Magnitude	8

 doi:10.20944/preprints202602.1162.v1

References

Adrian, E. D. The impulses produced by sensory nerve endings: Part i. *The Journal of physiology*, 61(1):49, 1926.

Allard, A. and Serrano, M. Á. Navigable maps of structural brain networks across species. *PLoS computational biology*, 16(2):e1007584, 2020.

Barabási, A.-L. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Bu, T., Fang, W., Ding, J., Dai, P., Yu, Z., and Huang, T. Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks. *arXiv preprint arXiv:2303.04347*, 2023.

Cacciola, A., Muscoloni, A., Narula, V., Calamuneri, A., Nigro, S., Mayer, E. A., Labus, J. S., Anastasi, G., Quattrone, A., Quartarone, A., et al. Coalescent embedding in the hyperbolic space unsupervisedly discloses the hidden geometry of the brain. *arXiv preprint arXiv:1705.04192*, 2017.

Cacciola, A., Naro, A., Milardi, D., Bramanti, A., Malatacca, L., Spitaleri, M., Leo, A., Muscoloni, A., Cannistraci, C. V., Bramanti, P., et al. Functional brain network topology discriminates between patients with minimally conscious state and unresponsive wakefulness syndrome. *Journal of clinical medicine*, 8(3):306, 2019.

Cannistraci, C. V. Modelling self-organization in complex networks via a brain-inspired network automata theory improves link reliability in protein interactomes. *Scientific reports*, 8(1):15760, 2018.

Cannistraci, C. V., Alanis-Lobato, G., and Ravasi, T. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Scientific reports*, 3(1):1613, 2013.

Cheng, H., Zhang, M., and Shi, J. Q. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99, 2018.

DeBole, M. V., Taba, B., Amir, A., Akopyan, F., Andreopoulos, A., Risk, W. P., Kusnitz, J., Otero, C. O., Nayak, T. K., Appuswamy, R., et al. Truenorth: Accelerating from zero to 64 million neurons in 10 years. *Computer*, 52(5): 20–29, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Eshraghian, J. K., Ward, M., Neftci, E. O., Wang, X., Lenz, G., Dwivedi, G., Bennamoun, M., Jeong, D. S., and Lu, W. D. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9): 1016–1054, 2023.

Evci, U., Gale, T., Menick, J., Castro, P. S., and Elsen, E. Rigging the lottery: Making all tickets winners. In *International conference on machine learning*, pp. 2943–2952. PMLR, 2020.

Fang, G., Ma, X., Song, M., Mi, M. B., and Wang, X. Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16091–16101, 2023.

Frantar, E. and Alistarh, D. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pp. 10323–10337. PMLR, 2023.

Han, B., Zhao, F., Zeng, Y., and Shen, G. Developmental plasticity-inspired adaptive pruning for deep spiking and artificial neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Han, B., Zhao, F., Pan, W., and Zeng, Y. Adaptive sparse structure development with pruning and regeneration for spiking neural networks. *Information Sciences*, 689:121481, 2025.

Hanming, L., Wang, Y., Zhang, Y., and Cannistraci, C. V. Cannistraci-hebb training of convolutional neural networks. In *NeurIPS 2025 Workshop on Symmetry and Geometry in Neural Representations*, 2025.

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

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hebb, D. O. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

Hu, Y., Zheng, Q., Jiang, X., and Pan, G. Fast-snn: Fast spiking neural network by converting quantized ann. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14546–14562, 2023.

Huang, Z., Fang, W., Bu, T., Xue, P., Hao, Z., Liu, W., Tang, Y., Yu, Z., and Huang, T. Differential coding for training-free ann-to-snn conversion. *arXiv preprint arXiv:2503.00301*, 2025.

Huo, D., Zhang, J., Dai, X., Zhang, J., Qian, C., Tang, K.-T., and Chen, H. Anp-g: A 28nm 1.04 pj/sop sub-mm2 spiking and back-propagation hybrid neural network asynchronous olfactory processor enabling few-shot class-incremental on-chip learning. In *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, pp. 1–2. IEEE, 2023.

Jayakumar, S., Pascanu, R., Rae, J., Osindero, S., and Elsen, E. Top-kast: Top-k always sparse training. *Advances in Neural Information Processing Systems*, 33:20744–20754, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lei, B., Zhang, R., Xu, D., and Mallick, B. Calibrating the rigged lottery: Making all tickets reliable. *arXiv preprint arXiv:2302.09369*, 2023.

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

Li, Y., Deng, S., Dong, X., and Gu, S. Error-aware conversion from ann to snn via post-training parameter calibration. *International Journal of Computer Vision*, 132(9):3586–3609, 2024.

Liu, S. and Deng, W. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*, pp. 730–734. IEEE, 2015.

Lynn, C. W., Holmes, C. M., and Palmer, S. E. Heavy-tailed neuronal connectivity arises from hebbian self-organization. *Nature Physics*, 20(3):484–491, 2024.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

Mocanu, D. C., Mocanu, E., Stone, P., Nguyen, P. H., Gibescu, M., and Liotta, A. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1): 2383, 2018.

Modaresi, F., Guthaus, M., and Eshraghian, J. K. Openspike: an openram snn accelerator. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. IEEE, 2023.

Muscoloni, A., Thomas, J. M., Ciucci, S., Bianconi, G., and Cannistraci, C. V. Machine learning meets complex networks via coalescent embedding in the hyperbolic space. *Nature communications*, 8(1):1615, 2017.

Narula, V., Zippo, A. G., Muscoloni, A., Biella, G. E. M., and Cannistraci, C. V. Can local-community-paradigm and epitopological learning enhance our understanding of how local brain connectivity is able to process, learn and memorize chronic pain? *Applied Network Science*, 2(1):28, 2017.

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

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Tmamna, J., Ayed, E. B., Fourati, R., Gogate, M., Arslan, T., Hussain, A., and Ayed, M. B. Pruning deep neural networks for green energy-efficient models: A survey. *Cognitive Computation*, 16(6):2931–2952, 2024.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Wang, Y., Zhang, M., Chen, Y., and Qu, H. Signed neuron with memory: Towards simple, accurate and high-efficient ann-snn conversion. In *IJCAI*, pp. 2501–2508, 2022.

Wang, Z., Lil, S., Ma, Z., and Yao, Q. High accurate, low latency conversion of spiking neural networks with blif neurons. In *2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, pp. 432–440. IEEE, 2024.

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

Yang, H., Yang, S., Zhang, L., Dou, H., Shen, F., and Zhao, J. Cs-qcfs: Bridging the performance gap in ultra-low latency spiking neural networks. *Neural Networks*, 184:107076, 2025.

Yao, M., Hu, J., Zhao, G., Wang, Y., Zhang, Z., Xu, B., and Li, G. Inherent redundancy in spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16924–16934, 2023.

Yao, M., Qiu, X., Hu, T., Hu, J., Chou, Y., Tian, K., Liao, J., Leng, L., Xu, B., and Li, G. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

You, K., Xu, Z., Nie, C., Deng, Z., Guo, Q., Wang, X., and He, Z. Spikezip-tf: Conversion is all you need for transformer-based snn. *arXiv preprint arXiv:2406.03470*, 2024.

 doi:10.20944/preprints202602.1162.v1

Yuan, G., Ma, X., Niu, W., Li, Z., Kong, Z., Liu, N., Gong, Y., Zhan, Z., He, C., Jin, Q., et al. Mest: Accurate and fast memory-economic sparse training framework on the edge. *Advances in Neural Information Processing Systems*, 34:20838–20850, 2021.

Zhang, J., Huo, D., Zhang, J., Qian, C., Liu, Q., Pan, L., Wang, Z., Qiao, N., Tang, K.-T., and Chen, H. Anp-i: a 28-nm 1.5-pj/sop asynchronous spiking neural network processor enabling sub-0.1-μ j/sample on-chip learning for edge-ai applications. *IEEE Journal of Solid-State Circuits*, 59(8):2717–2729, 2024a.

Zhang, Y., Bai, H., Lin, H., Zhao, J., Hou, L., and Cannistraci, C. V. Plug-and-play: An efficient post-training pruning method for large language models. 2024b.

Zhang, Y., Zhao, J., Wu, W., and Muscoloni, A. Epitopological learning and cannistraci-hebb network shape intelligence brain-inspired theory for ultra-sparse advantage in deep learning. In *The Twelfth International Conference on Learning Representations*, 2024c.

Zhang, Y., Cerretti, D., Zhao, J., Wu, W., Liao, Z., Michieli, U., and Cannistraci, C. V. Brain network science modelling of sparse neural networks enables transformers and LLMs to perform as fully connected. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=OM0Qkq9xtY.

Zhao, J., Muscoloni, A., Michieli, U., Zhang, Y., and Cannistraci, C. V. Adaptive cannistraci-hebb network automata modelling of complex networks for path-based link prediction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Zhu, M. and Gupta, S. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.