CLAG-SNN: COMPACT LOCAL, ABSTRACT GLOBAL FOR EFFICIENT SPIKING 3D POINT CLOUD RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) offer prominent advantages in efficient 3D point-cloud recognition. However, the discrete nature of spiking representations and the irregular structure of point clouds make it difficult to simultaneously preserve local geometry and form high-level semantic abstractions. Inspired by the hierarchical processing mechanism of biological vision, we propose a **Compact Local, Abstract Global (CLAG)** principle, which preserves nonlinear dependencies before and after compression at the local scale to maintain structure, and suppresses nonlinear dependencies between inputs and spiking features at the global scale to promote abstraction. We instantiate this principle as a hierarchical framework termed CLAG-SNN. First, at the local scale, we propose **Hierarchical Structure-aware Token Compaction (HSTC)**, which preserves local geometric dependencies through structure-aware token aggregation and compaction, achieving local compactness. Furthermore, at the global scale, we propose **Nonlinearity-Aware Abstraction Bottleneck (NAAB)**, which suppresses redundant input–feature nonlinear coupling and abstracts compact tokens into discriminative global semantics. Experiments on multiple 3D point-cloud recognition benchmarks show that CLAG-SNN achieves consistent improvements over the baseline: while improving accuracy, it increases inference speed by up to 24.4% and reduces theoretical energy consumption by up to one-third; with only 1% training data, it still achieves a 3.32% performance gain.

1 INTRODUCTION

Spiking Neural Networks (SNNs), which emulate the event-driven computational mechanism of biological neurons, have recently emerged as an important paradigm for efficient 3D point cloud recognition Ren et al. (2023); Lan et al. (2023); Qiu et al. (2025). By encoding and transmitting information through sparse spikes and temporal dynamics, SNNs can exploit event-driven computation to reduce redundant neural activity Yang & Chen (2023); Yang et al. (2024), showing considerable potential for low-power computing and deployment on neuromorphic hardware Yao et al. (2023); Yang et al. (2026). This computational property is naturally aligned with the sparsity and irregular spatial distribution of 3D point clouds Chang & Wang (2025); Zeng et al. (2024), which has motivated growing interest in SNN-based 3D point cloud recognition. However, unlike the continuous feature representations used in conventional artificial neural networks, spike-based representations are inherently discrete, sparse, and temporally dynamic. While these characteristics bring substantial advantages in computational efficiency, they also introduce new challenges for effectively preserving and modeling 3D geometric information.

More specifically, effective 3D recognition requires SNNs to balance local geometric preservation and high-level semantic abstraction. Fine-grained geometric patterns, including spatial distributions, neighborhood relations, and curvature, provide essential recognition cues, motivating hierarchical point cloud networks to progressively transform local structures into semantic representations Qi et al. (2017b). However, spike-based representations often suffer from information loss and limited capability in modeling complex 3D spatial relationships Ren et al. (2023); Lu et al. (2025). Meanwhile, progressive token reduction in hierarchical architectures further introduces a fundamental trade-off between computational efficiency and structural preservation. This observation suggests that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

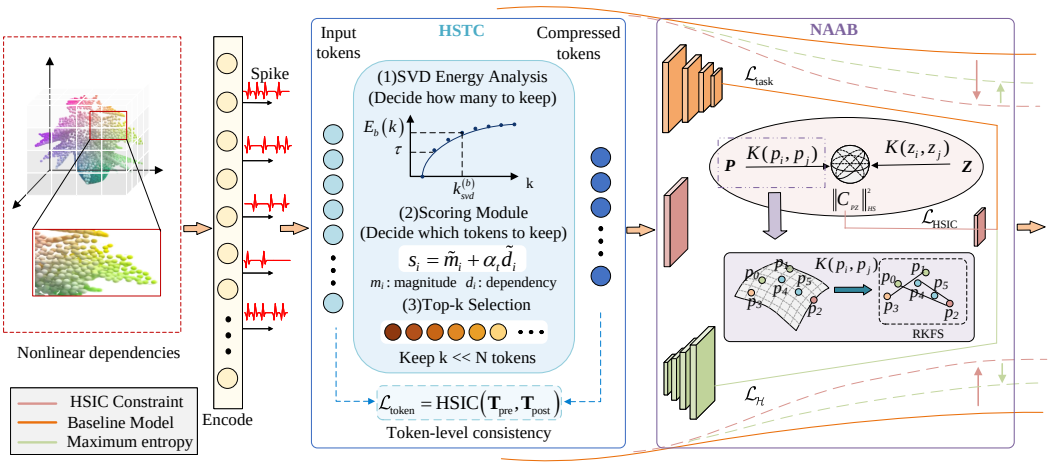

Figure 1: **Illustration of CLAG-SNN.** HSTC first compacts sparse spiking tokens through SVD-based structure estimation and HSIC-guided token consistency, preserving informative local geometry under adaptive token reduction. The compact representations are then processed by NAAB, which suppresses nonlinear dependence on input-specific details to encourage global semantic abstraction, while entropy maximization preserves latent diversity and prevents representation collapse. Colored trajectories denote representation evolution under different strategies.

efficient spiking 3D recognition requires hierarchy-aware information regulation rather than uniform compression or preservation.

We argue that information preservation should not be treated as a universal strategy throughout the representation hierarchy. At local scales, compressed representations should maintain strong dependency with their original structures to prevent geometric degradation caused by discretization and token reduction. In contrast, as the receptive field expands and semantic levels increase, high-level representations should gradually reduce excessive dependency on input-specific variations, such as sampling density, and evolve toward category-level semantic representations. Therefore, the key challenge is not whether information should be preserved or discarded, but how the nonlinear dependency between inputs and learned representations should be regulated according to different hierarchical scales.

Interestingly, biological vision systems naturally exhibit such hierarchy-dependent information regulation. Early visual areas, such as V1 and V2, preserve local structures including edges, contours, and spatial configurations Hubel & Wiesel (1962); Hegdé & Van Essen (2003), whereas higher visual regions, particularly the inferotemporal (IT) cortex, progressively integrate these responses into abstract and viewpoint-invariant object representations Ratan Murty & Arun (2015); DiCarlo et al. (2012). This hierarchical organization suggests that early representations should selectively preserve informative local structures, while higher representations should gradually discard input-specific details and develop semantic abstractions.

Mirroring this hierarchical organization, we propose the **Compact Local, Abstract Global (CLAG)** representation principle for spiking 3D recognition and develop a hierarchical framework termed **CLAG-SNN**. The key insight of CLAG is that nonlinear dependency, measured by the Hilbert-Schmidt Independence Criterion (HSIC), should not be uniformly maximized or minimized across the network. Instead, its optimal regulation direction depends on the representation hierarchy: local scales should maintain strong dependency between pre- and post-compaction representations to preserve structural information, whereas global scales should suppress excessive dependency between input descriptors and latent representations to promote semantic abstraction.

Specifically, at the **Compact Local** scale, we introduce the **Hierarchical Structure-aware Token Compaction (HSTC)** mechanism. HSTC estimates the intrinsic structural complexity of token features through spectral analysis and effective rank, enabling adaptive token allocation. Subsequently, structure-aware token selection and HSIC-based dependency preservation are employed to maintain local nonlinear structures during token compaction.

At the **Abstract Global** scale, we propose the **Nonlinearity-Aware Abstraction Bottleneck (NAAB)**, which acts on the last-layer latent representations. NAAB suppresses nonlinear dependency between input-level global descriptors and latent representations through an HSIC constraint, reducing excessive reliance on input-specific details and encouraging task-relevant semantic abstraction. Meanwhile, entropy maximization is incorporated to preserve latent diversity and prevent representation collapse during dependency suppression.

Extensive experiments are conducted on ModelNet10, ModelNet40, KITTI, SemanticKITTI, and ScanObjectNN. The results demonstrate that CLAG-SNN improves both recognition accuracy and computational efficiency compared with existing spiking models. Beyond accuracy improvement, CLAG-SNN achieves up to a 24.4% inference speed improvement and maintains strong performance under limited-data training conditions, demonstrating the effectiveness of hierarchy-dependent nonlinear dependency regulation.

Our main contributions are summarized as follows:

1. We propose the **Compact Local, Abstract Global (CLAG)** hierarchical representation principle for spiking 3D recognition, revealing that nonlinear dependency should be regulated in opposite directions at different representation scales to balance local structure preservation and semantic abstraction.

2. We introduce **HSTC**, a structure-aware token compaction mechanism that combines spectral analysis, adaptive token allocation, and HSIC-based dependency preservation to achieve efficient and structure-preserving local representation learning.

3. We develop **NAAB**, a nonlinearity-aware abstraction mechanism that promotes high-level semantic abstraction by suppressing input–representation dependency while maintaining latent diversity through entropy maximization.

4. Extensive experiments on multiple benchmark datasets demonstrate the effectiveness and generalization capability of CLAG-SNN, particularly under limited-data training scenarios.

2 Preliminary

2.1 Hilbert-Schmidt Independence Criterion (HSIC)

All dependency statements in this paper are made through one measure. The Hilbert–Schmidt Independence Criterion Gretton et al. (2005) embeds two random vectors $\mathbf{X}$ and $\mathbf{Z}$ into a reproducing-kernel Hilbert space (RKHS) via the feature maps ϕ_X and ϕ_Z induced by positive-definite kernels k_X and k_Z, and scores their dependency by the squared Hilbert–Schmidt norm of the empirical cross-covariance operator:

$$\mathrm{HSIC}(\mathbf{X}, \mathbf{Z}) = \left\| \widehat{C}_{\phi_X(\mathbf{X}), \phi_Z(\mathbf{Z})} \right\|_{\mathrm{HS}}^2 = \frac{1}{(B-1)^2} \operatorname{tr}(H_B K_X H_B K_Z)\,, \tag{1}$$

where $K_X(b, b') = k_X(\mathbf{x}_b, \mathbf{x}_{b'})$ and $K_Z(b, b') = k_Z(\mathbf{z}_b, \mathbf{z}_{b'})$ are Gram matrices over a mini-batch of B samples, and $H_B = I_B - B^{-1}\mathbf{1}\mathbf{1}^\top$ centers them. Three properties make HSIC the right substrate for our design. First, it detects nonlinear dependency: for characteristic kernels such as the Gaussian kernels used throughout this paper, the population HSIC vanishes if and only if the two variables are independent. Second, the estimator in equation 1 is non-negative, bounded, and differentiable with respect to its arguments, so it can be optimized end-to-end by backpropagation. Third, it requires no density estimation and costs only two Gram matrices per application, which is negligible relative to the network forward pass.

3 Compact Local, Abstract Global

3.1 CLAG Principle and Problem Definition

Let $P = \{p_n\}_{n=1}^N$ be an input point cloud, where $p_n = (\mathbf{c}_n, \mathbf{f}_n)$ collects the coordinate $\mathbf{c}_n \in \mathbb{R}^3$ and the attribute feature $\mathbf{f}_n \in \mathbb{R}^{d_f}$. A hierarchical SNN backbone processes P layer by layer. A spiking unit integrates its input current and fires through the LIF dynamics

$$U^{(l)}[t] = H^{(l)}[t-1] + I^{(l)}[t], \qquad S^{(l)}[t] = \Theta\big(U^{(l)}[t] - V_{\mathrm{th}}\big), \tag{2}$$

where $I^{(l)}[t]$ is the input current to layer l at timestep t, $H^{(l)}[t]$ the membrane potential, V_{th} the firing threshold, and $\Theta(\cdot)$ the Heaviside step; after a spike the membrane potential is reset and the integration restarts. The integer LIF (I-LIF) neuron Luo et al. (2024) replaces the binary threshold-and-reset by an integer quantization of the accumulated potential: the continuous pre-spiking activation $\mathbf{A}^{(l)}$ is converted into integer spikes $\mathbf{S}^{(l)} = \text{Clip}(\text{round}(\mathbf{A}^{(l)}), 0, D)$, where $\text{round}(\cdot)$ rounds to the nearest integer, $\text{Clip}(\cdot)$ restricts the output to $[0, D]$, and D caps the spike magnitude; retaining integer amplitudes reduces the quantization error of binary spiking. We stack the layer-l responses into a token matrix $\mathbf{X}^{(l)} = \psi_l(\mathbf{S}^{(l)}) \in \mathbb{R}^{N_l \times d_l}$, where ψ_l is an identity map or a lightweight projection. For sample b, $\mathcal{I}_b \subseteq \{1, \dots, N_l\}$ collects the active token indices, i.e., the spatial sites that carry nonzero spiking activity at layer l, with $N_b = |\mathcal{I}_b|$, and $\mathbf{x}_i$ denotes the i-th token. As l deepens, $\mathbf{X}^{(l)}$ moves from fine local geometry toward high-level semantics, which is exactly the span across which CLAG operates.

Definition 1 (Descriptor dependency)**.** For two descriptor stacks over the same mini-batch, $\mathbf{A} = [\mathbf{a}_1; \dots; \mathbf{a}_B]$ and $\mathbf{A}' = [\mathbf{a}'_1; \dots; \mathbf{a}'_B]$, we write $\text{Dep}(\mathbf{A}, \mathbf{A}')$ for the empirical HSIC between them, computed with Gaussian kernels whose bandwidths follow the median heuristic.

The CLAG principle now reads as follows. At local scales, where tokens are about to be compacted, the nonlinear dependency between the pre- and post-compaction descriptors should be maximized; at the global scale, the dependency between the input descriptor and the latent descriptor should be minimized, so that category-level semantics replace input-specific detail:

$$\max \text{Dep}(\mathbf{T}_{\text{pre}}, \mathbf{T}_{\text{post}}) \quad \text{(Compact Local)}, \qquad \min \text{Dep}(\mathbf{X}^{\text{in}}, \mathbf{Z}) \quad \text{(Abstract Global)}. \tag{3}$$

HSTC and NAAB realize the two directions, respectively.

Which tokens count as local and which as global is fixed by the hierarchy itself: the local direction applies at the merging sites, where detail is consolidated and hardest to recover, and the global direction acts at the last layer, whose tokens already serve the prediction head; residual input dependency there would preserve sampling-specific detail at the expense of category-level structure. The two scales are thus the two roles that any hierarchical spiking network must play, and CLAG makes each role explicit and measurable.

3.2 Compact Local: Hierarchical Structure-aware Token Compaction

HSTC compacts tokens adaptively according to the structure they carry. We omit the layer superscript in this subsection and write $\mathbf{X}^{(b)} \in \mathbb{R}^{N_b \times d_l}$ for the token matrix of sample b.

The token budget derives from spectral structure. Centering the token matrix and taking its singular value decomposition,

$$\bar{\mathbf{X}}^{(b)} = \big(I_{N_b} - \tfrac{1}{N_b}\mathbf{1}\mathbf{1}^\top\big)\mathbf{X}^{(b)} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \qquad \sigma_1^{(b)} \geq \cdots \geq \sigma_{r_b}^{(b)}, \;\; r_b = \min(N_b, d_l), \tag{4}$$

define the cumulative spectral energy $E_b(k) = \dfrac{\sum_{j=1}^{k}\big(\sigma_j^{(b)}\big)^2}{\sum_{j=1}^{r_b}\big(\sigma_j^{(b)}\big)^2}$, which rises from 0 to 1.

Definition 2 (Intrinsic token rank)**.** For a threshold $\tau \in (0, 1)$, the intrinsic token rank of sample b is $k_{\text{svd}}^{(b)} = \min\{k \in \{1, \dots, r_b\} : E_b(k) \geq \tau\}$.

Since E_b is non-decreasing with $E_b(r_b) = 1$, the minimum in Definition 2 is well defined. A spectrum spread over many directions yields a large rank and hence a large token budget, while a low-rank spectrum yields a small one. The budget is clipped for stability, $k^{(b)} = \Pi_{[N_{\min}, N_{\max}]}\big(\max\{k_{\text{svd}}^{(b)}, \lfloor \rho_{\min} N_b \rfloor\}\big)$, where $\rho_{\min}$ is a minimum keep ratio and $\Pi_{[a,b]}(\cdot)$ clips its argument into $[a, b]$.

Given the budget, tokens are ranked by a saliency score. The sample descriptor is $\mathbf{g}^{(b)} = \frac{1}{N_b}\sum_{i \in \mathcal{I}_b} \mathbf{x}_i$. For token $i \in \mathcal{I}_b$, the magnitude score $m_i = \frac{1}{d_l}\|\mathbf{x}_i\|_1$ measures activation strength, and the dependency-aware score $d_i = \frac{1}{d_l}\big|(\mathbf{x}_i - \mathbf{g}^{(b)})^\top \mathbf{g}^{(b)}\big|$ measures the contribution of token i along the sample direction. After per-sample standardization, the saliency is

$$s_i = \tilde{m}_i + \alpha_t \tilde{d}_i = \frac{m_i - \mu_m^{(b)}}{\sigma_m^{(b)} + \varepsilon} + \alpha_t \, \frac{d_i - \mu_d^{(b)}}{\sigma_d^{(b)} + \varepsilon}, \qquad i \in \mathcal{I}_b, \tag{5}$$

with $\alpha_t > 0$, where $\mu_m^{(b)}, \sigma_m^{(b)}$ and $\mu_d^{(b)}, \sigma_d^{(b)}$ are the per-sample means and standard deviations of the two scores, computed over the active tokens $\mathcal{I}_b$ and ε guards numerical stability, and HSTC retains $\mathcal{I}_b^{\text{keep}} = \text{TopK}(\{s_i\}_{i\in\mathcal{I}_b}, k^{(b)})$. The pre- and post-compaction descriptors are $\mathbf{t}_{\text{pre}}^{(b)} = \frac{1}{N_b}\sum_{i\in\mathcal{I}_b}\mathbf{x}_i$ and $\mathbf{t}_{\text{post}}^{(b)} = \frac{1}{k^{(b)}}\sum_{i\in\mathcal{I}_b^{\text{keep}}}\mathbf{x}_i$, and the average keep ratio over a batch is $r_{\text{keep}} = \frac{1}{B}\sum_{b=1}^{B} k^{(b)}/N_b$.

Selection alone, however, offers no guarantee on what compaction destroys, so we first isolate the descriptor-level price of dropping tokens.

Proposition 1 (Descriptor distortion of token compaction)**.** *Let $\mathbf{t}_{\text{disc}}$ be the mean of the discarded tokens when $k^{(b)} < N_b$, $r_{\text{keep}}^{(b)} = k^{(b)}/N_b$, and $R_b = \max_{i\in\mathcal{I}_b}\|\mathbf{x}_i - \mathbf{t}_{\text{pre}}^{(b)}\|_2$. Then*

$$\left\|\mathbf{t}_{\text{post}}^{(b)} - \mathbf{t}_{\text{pre}}^{(b)}\right\|_2 = \left(1 - r_{\text{keep}}^{(b)}\right)\left\|\mathbf{t}_{\text{post}}^{(b)} - \mathbf{t}_{\text{disc}}\right\|_2 \le 2\left(1 - r_{\text{keep}}^{(b)}\right)R_b. \tag{6}$$

The identity follows from $\mathbf{t}_{\text{pre}}^{(b)} = r_{\text{keep}}^{(b)}\mathbf{t}_{\text{post}}^{(b)} + (1 - r_{\text{keep}}^{(b)})\mathbf{t}_{\text{disc}}$, and the inequality from the triangle inequality, since both $\mathbf{t}_{\text{post}}^{(b)}$ and $\mathbf{t}_{\text{disc}}$ are convex combinations of tokens that lie within R_b of $\mathbf{t}_{\text{pre}}^{(b)}$. Distortion therefore grows linearly with the discarded mass $1 - r_{\text{keep}}^{(b)}$ and with the dispersion R_b, which are precisely the two quantities that the adaptive budget and the consistency term below control.

Mean descriptors, moreover, ignore the relational structure across samples. HSTC therefore measures consistency at the level of sample relations: stacking $\mathbf{t}_{\text{pre}}^{(b)}$ and $\mathbf{t}_{\text{post}}^{(b)}$ over the batch into $\mathbf{T}_{\text{pre}}, \mathbf{T}_{\text{post}} \in \mathbb{R}^{B\times d_l}$, the Compact Local loss is

$$\mathcal{L}_{\text{HSTC}} = \text{Dep}(\mathbf{T}_{\text{pre}}, \mathbf{T}_{\text{post}}) = \frac{1}{(B-1)^2}\,\text{tr}\big(H_B K_{\text{pre}} H_B K_{\text{post}}\big), \tag{7}$$

where $K_{\text{pre}}(b,b') = \exp\big(-\|\mathbf{t}_{\text{pre}}^{(b)} - \mathbf{t}_{\text{pre}}^{(b')}\|_2^2/2\sigma_{\text{pre}}^2\big)$ and K_{post} is defined analogously on $\mathbf{T}_{\text{post}}$, with bandwidths following the median heuristic of Definition 1. HSTC maximizes equation 7. Maximizing it keeps the joint distribution of sample descriptors aligned across compaction, preserving not only the mean geometry but also the relative structure among samples. The consistency is posed on batch descriptors rather than individual tokens: a per-token reconstruction objective would anchor retained tokens to the redundant ones they replace, whereas equation 7 only asks that the batch-level joint structure survive.

For 3D detection, hard pruning may discard localization-sensitive evidence for small or distant objects represented by only a few sparse tokens. The detection branch therefore keeps the full token set and adds only a training-time constraint $\mathcal{L}_{\text{det}} = \text{Dep}(\mathbf{T}_{\text{det-pre}}, \mathbf{T}_{\text{det-post}})$. The inference path is unchanged, while the backbone still learns structurally compact and dependency-preserving tokens. Figure 2 summarizes the module.

The dominant overhead of HSTC is the SVD-based budget estimate, $O(\min(N_b d_l^2,\, N_b^2 d_l))$ per sample on the active token set; scoring and descriptor aggregation are linear in the active tokens. At inference the reduced budget more than repays this cost: CLAG-SNN attains the highest FPS and the lowest theoretical power among the compared spiking models (Table 1).

3.3 Abstract Global: Nonlinearity-Aware Abstraction Bottleneck

HSTC keeps local structure; the Abstract Global direction asks high-level representations to loosen their grip on input-specific factors. NAAB acts on the last layer L: the latent descriptor of sample b is $\mathbf{z}_b = \frac{1}{N_L}\sum_{i\in\mathcal{I}_b}\mathbf{x}_i^{(L)}$, and the input descriptor is $\mathbf{x}_b^{\text{in}} = \frac{1}{N_0}\sum_{i\in\mathcal{I}_b^{(0)}}\mathbf{u}_i$ with $\mathbf{u}_i = [\mathbf{c}_i, \mathbf{f}_i]$ the raw coordinate–attribute vector, where $\mathcal{I}_b^{(0)}$ indexes the input tokens. Stacking over the batch gives $\mathbf{X}^{\text{in}} \in \mathbb{R}^{B\times d_x}$ and $\mathbf{Z} \in \mathbb{R}^{B\times d_z}$.

The input-to-latent dependency is quantified by the same estimator as equation 7, applied at the global scale:

$$\mathcal{L}_{\text{dep}} = \text{Dep}(\mathbf{X}^{\text{in}}, \mathbf{Z}) = \frac{1}{(B-1)^2}\,\text{tr}\big(H_B K_X H_B K_Z\big), \tag{8}$$

where $K_X(b,b') = \exp\big(-\|\mathbf{x}_b^{\text{in}} - \mathbf{x}_{b'}^{\text{in}}\|_2^2/2\sigma_X^2\big)$ and K_Z is defined analogously on $\mathbf{Z}$, with bandwidths following the median heuristic of Definition 1. By equation 1, $\mathcal{L}_{\text{dep}}$ is the squared RKHS norm of

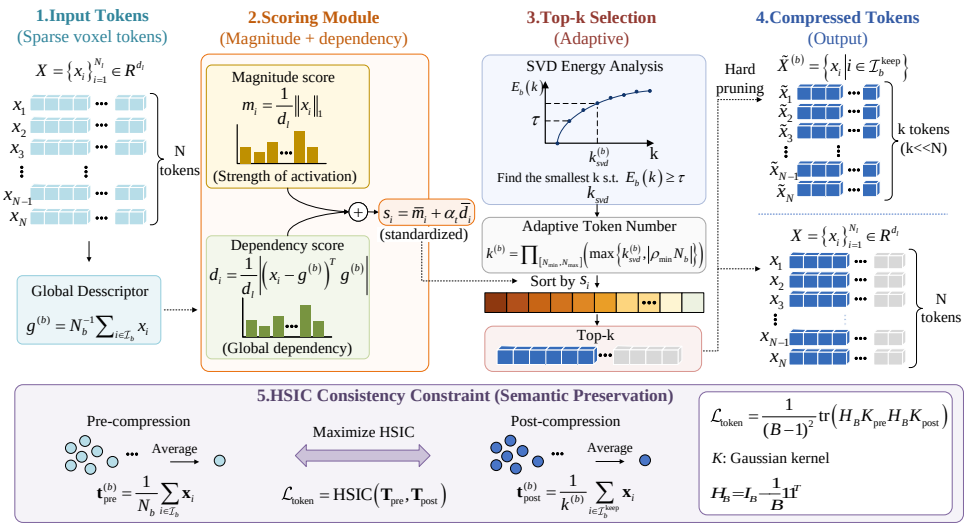

Figure 2: **The HSTC module.** A scoring module evaluates each token by activation magnitude and global dependency; an adaptive Top-k strategy, guided by SVD spectral analysis, determines how many tokens to retain. The selected tokens form the compact representation, and an HSIC-based consistency constraint preserves the nonlinear structure across compaction.

the empirical cross-covariance between the embedded descriptors, so minimizing it removes the statistical dependence between what the raw cloud looks like and what the network encodes, while the task loss decides which semantic factors survive.

Dependency suppression without a counterweight can collapse different samples into a narrow latent region. NAAB therefore couples the bottleneck with a diversity mechanism. Under a diagonal Gaussian approximation of the batch latent distribution, with channel statistics $\mu_c = \frac{1}{B}\sum_{b=1}^{B} z_{b,c}$ and $\sigma_c^2 = \frac{1}{B}\sum_{b=1}^{B}(z_{b,c}-\mu_c)^2 + \varepsilon$, where $\varepsilon > 0$ guards numerical stability, the latent entropy is $H(\mathbf{Z}) = \frac{1}{2}\sum_{c=1}^{d_z}\log\big(2\pi e\,\sigma_c^2\big)$. A kernel-density guard $\mathcal{L}_{\mathrm{KDE}} = -\frac{1}{B}\sum_{b=1}^{B}\log\Big[\frac{1}{B}\sum_{b'=1}^{B}\exp\big(-\|\mathbf{z}_b - \mathbf{z}_{b'}\|_2^2/2\eta^2\big)\Big]$, grows as the latent kernel density becomes diffuse, and thus costs excessive dispersion. The two terms anchor the latent scale from opposite sides, entropy against collapse and $\mathcal{L}_{\mathrm{KDE}}$ against fragmentation, and form one diversity functional $\mathcal{L}_{\mathcal{H}} = H(\mathbf{Z}) - \kappa\mathcal{L}_{\mathrm{KDE}}$ with $\kappa > 0$. NAAB unifies suppression with this anchor in a single functional:

$$\mathcal{L}_{\mathrm{NAAB}} = \mathcal{L}_{\mathrm{dep}} - \beta\,\mathcal{L}_{\mathcal{H}} = \mathcal{L}_{\mathrm{dep}} - \beta\big(H(\mathbf{Z}) - \kappa\,\mathcal{L}_{\mathrm{KDE}}\big), \tag{9}$$

with $\beta > 0$, which NAAB minimizes. The anchor is internal to the bottleneck: entropy keeps the latent space dispersed while the bottleneck squeezes input dependency out, and $\mathcal{L}_{\mathrm{KDE}}$ keeps that dispersion bounded.

3.4 The CLAG Objective and Generalization Bound

CLAG is one constrained program with one budget per scale, not a list of terms. Writing $\mathcal{L}_{\mathrm{task}}$ for the supervised loss of the downstream task, CLAG seeks parameters θ that solve

$$\min_{\theta}\ \mathcal{L}_{\mathrm{task}}(\theta) \quad \text{s.t.} \quad \mathcal{L}_{\mathrm{HSTC}} \geq \varepsilon_{\mathrm{loc}},\ \mathcal{L}_{\mathrm{NAAB}} \leq \varepsilon_{\mathrm{glb}}, \tag{10}$$

that is, the local module must retain structural dependency above $\varepsilon_{\mathrm{loc}}$, and the global module must keep its objective equation 9 below $\varepsilon_{\mathrm{glb}}$; the budget is posed on the anchored functional rather than on the bare dependency, which is what keeps the cap from collapsing the latent space. Each module answers for one budget at its own scale, and no third objective exists. The Lagrangian of equation 10, with prices $\lambda_{\mathrm{HSTC}}, \lambda_{\mathrm{NAAB}} > 0$ and the constants absorbed, is the CLAG-SNN training objective

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{task}} - \lambda_{\mathrm{HSTC}}\,\mathcal{L}_{\mathrm{HSTC}} + \lambda_{\mathrm{NAAB}}\,\mathcal{L}_{\mathrm{NAAB}}. \tag{11}$$

It contains exactly two module terms alongside the task: the local module is rewarded for structural retention, the global module is charged for its residual bottleneck cost, and both charges evaluate the same HSIC measure at their own scale.

Proposition 2 (Dual form of the CLAG objective)**.** *Let $\mathcal{L}_{\text{total}}$ be as in equation 11. (i) There is a constant C, fixed once the prices are, such that $\mathcal{L}_{\text{total}}(\theta) \le \mathcal{L}_{\text{task}}(\theta) + C$ for every θ feasible in equation 10; hence $\min_\theta \mathcal{L}_{\text{total}}(\theta)$ lower-bounds the optimal task loss under the two budgets, and the prices set the rate at which the budgets are traded against the task. (ii) At a saddle point the prices are complementary to the budgets,*

$$\lambda_{\text{HSTC}}\big(\mathcal{L}_{\text{HSTC}} - \varepsilon_{\text{loc}}\big) = 0, \qquad \lambda_{\text{NAAB}}\big(\varepsilon_{\text{glb}} - \mathcal{L}_{\text{NAAB}}\big) = 0, \tag{12}$$

so a price is positive only while its budget binds.

This dual view is what makes equation 11 one objective rather than a stack of terms. In the symmetric case $\lambda_{\text{HSTC}} = \lambda_{\text{NAAB}} = \lambda$, the objective reduces to

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda\big(\mathcal{L}_{\text{NAAB}} - \mathcal{L}_{\text{HSTC}}\big), \tag{13}$$

a single contrast between what the global scale must shed and what the local scale must keep. Moreover, by equation 12 a positive price coincides with an active budget, so at the optimum the training pressure concentrates on the binding constraint instead of being spread uniformly over the network. Training minimizes equation 11 by gradient descent through the surrogate gradients of the spiking units; the full procedure is given in Appendix A.

These budgets are not heuristic choices but the levers that control generalization.

Theorem 1 (Generalization bound for CLAG)**.** *Let $h = g \circ f_\theta$, where f_θ is the spiking backbone and g the prediction head, and suppose the assumptions in Appendix B hold. Then, with probability at least $1 - \delta$ over an i.i.d. training set of size n,*

$$\begin{aligned} R(h) - \widehat{R}_S(h) \le & L_\ell L_g \Big[c_1 \sqrt{\mathcal{L}_{\text{dep}}} + \frac{c_2}{\sqrt{H(\mathbf{Z}) + \varepsilon_0}} + c_3\, \mathcal{L}_{\text{KDE}} + \frac{c_4}{\sqrt{\mathcal{L}_{\text{HSTC}} + \varepsilon_t}} + c_5 (1 - r_{\text{keep}}) \Big] \\ & + c_6 \sqrt{\frac{d_{\text{eff}}}{n}} + \sqrt{\frac{\log(2/\delta)}{2n}}, \end{aligned} \tag{14}$$

where d_{eff} is the intrinsic dimension of the data manifold and $\varepsilon_0, \varepsilon_t > 0$ are small constants.

Every term in equation 14 is moved in the favorable direction by CLAG: minimizing $\mathcal{L}_{\text{dep}}$ removes the first term, raising the latent entropy $H(\mathbf{Z})$ shrinks the second, suppressing latent dispersion through $\mathcal{L}_{\text{KDE}}$ controls the third, maximizing the structural consistency $\mathcal{L}_{\text{HSTC}}$ shrinks the fourth, and the adaptive budget of HSTC manages the fifth, the intrinsic price of token economy from Proposition 1. The two opposite dependency directions of CLAG thus serve a single goal (proof in Appendix B).

4 Experiment

Following the experimental protocol of Qiu et al. (2025), we conduct evaluations on the ModelNet10, ModelNet40 Wu et al. (2015), ScanObjectNN Uy et al. (2019), KITTI Geiger et al. (2012), and Semantic KITTI Behley et al. (2019) datasets. In addition to the main comparison experiments on classification, detection, and segmentation tasks, we also conduct ablation studies on CLAG-SNN and perform low-data regime experiments on the ModelNet40 dataset. Robustness analysis and parameter sensitivity analysis are provided in Appendices F and G; dataset descriptions, energy estimation methods, training configurations, and comparisons with conventional artificial neural networks (ANNs) are placed in Appendices C to E and Appendix H.

4.1 Comparison with Existing Methods

Results on 3D Classification. We conduct experiments on the widely used ModelNet10 and ModelNet40 datasets, as well as the more challenging ScanObjectNN dataset. As shown in Table 1, CLAG-SNN significantly outperforms existing SNN-based methods on both the ModelNet10 and ModelNet40 datasets, and also achieves better performance than the Vanilla ANN with the same

Table 1: Comparison results with existing SNN methods on the 3D classification task over the ModelNet10, ModelNet40, and ScanObjectNN datasets, and on the Semantic KITTI val benchmark for 3D semantic segmentation. Following previous representative mainstream studies Shan et al. (2025); Hu et al. (2024), this work also following their method. Detailed description refers to Appendix D. The symbol † denotes results reproduced based on official code, and the symbol $*$ indicates results directly reported from the corresponding literature Qiu et al. (2025). FPS denotes the average number of samples processed per second during the inference stage.

ModelNet10 Wu et al. (2015)			
Method	**Power(mJ) ↓**	**FPS ↑ / T**	**Accuracy (%) ↑**
Vanilla ANN† Qiu et al. (2025)[AAAI]	0.08	48.21 / -	93.63
Spiking PointNet* Ren et al. (2023)[NeurIPS]	0.03	- / 4	93.31
Spike PointNet* Lan et al. (2023)[ICCV]	0.13	- / 16	93.54
E-3DSNN† Qiu et al. (2025)[AAAI]	0.009	45.78 / 1	91.84
CLAG-SNN (Ours)	**0.007**	**50.69** / 1	**94.86**
ModelNet40 Wu et al. (2015)			
Method	**Power(mJ) ↓**	**FPS ↑ / T**	**Accuracy (%) ↑**
Vanilla ANN† Qiu et al. (2025)[AAAI]	0.08	51.11 / -	90.90
Spiking PointNet * Ren et al. (2023)[NeurIPS]	0.03	- / 4	88.20
Spike PointNet * Lan et al. (2023)[ICCV]	0.13	- / 16	88.60
P2SResLNet * Wu et al. (2024a)[AAAI]	-	- / 4	88.70
E-3DSNN† Qiu et al. (2025)[AAAI]	0.012	48.34 / 1	90.72
CLAG-SNN (Ours)	**0.008**	**60.13** / 1	**91.47**
ScanObjectNN Uy et al. (2019)			
Method	**OBJ-BG ↑**	**OBJ-ONLY ↑**	**PB-T50-RS ↑**
Spiking PointNet* Ren et al. (2023)[NeurIPS]	72.2	76.4	64.1
P2SResLNet* Wu et al. (2024a)[AAAI]	78.6	80.2	74.5
SpikingPointTransformer* Wu et al. (2025a)[AAAI]	82.8	83.4	78.0
SpikingPointMamba† Wu et al. (2025b)[ICCV]	88.7	88.2	82.5
CLAG-SNN (Ours)	**89.2**	**88.6**	**82.8**
Semantic KITTI Behley et al. (2019)			
Method	**Power(mJ) ↓**	**FPS ↑ / T**	**mIoU(%) ↑**
Vanilla ANN† Qiu et al. (2025)	54.10	3.86 / -	69.40
SpikePointNet* Lan et al. (2023)[ICCV]	-	- / 1	12.10
SpikePointformer* Wu et al. (2024b)[CVPR]	13.10	- / 1	67.20
E-3DSNN† Qiu et al. (2025)[AAAI]	4.72	4.30 / 1	68.03
CLAG-SNN (Ours)	**4.62**	**4.83** / 1	**69.46**

Table 2: Comparison results with existing SNN methods on the KITTI validation set for 3D object detection. We report Car 3D AP under the R11 protocol. The symbol † denotes results reproduced based on official code, and $*$ denotes results directly reported from the corresponding literature. FPS indicates inference speed, and T denotes inference timesteps.

Method	**Power(mJ)↓**	**FPS ↑ / T**	**Car 3D AP (R11) ↑**		
			Easy	**Mod.**	**Hard**
Vanilla ANN† Qiu et al. (2025)[AAAI]	31.20	68.5 / -	89.4	83.7	78.2
SpikePointRCNN* Qiu et al. (2025)[AAAI]	4.40	- / 1	83.8	72.1	71.9
E-3DSNN† Qiu et al. (2025)[AAAI]	3.42	62.8 / 1	89.3	83.6	78.6
CLAG-SNN (Ours)	**3.34**	**66.7 / 1**	**89.7**	**85.1**	**79.1**

network architecture. The key advantage of CLAG-SNN lies in its exceptional computational efficiency. While achieving classification accuracy that surpasses existing SOTA methods, CLAG attains a high inference speed of 60.13 FPS, representing a 24.4% improvement over the E-3DSNN model. Moreover, CLAG-SNN further reduces the spike firing rate of the network, which theoretically leads to lower energy consumption.

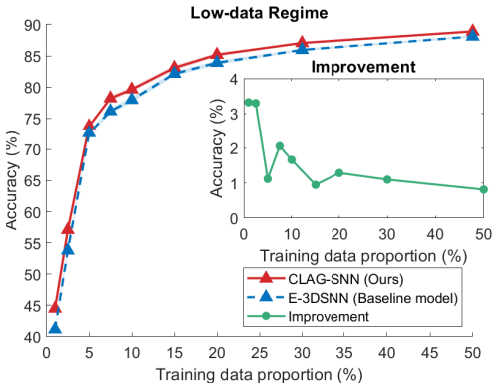

Figure 3: Low-data regime expriments. Classification accuracy comparison between CLAG-SNN and the E-3DSNN under different training data proportion on the ModelNet40 dataset.

Results on 3D Semantic Segmentation. We further evaluate CLAG-SNN on the more challenging, large-scale, and real-world Semantic KITTI dataset. As reported in Table 1, CLAG-SNN achieves an mIoU of 69.46%, surpassing the existing SOTA methods. In addition, CLAG-SNN also demonstrates clear advantages in terms of inference speed and power consumption, further validating its effectiveness and efficiency in complex 3D recognition task.

Results on 3D Object Detection. Following prior work, we use the VoxelRCNN detection framework and report the Car 3D AP under the R11 protocol for Easy, Moderate, and Hard difficulty levels. As shown in Table 2, compared with the E-3DSNN model, CLAG-SNN consistently improves detection performance across all difficulty levels, and obtains a prominent gain on the moderate difficulty case. In addition, the inference frame rate increases from 62.8 FPS to 66.7 FPS, achieving a 6.2% improvement.

Results under the Low-data Regime. We conduct low-data regime experiments on the ModelNet40 dataset. As illustrated in Figure 3, CLAG-SNN consistently outperforms the E-3DSNN model across all data proportions. The advantage becomes particularly prominent when training data is scarce, achieving performance improvements of 3.32% and 3.30% at 1% and 2.5% data ratios, respectively. These results demonstrate that CLAG-SNN has stronger sample utilization capability, enabling it to achieve superior performance in data-limited scenarios.

Table 3: Ablation study of CLAG-SNN on the ModelNet40 and Semantic KITTI val datasets. FPS denotes the average number of samples processed per second during the inference stage.

NAAB	HSTC	Accuracy / mIoU (%)	FPS	Power
3D classification with ModelNet40 dataset				
✗	✗	90.72	48.34	0.012
✗	✓	91.00	56.18	0.010
✓	✗	91.45	51.38	0.010
✓	✓	**91.47**	**60.13**	**0.008**
3D segmentation with Semantic KITTI dataset				
✗	✗	68.03	4.30	4.72
✗	✓	68.11	4.34	4.69
✓	✗	68.65	4.31	4.71
✓	✓	**69.46**	**4.83**	**4.62**

4.2 Ablation Study

We conduct the ablation study on the ModelNet40 and Semantic KITTI datasets. As shown in Table 3, the results indicate that using either NAAB or HSTC alone fails to achieve optimal performance. In contrast, CLAG-SNN framework significantly improves efficiency while enhancing accuracy. On ModelNet40, the FPS increases from 48.34 to 60.13, corresponding to a 24.4% improvement. These results demonstrate that the proposed method achieves a better balance between information compression and feature discriminability.

5 Conclusion

This paper proposes CLAG-SNN, a hierarchical nonlinear dependency regulation framework for SNN-based 3D point cloud recognition. By introducing the Compact Local, Abstract Global (CLAG) principle, CLAG-SNN adaptively regulates nonlinear dependency across representation hierarchies, preserving local structural information while promoting high-level semantic abstraction. Extensive experiments on 3D classification, object detection, and semantic segmentation benchmarks demonstrate that CLAG-SNN consistently outperforms existing SNN-based methods. Furthermore, CLAG-SNN achieves improvements in inference efficiency and maintains robust performance under limited-data settings, validating the effectiveness of hierarchy-aware dependency regulation for efficient spike-based 3D recognition.

AI USE STATEMENT

In this work, we used generative AI tools to refine language expression and proofread the English manuscript.

REFERENCES

Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9297–9307, 2019.

Yuchen Chang and Kaiping Wang. Multi-stage refinement network for point cloud completion based on geodesic attention. *Scientific Reports*, 15(1):3570, 2025.

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.

James J DiCarlo, Davide Zoccolan, and Nicole C Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, 2012.

Steve Furber. Large-scale neuromorphic computing systems. *Journal of neural engineering*, 13(5): 051001, 2016.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pp. 63–77. Springer, 2005.

Jay Hegdé and David C Van Essen. Strategies of shape representation in macaque visual area v2. *Visual neuroscience*, 20(3):313–328, 2003.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*, pp. 10–14. IEEE, 2014.

JiaKui Hu, Man Yao, Xuerui Qiu, Yuhong Chou, Yuxuan Cai, Ning Qiao, Yonghong Tian, Bo Xu, and Guoqi Li. High-performance temporal reversible spiking neural networks with $o(l)$ training memory and $o(1)$ inference cost. *arXiv preprint arXiv:2405.16466*, 2024.

David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3d recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17545–17555, 2023.

Yuxiang Lan, Yachao Zhang, Xu Ma, Yanyun Qu, and Yun Fu. Efficient converted spiking neural network for 3d and 2d classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9211–9220, 2023.

Yijie Lu, Zhiyi Pan, Renrui Zhang, Yanhao Jia, Kaiwei Che, and Zhaokun Zhou. Spatially-enhanced spiking neural network for efficient point cloud analysis. *Neural Networks*, pp. 108190, 2025.

Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *European Conference on Computer Vision*, pp. 253–272. Springer, 2024.

Priyadarshini Panda, Sai Aparna Aketi, and Kaushik Roy. Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization. *Frontiers in Neuroscience*, 14:653, 2020.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Ning Qiao, Hesham Mostafa, Federico Corradi, Marc Osswald, Fabio Stefanini, Dora Sumislawska, and Giacomo Indiveri. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Frontiers in neuroscience*, 9:141, 2015.

Xuerui Qiu, Man Yao, Jieyuan Zhang, Yuhong Chou, Ning Qiao, Shibo Zhou, Bo Xu, and Guoqi Li. Efficient 3d recognition with event-driven spike sparse convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20086–20094, 2025.

N Apurva Ratan Murty and Sripati P Arun. Dynamics of 3d view invariance in monkey inferotemporal cortex. *Journal of Neurophysiology*, 113(7):2180–2194, 2015.

Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. *arXiv preprint arXiv:2005.01807*, 2020.

Dayong Ren, Zhe Ma, Yuanpei Chen, Weihang Peng, Xiaode Liu, Yuhan Zhang, and Yufei Guo. Spiking pointnet: Spiking neural networks for point clouds. *Advances in Neural Information Processing Systems*, 36:41797–41808, 2023.

Yimeng Shan, Malu Zhang, Rui-jie Zhu, Xuerui Qiu, Jason K Eshraghian, and Haicheng Qu. Advancing spiking neural networks towards multiscale spatiotemporal interaction learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1501–1509, 2025.

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1588–1597. IEEE, 2019.

Daniel Worrall and Gabriel Brostow. Cubenet: Equivariance to 3d rotation and translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 567–584, 2018.

Peixi Wu, Bosong Chai, Hebei Li, Menghua Zheng, Yansong Peng, Zeyu Wang, Xuan Nie, Yueyi Zhang, and Xiaoyan Sun. Spiking point transformer for point cloud classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21563–21571, 2025a.

Peixi Wu, Bosong Chai, Menghua Zheng, Wei Li, Zhangchi Hu, Jie Chen, Zheyu Zhang, Hebei Li, and Xiaoyan Sun. Efficient spiking point mamba for point cloud analysis. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26393–26403. IEEE, 2025b.

Qiaoyun Wu, Quanxiao Zhang, Chunyu Tan, Yun Zhou, and Changyin Sun. Point-to-spike residual learning for energy-efficient 3d point cloud classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 6092–6099, 2024a.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4840–4851, 2024b.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Shuangming Yang and Badong Chen. Effective surrogate gradient learning with high-order information bottleneck for spike-based machine intelligence. *IEEE transactions on neural networks and learning systems*, 36(1):1734–1748, 2023.

Shuangming Yang, Bernabé Linares-Barranco, Yuzhu Wu, and Badong Chen. Self-supervised high-order information bottleneck learning of spiking neural network for robust event-based optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2280–2297, 2024.

Shuangming Yang, Jianshi Tang, Tianhang Zhou, Bernabe Linares-Barranco, Giacomo Indiveri, Xinhu Zheng, Mostafa Rahimi Azghadi, and Badong Chen. Efficient training of neuromorphic electronics. *Nature Electronics*, pp. 1–17, 2026.

Man Yao, Guangshe Zhao, Hengyu Zhang, Yifan Hu, Lei Deng, Yonghong Tian, Bo Xu, and Guoqi Li. Attention spiking neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):9393–9410, 2023.

Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021.

Ruokai Yin, Abhishek Moitra, Abhiroop Bhattacharjee, Youngeun Kim, and Priyadarshini Panda. Sata: Sparsity-aware training accelerator for spiking neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(6):1926–1938, 2022.

Changyu Zeng, Wei Wang, Anh Nguyen, Jimin Xiao, and Yutao Yue. Self-supervised learning for point cloud data: A survey. *Expert Systems with Applications*, 237:121354, 2024.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16259–16268, 2021.

Algorithm 1: CLAG-SNN training procedure

Input: Mini-batch $\{(P_b, Y_b)\}_{b=1}^{B}$; SNN encoder f_θ; I-LIF firing Q; spectral energy threshold τ (Definition 2); keep ratio $\rho_{\min}$; prices $\lambda_{\mathrm{HSTC}}, \lambda_{\mathrm{NAAB}}$ (Proposition 2); anchor β; density weight κ.
Output: Updated parameters θ.

Spiking token generation.
Voxelize $\{P_b\}_{b=1}^{B}$ and compute pre-spike features $\mathbf{A}^{(l)} \leftarrow f_\theta(P)$.
Obtain integer spikes $\mathbf{S}^{(l)} \leftarrow Q(\mathbf{A}^{(l)})$ and tokens $\mathbf{X}^{(l)} \leftarrow \psi_l(\mathbf{S}^{(l)})$.

Compact Local (HSTC).
for $b = 1$ **to** B **do**
  Extract the active token matrix $\mathbf{X}^{(b)} \in \mathbb{R}^{N_b \times d_l}$.
  Center and factorize $\bar{\mathbf{X}}^{(b)} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.
  Compute the spectral energy $E_b(k)$ and $k_{\mathrm{svd}}^{(b)} = \min\{k \mid E_b(k) \geq \tau\}$.
  Set the budget $k^{(b)} = \Pi_{[N_{\min}, N_{\max}]}(\max\{k_{\mathrm{svd}}^{(b)}, \lfloor \rho_{\min} N_b \rfloor\})$.
  Score tokens $s_i = \tilde{m}_i + \alpha_t \tilde{d}_i$ and select $\mathcal{I}_b^{\mathrm{keep}} = \mathrm{TopK}(\{s_i\}, k^{(b)})$.
  Compute $\mathbf{t}_{\mathrm{pre}}^{(b)}$ and $\mathbf{t}_{\mathrm{post}}^{(b)}$.
Stack the descriptors into $\mathbf{T}_{\mathrm{pre}}$ and $\mathbf{T}_{\mathrm{post}}$ and set $\mathcal{L}_{\mathrm{HSTC}} = \mathrm{Dep}(\mathbf{T}_{\mathrm{pre}}, \mathbf{T}_{\mathrm{post}})$.

Abstract Global (NAAB).
Compute input descriptors $\mathbf{X}^{\mathrm{in}}$ and latent descriptors $\mathbf{Z}$.
Set $\mathcal{L}_{\mathrm{dep}} = \mathrm{Dep}(\mathbf{X}^{\mathrm{in}}, \mathbf{Z})$ by equation 8.
Estimate the latent entropy $H(\mathbf{Z})$ and the dispersion guard $\mathcal{L}_{\mathrm{KDE}}$, and set $\mathcal{L}_{\mathcal{H}} = H(\mathbf{Z}) - \kappa \mathcal{L}_{\mathrm{KDE}}$.
Set the global module objective $\mathcal{L}_{\mathrm{NAAB}} = \mathcal{L}_{\mathrm{dep}} - \beta \mathcal{L}_{\mathcal{H}}$ by equation 9.

Optimization.
$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{task}} - \lambda_{\mathrm{HSTC}} \mathcal{L}_{\mathrm{HSTC}} + \lambda_{\mathrm{NAAB}} \mathcal{L}_{\mathrm{NAAB}}$.
Update $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathrm{total}}$.
return θ.

A Training Algorithm

CLAG-SNN processes 3D point clouds with a spiking backbone and trains by minimizing equation 11, which enforces the two dependency budgets of CLAG. Algorithm 1 summarizes the procedure.

B Proofs of the Theoretical Results

We first prove Proposition 2, whose argument is self-contained, and then state the assumptions used in Theorem 1 and prove Lemma 1, Theorem 1, and Theorem 2.

Proof of Proposition 2. For (i), rearrange equation 11 as

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{task}} - \lambda_{\mathrm{HSTC}}\big(\mathcal{L}_{\mathrm{HSTC}} - \varepsilon_{\mathrm{loc}}\big) + \lambda_{\mathrm{NAAB}}\big(\mathcal{L}_{\mathrm{NAAB}} - \varepsilon_{\mathrm{glb}}\big) + C,$$

where $C = \lambda_{\mathrm{NAAB}}\varepsilon_{\mathrm{glb}} - \lambda_{\mathrm{HSTC}}\varepsilon_{\mathrm{loc}}$ is constant once the prices are fixed. For feasible θ, the two slacks satisfy $\mathcal{L}_{\mathrm{HSTC}} - \varepsilon_{\mathrm{loc}} \geq 0$ and $\mathcal{L}_{\mathrm{NAAB}} - \varepsilon_{\mathrm{glb}} \leq 0$, so each price term is non-positive and $\mathcal{L}_{\mathrm{total}}(\theta) \leq \mathcal{L}_{\mathrm{task}}(\theta) + C$. Minimizing over θ preserves the inequality against the constrained optimum, which yields the stated lower bound; tightening it by choosing the multipliers is exactly dual maximization, which is why we call them prices. For (ii), at a saddle point a positive price requires its constraint to be tight, since otherwise the Lagrangian could be raised by lowering that price; each price–slack product in equation 12 therefore vanishes. □

Assumption 1 (Compact manifold and bounded representations)**.** *The input distribution is supported on a compact d_{eff}-dimensional manifold $\mathcal{M}$, and the latent descriptors satisfy $\|\mathbf{z}_b\|_2 \leq B_z$ for all samples.*

Assumption 2 (Lipschitz prediction)**.** *The loss $\ell(\cdot, y)$ is L_ℓ-Lipschitz in its first argument, and the prediction head g is L_g-Lipschitz.*

Assumption 3 (Dependency-controlled distortion)**.** *With the median-heuristic Gaussian kernels of Definition 1, there exist constants $c_1, c_4 > 0$ such that (i) the semantic distortion attributable to residual input dependency satisfies $\mathfrak{D}(X, Z) \le c_1\sqrt{\mathrm{HSIC}(\mathbf{X}^{\mathrm{in}}, \mathbf{Z})}$, and (ii) the structural distortion attributable to token compaction satisfies $\mathfrak{E}_{\mathrm{token}} \le \frac{c_4}{\sqrt{\mathrm{HSIC}(\mathbf{T}_{\mathrm{pre}}, \mathbf{T}_{\mathrm{post}}) + \varepsilon_t}}$, where $\varepsilon_t > 0$ prevents degeneration when the dependency is close to zero.*

Assumption 4 (Entropy-controlled latent covering)**.** *The effective covering complexity of the latent set satisfies $\mathcal{N}(\mathcal{Z}, \epsilon) \le \left(\frac{C}{\epsilon}\right)^{d_{\mathrm{eff}}} \left(1 + \frac{1}{H(\mathbf{Z}) + \varepsilon_0}\right)$, where $H(\mathbf{Z})$ is the latent entropy in equation **??** and $\varepsilon_0 > 0$.*

Assumption 5 (Bounded dispersion)**.** *Uncontrolled latent dispersion adds representation distortion bounded by $\mathfrak{E}_{\mathrm{KDE}} \le c_3\, \mathcal{L}_{\mathrm{KDE}}$.*

Lemma 1 (Latent perturbation bound)**.** *Under Assumptions 1–5, the discrepancy $\Delta_{\mathcal{Z}}$ between the learned latent representation and an ideal compact, task-relevant representation satisfies*

$$\Delta_{\mathcal{Z}} \le c_1\sqrt{\mathcal{L}_{\mathrm{dep}}} + c_2\frac{1}{\sqrt{H(\mathbf{Z}) + \varepsilon_0}} + c_3\, \mathcal{L}_{\mathrm{KDE}} + c_4\frac{1}{\sqrt{\mathcal{L}_{\mathrm{HSTC}} + \varepsilon_t}} + c_5\,(1 - r_{\mathrm{keep}}).$$

Proof. The discrepancy aggregates five sources.

First, residual input dependency: by Assumption 3(i), the distortion caused by the remaining dependence between the input descriptor and the latent descriptor is at most $c_1\sqrt{\mathcal{L}_{\mathrm{dep}}}$.

Second, latent collapse: by Assumption 4, a small entropy tightens the covering complexity of the latent set, which inflates the collapse-induced distortion as $c_2/\sqrt{H(\mathbf{Z}) + \varepsilon_0}$; a larger entropy corresponds to a broader, more diverse latent support and reduces this term.

Third, excess dispersion: by Assumption 5, the distortion attributable to uncontrolled dispersion is at most $c_3\, \mathcal{L}_{\mathrm{KDE}}$, which is why NAAB anchors dispersion through $\mathcal{L}_{\mathrm{KDE}}$.

Fourth, token compaction: by Assumption 3(ii), the structural distortion induced by compaction is at most $c_4/\sqrt{\mathcal{L}_{\mathrm{HSTC}} + \varepsilon_t}$.

Finally, discarded mass: by Proposition 1, the descriptor distortion of compaction is at most $2(1 - r_{\mathrm{keep}}) \max_b R_b$, and since $\|\mathbf{z}_b\|_2 \le B_z$ by Assumption 1, $\max_b R_b$ is a constant absorbed into c_5.

Summing the five contributions yields the stated bound. □

Proof of Theorem 1. Under Assumptions 1–5, fix $h = g \circ f_\theta$ and $\delta \in (0, 1)$. Decompose the generalization gap into a representation error and an estimation error:

$$R(h) - \widehat{R}_S(h) = \underbrace{\big(R(h) - R(h^\star)\big)}_{\text{representation error}} + \underbrace{\big(R(h^\star) - \widehat{R}_S(h)\big)}_{\text{estimation error}},$$

where $h^\star$ denotes the hypothesis induced by an ideal compact, task-relevant representation.

Since ℓ is L_ℓ-Lipschitz and g is L_g-Lipschitz (Assumption 2), the representation error is bounded by $\big|R(h) - R(h^\star)\big| \le L_\ell L_g\, \Delta_{\mathcal{Z}}$, and Lemma 1 supplies the five-term bound in equation 14.

For the estimation error, since the data lie on a compact d_{eff}-dimensional manifold (Assumption 1), the effective statistical complexity of the hypothesis class scales with d_{eff}. A standard covering-number (or Rademacher complexity) argument gives, with probability at least $1 - \delta$,

$$\big|R(h^\star) - \widehat{R}_S(h)\big| \le c_6\sqrt{\frac{d_{\mathrm{eff}}}{n}} + \sqrt{\frac{\log(2/\delta)}{2n}}.$$

Combining the two terms proves the claim. □

Theorem 2 (Sample complexity)**.** *Under the assumptions of Theorem 1, to guarantee $R(h) - \widehat{R}_S(h) \le \epsilon$ with probability at least $1 - \delta$, it suffices that*

$$n \ge \frac{C}{\epsilon^2}\left[d_{\mathrm{eff}} \log\left(\frac{1}{\epsilon}\right) \cdot \Gamma_{\mathrm{CLAG}} + \log\left(\frac{1}{\delta}\right)\right],$$

where

$$\Gamma_{\mathrm{CLAG}} = (1+\mathcal{L}_{\mathrm{dep}})\left(1+\frac{1}{H(\mathbf{Z})+\varepsilon_0}\right)(1+\mathcal{L}_{\mathrm{KDE}})^2\left(1+\frac{1}{\mathcal{L}_{\mathrm{HSTC}}+\varepsilon_t}\right)(1+(1-r_{\mathrm{keep}}))^2 .$$

Proof. From Theorem 1, the gap is bounded by

$$R(h)-\widehat{R}_S(h) \le C\,\Omega_{\mathrm{CLAG}} + C\sqrt{\frac{d_{\mathrm{eff}}}{n}} + \sqrt{\frac{\log(2/\delta)}{2n}},$$

where

$$\Omega_{\mathrm{CLAG}} = \sqrt{\mathcal{L}_{\mathrm{dep}}} + \frac{1}{\sqrt{H(\mathbf{Z})+\varepsilon_0}} + \mathcal{L}_{\mathrm{KDE}} + \frac{1}{\sqrt{\mathcal{L}_{\mathrm{HSTC}}+\varepsilon_t}} + (1-r_{\mathrm{keep}}),$$

and the constants are absorbed into C.

For a hypothesis class supported on a compact d_{eff}-dimensional manifold, the covering number satisfies $\log \mathcal{N}(\epsilon, \mathcal{H}, \|\cdot\|) \le d_{\mathrm{eff}} \log(C/\epsilon)$. The representation-dependent factors inflate the effective covering number multiplicatively, which gives

$$\log \mathcal{N}_{\mathrm{eff}} \le d_{\mathrm{eff}} \log\left(\frac{C}{\epsilon}\right)\cdot \Gamma_{\mathrm{CLAG}},$$

with Γ_{CLAG} as stated in the theorem. By a standard uniform-convergence argument, with probability at least $1-\delta$,

$$R(h)-\widehat{R}_S(h) \le C\sqrt{\frac{\log \mathcal{N}_{\mathrm{eff}} + \log(1/\delta)}{n}}.$$

Requiring the statistical term to be at most ϵ yields the stated sample size. □

C Introduction to the Datasets

ModelNet40: The ModelNet40 dataset Wu et al. (2015) is a widely used benchmark for 3D shape classification. It contains 40 common object categories, such as tables, chairs, and airplanes, with a total of 12,311 3D CAD models. Following the standard split, 9,843 samples are used for training and 2,468 samples are used for testing. Due to its rich categories and standardized evaluation protocol, ModelNet40 has become one of the core benchmarks for evaluating 3D classification methods.

ModelNet10: The ModelNet10 dataset Wu et al. (2015) is a subset of ModelNet40 and contains 10 representative object categories, such as bed, toilet, sofa, and chair. It consists of 4,899 3D CAD models, including 3,991 training samples and 908 testing samples. Compared with ModelNet40, ModelNet10 contains fewer categories and is commonly used to evaluate 3D recognition models under a relatively simpler classification setting.

ScanObjectNN: The ScanObjectNN dataset Uy et al. (2019) is a real-world benchmark for 3D object classification built from ScanNet indoor scans. Unlike the synthetic CAD models of ModelNet, its objects are cluttered with background and are partially observed due to occlusion, which makes it a strictly harder test of recognition. It contains about 15,000 scanned objects covering 15 common categories, drawn from 2,902 unique object instances, and is evaluated in three variants: OBJ-BG, which retains the background scene; OBJ-ONLY, which keeps the segmented object; and PB-T50-RS, the hardest variant, in which about half of the points are replaced by background clutter and the objects are further rotated and scaled. We evaluate on all three variants; the background retained by OBJ-BG and PB-T50-RS is the kind of input-specific content that the global direction of CLAG suppresses.

KITTI: The KITTI dataset Geiger et al. (2012) is a large-scale real-world benchmark for autonomous driving and 3D object detection. It provides LiDAR point clouds and synchronized camera images collected from urban driving scenarios. The KITTI 3D object detection benchmark contains 7,481 labeled training samples, which are commonly divided into 3,712 training samples and 3,769 validation samples. In this work, we evaluate CLAG-SNN on the KITTI validation set for the Car category. Following the standard KITTI evaluation protocol, we report Car 3D Average Precision (AP) under the Easy, Moderate, and Hard difficulty levels.

Semantic KITTI: The Semantic KITTI dataset Behley et al. (2019) is a large-scale benchmark for LiDAR-based 3D semantic segmentation. It is built upon the raw KITTI odometry benchmark and contains 22 sequences of real-world driving scenes. Each LiDAR scan is densely annotated with point-wise semantic labels, enabling the evaluation of scene-level semantic understanding. Following common practice, we evaluate semantic segmentation performance using the mean Intersection over Union (mIoU) metric.

D Power Consumption

In the field of neuromorphic computing, theoretical evaluation of algorithmic energy efficiency typically relies on modeling the energy consumption of key operations. A widely adopted method estimates energy consumption from floating-point operations (FLOPs) combined with benchmark energy values for specific operations Panda et al. (2020); Yin et al. (2021). It computes the total theoretical energy by multiplying the computational load (FLOPs) of each model layer by the unit energy cost of the corresponding operation, such as multiply-accumulate (MAC) or accumulate (AC) operations:

$$\begin{aligned} E_{\text{total}} &= E_{\text{MAC}} \cdot FL^1_{\text{conv}} \\ &\quad + E_{\text{AC}} \cdot T \cdot \left(\sum_{n=2}^{N} FL^n_{\text{conv}} \cdot fr^n + \sum_{m=1}^{M} FL^m_{\text{fc}} \cdot fr^m \right) \end{aligned} \tag{15}$$

where N and M represent the total numbers of convolutional (Conv) and fully-connected (FC) layers respectively, E_{MAC} and E_{AC} denote the energy of a single MAC and a single AC operation (typically under specific process-node assumptions, e.g., $E_{\text{MAC}} = 4.6$ pJ and $E_{\text{AC}} = 0.9$ pJ at 45 nm Horowitz (2014); Rathi et al. (2020)), FL^n_{conv} and FL^m_{fc} indicate the FLOPs of the n-th convolutional layer and the m-th fully-connected layer, fr^n and fr^m represent the spike firing rates of the corresponding layers (the proportion of nonzero elements in the spike tensor), and T is the timestep. The model's main advantage is its simplicity, which facilitates qualitative energy comparison between different SNN and ANN algorithms.

However, the FLOPs-based model has notable limitations: it overlooks factors that matter in real hardware, such as memory access, data transfer overhead, and circuit dynamics Furber (2016). The estimate may therefore deviate from actual chip energy efficiency, particularly for two classes of architectures: (1) memory-access-dominated architectures, such as SATA Yin et al. (2022), where memory access dominates the energy and depends heavily on data-flow patterns; and (2) analog synapse architectures, such as Qiao et al. (2015), where spike generation may consume energy disproportionate to the computational load. An alternative metric is the number of synaptic operations (SOPs). In neuromorphic chips, synaptic processing, i.e., spike transmission and integration across synapses, is the dominant factor in system energy consumption, and the average energy of transmitting a single spike through a synapse is a core metric of overall energy efficiency Furber (2016). Many SNN studies have accordingly adopted SOPs as the basis of energy estimation **?**. For architectures that exploit event sparsity and event-driven computation, such as Loihi Davies et al. (2018), the total system energy can be reduced to a level approximately linear in the number of effective synaptic operations, thanks to on-core integrated memory and asynchronous circuit design. The formula is as follows:

$$E = C_E \cdot \#SOP = C_E \sum_i s_i c_i \tag{16}$$

where C_E denotes the energy per synaptic operation, $\#SOP = \sum_i s_i c_i$ represents the total number of synaptic operations, s_i is the number of spikes emitted by neuron i, and c_i is the number of synaptic connections of neuron i. This linear energy model highlights the direct proportionality between energy consumption and effective information transmission, providing a physically meaningful framework for energy-efficiency evaluation in neuromorphic systems.

E Hyper-Parameters and Experimental Environment

The experiments were conducted on a Linux server equipped with four NVIDIA GeForce RTX 4090 GPUs. The training configurations of each experiment are summarized in Table 4.

Table 4: Hyper-parameter settings.

Parameter	ModelNet40 / 10	Parameter sensitivity	Ablation Study	Low-data regime	Semantic KITTI	KITTI
Learning Rate	1e-1	1e-1	1e-1	1e-1	2e-3	1e-2
Weight Decay	1e-4	1e-4	1e-4	1e-4	5e-3	1e-2
Batch Size	16	16	16	16	2	8
Training Epochs	100	100	100	10	10	80

Table 5: Robustness testing results. † means our implementation based on official code.

Noise type	Method	Accuracy (%)
SP	E-3DSNN† Qiu et al. (2025)[AAAI]	89.95
	CLAG-SNN (Ours)	**90.39**
Discrete point	E-3DSNN† Qiu et al. (2025)[AAAI]	89.99
	CLAG-SNN (Ours)	**90.56**

For classification we adopt Overall Accuracy (OA), which integrates the classification results over all categories into a single global measure. For semantic segmentation we adopt mean Intersection over Union (mIoU), defined as the average IoU over all categories, which treats all classes equally and reflects multi-class segmentation quality.

F Robustness Testing

We evaluated the robustness of CLAG-SNN in 3D classification and compared it with the baseline. Following common practice, we added salt-and-pepper (SP) noise and discrete point noise to the point clouds. As shown in Table 5, CLAG-SNN improves accuracy under both noise types, which suggests that structure-preserving token compaction also absorbs some of the corrupted local evidence.

G Parameter Sensitivity Analysis

We analyze the sensitivity of NAAB to its two weights, λ_{NAAB}, which prices the global budget in equation 10, and β, which sets the strength of the entropy anchor in equation 9; results are shown in Table 6. When β is too small, the entropy safeguard becomes ineffective and the latent representations degenerate, causing information loss; when β is too large, the latent features become overly dispersed and accuracy drops. Similarly, a small λ_{NAAB} fails to remove enough redundant input dependency, while an excessively large value compresses informative structures. Across the tested grid, $\beta = 0.1$ with $\lambda_{\mathrm{NAAB}} = 0.1$ performs best (91.45%), and the accuracy varies within 0.8% over the whole grid, indicating that NAAB does not rely on delicate weight tuning.

Table 6: Parameter sensitivity analysis of NAAB.

Accuracy (%) β \ λ_{NAAB}	0.01	0.05	0.1	0.2
0.01	90.73	90.92	90.80	90.65
0.05	91.05	90.88	91.21	91.25
0.1	91.37	91.41	**91.45**	91.37
0.2	91.28	91.35	91.26	91.30

Table 7: Comparisons with representative ANN and SNN methods on ModelNet10 and ModelNet40 in 3D classification. Power is estimated following Shan et al. (2025); Hu et al. (2024). T is the number of timesteps, and D is the maximum integer spike value of the I-LIF neuron. † means our implementation based on official code.

Method	T×D	Power(mJ) ↓	Accuracy (%) ↑
ModelNet10 Wu et al. (2015)			
Vanilla ANN† Qiu et al. (2025)[AAAI]	N/A	0.08	93.63
CubeNet Worrall & Brostow (2018)[CVPR]	N/A	-	94.60
3DShapeNets Wu et al. (2015)[CVPR]	N/A	-	83.54
Spiking PointNet Ren et al. (2023)[NeurIPS]	4×1	0.03	93.31
Spike PointNet Lan et al. (2023)[ICCV]	16×1	0.13	93.54
E-3DSNN† Qiu et al. (2025)[AAAI]	1×4	0.009	91.84
CLAG-SNN (Ours)	1×4	0.007	94.86
ModelNet40 Wu et al. (2015)			
Vanilla ANN† Qiu et al. (2025)[AAAI]	N/A	0.08	90.90
PointNet Qi et al. (2017a)[CVPR]	N/A	0.14	89.20
Pointformer Zhao et al. (2021)[ICCV]	N/A	0.11	93.70
3DShapeNets Wu et al. (2015)[CVPR]	N/A	0.15	77.30
Spiking PointNet Ren et al. (2023)[NeurIPS]	4×1	0.03	88.20
Spike PointNet Lan et al. (2023)[ICCV]	16×1	0.13	88.60
P2SResLNet Wu et al. (2024a)[AAAI]	4×1	-	88.70
E-3DSNN† Qiu et al. (2025)[AAAI]	1×4	0.012	90.72
CLAG-SNN (Ours)	1×4	0.008	91.47

Table 8: Comparisons with representative ANN and SNN methods on the Semantic KITTI val benchmark Behley et al. (2019). Power is estimated following Shan et al. (2025); Hu et al. (2024). T is the number of timesteps, and D is the maximum integer spike value of the I-LIF neuron. † means our implementation based on official code.

Method	T×D	Power (mJ) ↓	mIoU (%) ↑
Vanilla ANN† Qiu et al. (2025)[AAAI]	N/A	54.10	69.40
PointNet Qi et al. (2017a)[CVPR]	N/A	-	14.60
Point Transformer V3 Wu et al. (2024b)[CVPR]	N/A	47.10	72.30
SphereFormer Lai et al. (2023)[CVPR]	N/A	49.20	67.80
Spike PointNet Lan et al. (2023)[ICCV]	1×4	-	12.10
SpikePointformer Wu et al. (2024b)[CVPR]	1×4	13.10	67.20
E-3DSNN† Qiu et al. (2025)[AAAI]	1×4	4.72	68.03
CLAG-SNN (Ours)	1×4	4.62	69.46

H Comparison with ANNs

Tables 7, 8, and 9 compare CLAG-SNN with representative ANN methods on 3D classification, 3D semantic segmentation, and 3D object detection. CLAG-SNN narrows the gap to ANN models on all three tasks while retaining the energy efficiency of spike-driven computation.

Table 9: Comparison with representative ANN and SNN methods on the KITTI val benchmark for 3D object detection. We report Car 3D AP under the R11 protocol. Power is estimated following Shan et al. (2025); Hu et al. (2024). T is the number of timesteps, and D is the maximum integer spike value of the I-LIF neuron. † means our implementation based on official code.

Method	T×D	Power(mJ)↓	Car 3D AP (R11) ↑		
			Easy	Mod.	Hard
Vanilla ANN† Qiu et al. (2025)[AAAI]	N/A	31.20	89.4	83.7	78.2
SpikePointRCNN* Qiu et al. (2025)[AAAI]	1×4	4.40	83.8	72.1	71.9
E-3DSNN† Qiu et al. (2025)[AAAI]	1×4	3.42	89.3	83.6	78.6
CLAG-SNN (Ours)	1×4	3.34	89.7	85.1	79.1