VGESP: VIEWPOINT-INDUCED GEOMETRIC EVENTS FOR SPIKING POINT CLOUD UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Point cloud understanding has advanced rapidly, but the growing computational demands of modern 3D networks motivate more energy-efficient alternatives. Spiking neural networks (SNNs) provide a promising paradigm through sparse binary spikes and event-driven computation. However, existing spiking point cloud methods primarily focus on network-side feature processing, while the temporal structure of static point cloud inputs remains underexplored. To address this gap, we propose VGESP, a spike-driven point cloud framework that combines geometry-induced temporal representation with efficient global spiking attention. Specifically, we introduce Viewpoint-Induced Geometric Events (VGE), which convert visibility transitions and significant depth variations along short, ordered virtual-camera trajectories into sparse binary event streams, providing explicit geometric temporal cues for spiking computation. To efficiently integrate these temporal geometric cues, we further develop Vector Subtractive Spiking Attention (VSSA), which models geometric variations through vector-level discrepancies between binary spike features without explicitly constructing dense point-to-point attention maps. Extensive experiments demonstrate that VGESP achieves state-of-the-art performance among SNN-based point cloud methods on multiple benchmarks, outperforms many ANN-based counterparts, and retains competitive computational and energy efficiency. Our source code will be released to the public.

1 INTRODUCTION

Point cloud understanding is fundamental to 3D perception and supports a wide range of applications, including autonomous driving, robotics, augmented reality, and embodied intelligence Zhou et al. (2026c); Ha et al. (2024); Zhou et al. (2025). Recent point-based networks have achieved remarkable progress through stronger local geometric modeling and long-range feature interaction, yet processing large numbers of irregularly distributed 3D points remains computationally demanding. Spiking neural networks (SNNs) Maass (1997) provide a promising direction toward this goal. By communicating through sparse binary spikes and exploiting temporally evolving neuronal states, SNNs offer the potential for event-driven computation and low energy consumption on neuromorphic hardware. Following their success in 2D vision, SNNs have also been introduced to point cloud analysis and shown potential for efficient modeling of irregular 3D structures.

Most existing studies, however, mainly focus on how point cloud features should be processed by spiking networks, while the temporal characteristics of the input representation itself remain relatively underexplored. A point cloud is typically represented as a static set of 3D samples, whereas SNNs perform computation through neuronal states evolving over multiple timesteps. Consequently, temporal information in current spiking point cloud models is often introduced through neural simulation or repeated encoding of the same static geometry. Although effective, such temporal evolution is not necessarily associated with explicit changes in the observed 3D structure. This motivates us to explore whether the rich spatial geometry of a static point cloud can itself be transformed into meaningful temporal signals for spiking computation.

A useful clue comes from viewpoint-dependent geometry. Existing virtual-camera-based point cloud methods Su et al. (2015); Zhang et al. (2025a) have demonstrated that observing a point cloud from different viewpoints provides structural cues beyond conventional Euclidean geometry, yet different virtual cameras are generally treated as independent observations. In fact, even when

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.

the underlying scene remains unchanged, its geometric observation varies as the viewpoint moves: occluded points may become visible, visible points may become occluded, and points around object contours, thin structures, or surface discontinuities may exhibit pronounced depth variations. These changes are directly induced by the underlying 3D structure and naturally evolve along the viewpoint trajectory. Therefore, moving the observer provides a natural mechanism for converting static spatial geometry into temporally varying geometric evidence.

Such viewpoint-evolving geometric cues are distributed across different points and multiple timesteps, and therefore require effective global interaction for integration. For 3D point clouds, however, global interaction should not only focus on feature similarity: geometric variations arising from different spatial positions and local structures are themselves informative and can further manifest as distinct firing states after spiking encoding. Conventional similarity-based attention primarily emphasizes feature agreement, while its all-pairs interaction also incurs substantial computational and memory costs for large point sets, which are further accumulated over multiple timesteps. Therefore, efficiently exploiting geometry-induced spike discrepancies while maintaining global interaction becomes another key challenge in utilizing the above temporal geometric representation.

To address these challenges, we propose VGESP, a spike-driven framework for point cloud understanding that combines Viewpoint-Induced Geometric Events (VGE) with Vector Subtractive Spiking Attention (VSSA). Given a static point cloud, VGE constructs short and ordered virtual-camera trajectories and converts visibility transitions and significant depth variations between adjacent viewpoints into sparse binary geometric events, providing explicit geometric temporal cues for spiking computation. VSSA performs efficient global interaction by modeling vector-level discrepancies between binary spike features, thereby integrating viewpoint-varying geometric information without explicitly constructing dense point-to-point attention maps. Together, VGE and VSSA allow VGESP to transform static 3D geometry into temporal events and efficiently model them in the spiking domain. Our main contributions are summarized as follows:

- We introduce Viewpoint-Induced Geometric Events (VGE), a geometry-driven temporal representation that converts static point clouds into sparse and ordered event streams through controlled virtual-camera motion, providing explicit geometric temporal cues for spiking computation.
- We develop Vector Subtractive Spiking Attention (VSSA), a discrepancy-driven attention mechanism for 3D spiking features that models geometric variations through vector-level spike discrepancies while enabling efficient global interaction with linear complexity.
- Extensive experiments demonstrate that VGESP achieves state-of-the-art performance among SNN-based point cloud methods on multiple benchmarks, outperforms many ANN-based counterparts, and retains competitive computational and energy efficiency.

2 Related Work

ANN-based Point Cloud Understanding. Deep learning-based point cloud understanding has evolved from point-wise feature extraction toward increasingly effective local and global geometric modeling. PointNet Qi et al. (2017a) pioneered direct learning on unordered point sets through shared MLPs and symmetric aggregation, while PointNet++ Qi et al. (2017b) further introduced hierarchical neighborhood grouping to progressively capture local geometric structures at multiple scales. PointMLP Ma et al. (2022) subsequently demonstrated that strong point cloud representations can be achieved with a simple residual MLP architecture and lightweight geometric transformations, highlighting the effectiveness of carefully designed local feature aggregation. Transformer-based methods, represented by the Point Transformer series Zhao et al. (2021); Wu et al. (2022; 2024b), further improve long-range dependency modeling through point-specific attention mechanisms and increasingly efficient neighborhood organization. More recently, Mamba-based methods such as PCM Zhang et al. (2025b) organize irregular point clouds into structured sequences and exploit state-space models for efficient global representation learning. Despite their strong representation capability, these ANN-based approaches predominantly rely on dense floating-point computation, and increasingly sophisticated local and global modeling can still introduce considerable computational and energy costs. Therefore, reducing computational and energy overhead while preserving effective geometric modeling remains an important challenge for point cloud understanding.

SNN-based Point Cloud Understanding. Spiking neural networks have recently attracted increasing attention for point cloud understanding due to their sparse spike-based computation and potential energy efficiency. Spiking PointNet Ren et al. (2024) extends conventional point-based learning to the spiking domain and demonstrates the feasibility of directly processing point clouds with SNN. P2SResLNet Wu et al. (2024a) further introduces spatially aware spiking neurons and spiking residual structures to enhance the modeling of irregular 3D geometry. SPT Wu et al. (2025a) incorporates Transformer-style feature interaction into point-cloud SNN and improves geometric representation through dedicated sampling and spiking neuron designs. SPM Wu et al. (2025b) further combines hierarchical point encoding with spiking state-space modeling to achieve efficient local-global feature extraction. More recently, SVL Qiu et al. (2026) extends spiking point cloud representations toward vision-language learning and more general 3D understanding. Despite these advances, existing methods mainly focus on network-side spiking feature modeling, while the temporal organization of static 3D geometry and more geometry-aware global spike interactions remain less explored. Our work addresses these aspects by introducing viewpoint-induced temporal geometric representations together with discrepancy-based global spiking interaction for point cloud understanding.

3 PRELIMINARIES

Integrate-and-Fire Neuron. We adopt the Integrate-and-Fire (IF) neuron as the basic spiking unit. Given the input $X[t]$ at timestep t, its membrane potential and spike output are computed as

$$H[t] = V[t-1] + X[t],\ S[t] = \Theta(H[t] - V_{\text{th}}),\ V[t] = H[t](1 - S[t]) + V_{\text{reset}} S[t]. \quad (1,2,3)$$

Here, $\Theta(\cdot)$ denotes the Heaviside step function. Through temporal accumulation and threshold firing, the IF neuron converts continuous activations into binary spike sequences.

Spiking Self-Attention. Spikformer Zhou et al. (2022) introduces Spiking Self-Attention (SSA) to perform global interaction with binary spike features through spike-driven computation. Given $\mathbf{X} \in \mathbb{R}^{T \times N \times D}$, spike-form Query, Key, and Value are generated as

$$\mathbf{Q} = \text{SN}(\text{BN}(\mathbf{X}\mathbf{W}_Q)),\ \mathbf{K} = \text{SN}(\text{BN}(\mathbf{X}\mathbf{W}_K)),\ \mathbf{V} = \text{SN}(\text{BN}(\mathbf{X}\mathbf{W}_V)). \quad (4,5,6)$$

SSA removes the softmax operation in conventional attention and performs spike-driven interaction over the resulting binary features as

$$\text{SSA}'(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SN}\big(\mathbf{Q}\mathbf{K}^{\mathsf{T}}\mathbf{V} \cdot s\big). \quad (7)$$

where s is a scaling factor. Its interaction is primarily based on Query-Key agreement, which motivates our subsequent discrepancy-based modeling for 3D point clouds.

4 METHOD

4.1 OVERVIEW

Our framework combines geometric event encoding with local and global spiking feature modeling for point cloud analysis. Given an input point cloud $\mathcal{P}$, VGE generates an ordered geometric event representation through virtual viewpoint changes while preserving point-wise correspondence and introducing an additional temporal dimension, denoted as $\mathbf{F}_{\text{VGE}} = \text{VGE}(\mathcal{P})$. We then stack M feature learning blocks to progressively process local geometry and temporal event information, while VSSA aggregates global context. The overall process is formulated as

$$\mathbf{F}_{\text{local}} = \text{SpikingMLP}(\text{LocalFusion}(\mathbf{F}_{\text{VGE}})), \quad (8)$$

$$\mathbf{F}_{\text{global}} = \text{VSSA}(\mathbf{F}_{\text{local}} + \text{SpikingPE}(\mathbf{F}_{\text{VGE}})). \quad (9)$$

Here, $\mathbf{F}_{\text{VGE}}$ denotes the point representation obtained by VGE, which preserves the original point-wise spatial correspondence while incorporating geometric event information organized along the temporal dimension. LocalFusion aggregates neighborhood geometry and point features, while SpikingMLP further encodes local representations through linear mappings, batch normalization, and spiking neurons. In parallel, SpikingPE transforms the temporal event information into temporal spiking positional encodings, which are added to the local features. The fused features are processed by VSSA for global interactions driven by spike discrepancies, followed by residual connections for feature updates. Stacking these modules progressively integrates local geometry, temporal events, and global information to support various downstream point cloud understanding tasks.

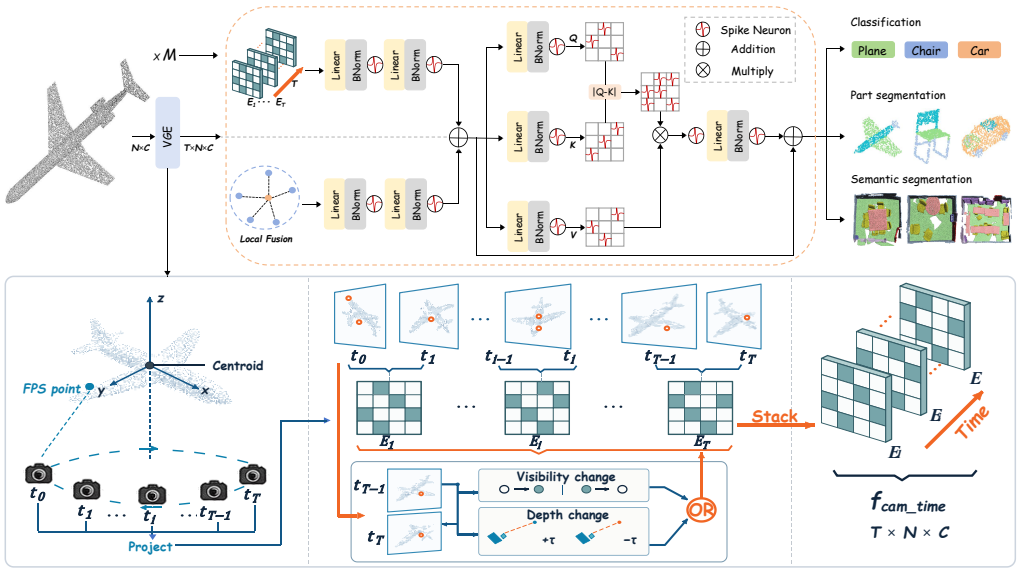

Figure 1: Overview of the proposed framework. VGE encodes a static point cloud into temporal geometric events through virtual viewpoint changes. Stacked blocks fuse local features with event-based positional encodings and model global spike discrepancies using VSSA for downstream point cloud understanding.

4.2 Viewpoint-Induced Geometric Events

A static point cloud contains rich spatial geometry but lacks explicit temporal cues, making it difficult to fully exploit the temporal dynamics of SNNs. To address this limitation, we introduce Viewpoint-Induced Geometric Events (VGE), which transforms static geometry into ordered temporal events through controlled virtual viewpoint changes. The resulting representation provides explicit geometric temporal cues for subsequent spiking feature modeling. Given a point cloud $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N}$, we select L anchors using farthest point sampling (FPS) and compute the centroid $\boldsymbol{\mu}$. Each anchor defines two virtual views: one camera located at the anchor and looking toward the centroid, and another located at the centroid and looking toward the anchor, yielding $C = 2L$ view channels. Let $\bar{\mathbf{o}}_c$ and $\mathbf{q}_c$ denote the reference position and fixed viewing target of camera c, respectively. Its position at timestep t is defined as

$$\mathbf{o}_c^t = \mathbf{q}_c + \mathbf{R}_z(\theta_t)\,(\bar{\mathbf{o}}_c - \mathbf{q}_c)\,, \qquad t = 0, \ldots, T. \tag{10}$$

Here, $\mathbf{R}_z(\theta_t)$ rotates the camera offset around the z-axis, while θ_t is sequentially sampled within a prescribed angular range. Each camera continuously points toward $\mathbf{q}_c$, producing $T+1$ ordered camera poses and therefore T consecutive event intervals. At each timestep, the point cloud is projected into every virtual camera to obtain the binary visibility $v_{i,c}^t$ and depth $d_{i,c}^t$ of point $\mathbf{p}_i$. We define two types of geometric events: visibility transitions and significant depth variations:

$$e_{\mathrm{vis},i,c}^t = v_{i,c}^{t-1} \oplus v_{i,c}^t, \tag{11}$$

$$e_{\mathrm{dep},i,c}^t = \left(v_{i,c}^{t-1} \wedge v_{i,c}^t\right) \wedge \mathbb{I}\left[\left|\log|d_{i,c}^t| - \log|d_{i,c}^{t-1}|\right| > \tau\right], \tag{12}$$

$$e_{i,c}^t = e_{\mathrm{vis},i,c}^t \vee e_{\mathrm{dep},i,c}^t, \qquad t = 1, \ldots, T. \tag{13}$$

Here, $\oplus$, $\wedge$, and $\vee$ denote logical XOR, AND, and OR, respectively; $\mathbb{I}[\cdot]$ is the indicator function, and τ is the depth-change threshold. The depth criterion is evaluated only when the point remains visible in both adjacent observations with valid nonzero depth. Visibility events capture points becoming visible or invisible under viewpoint changes, while depth events characterize significant relative depth variations of continuously visible points. Their logical union forms the final binary geometric event. At each event timestep, responses from all points and views form an event matrix $\mathbf{E}_t = [e_{i,c}^t] \in \{0,1\}^{N\times C}$. Stacking these matrices in temporal order yields

$$\mathbf{F}_{\mathrm{VGE}} = \mathrm{Stack}_{t=1}^{T}\left(\mathbf{E}_t\right) \in \{0,1\}^{T\times N\times C}. \tag{14}$$

The resulting representation preserves the correspondence between point indices and view channels while retaining the temporal order of event occurrence, enabling subsequent spiking processing to exploit geometric variations induced by controlled viewpoint motion.

4.3 VECTOR SUBTRACTIVE SPIKING ATTENTION

3D points are explicitly embedded in spatial geometry, where variations in spatial position and local structure naturally induce heterogeneous feature responses. After spiking encoding, such geometric variations can further manifest as discrepancies between binary firing patterns. Motivated by this property, we introduce Vector Subtractive Spiking Attention (VSSA) to aggregate global point context through vector-level spike discrepancies rather than relying solely on feature similarity. Let $\mathbf{X} \in \mathbb{R}^{T\times N\times D}$ denote the temporal point features. Independent linear projections, batch normalization, and spiking neurons are first applied to generate binary Query, Key, and Value representations:

$$\mathbf{Q} = \mathrm{SN}(\mathrm{BN}(\mathbf{X}\mathbf{W}_Q))\,,\ \mathbf{K}= \mathrm{SN}(\mathrm{BN}(\mathbf{X}\mathbf{W}_K))\,,\ \mathbf{V} = \mathrm{SN}(\mathrm{BN}(\mathbf{X}\mathbf{W}_V))\,. \tag{15,16,17}$$

The neuronal states evolve along the temporal axis, allowing geometric features from successive timesteps to influence the spike responses. For the i-th Query and the j-th Key, VSSA characterizes their relation using the element-wise absolute difference and directly uses this vector discrepancy to modulate the corresponding Value:

$$\mathbf{A}_{ij} = |\mathbf{q}_i - \mathbf{k}_j|\,,\ \mathbf{z}_i= \frac{1}{N}\sum_{j=1}^{N}(\mathbf{A}_{ij}\odot\mathbf{v}_j)\,. \tag{18,19}$$

Here, $\odot$ denotes element-wise multiplication. Unlike conventional attention that compresses a Query-Key relation into a scalar similarity score, VSSA preserves channel-wise spike discrepancies and directly associates them with the corresponding Value channels. For example, when $\mathbf{q}_i = [1, 0, 1, 0]$ and $\mathbf{k}_j = [1, 1, 0, 0]$, their element-wise absolute difference is $|\mathbf{q}_i-\mathbf{k}_j| = [0, 1, 1, 0]$, where entries equal to 1 directly indicate channels with mismatched spike states. The factor $1/N$ normalizes the aggregation over all points, and no softmax operation is required. Since Query and Key are binary spikes, their element-wise absolute difference is exactly equivalent to XOR:

$$|q - k| = q \oplus k = q + k - 2qk. \tag{20}$$

Thus, identical spike states produce zero discrepancy, whereas mismatched states produce one. Based on this property, we define two global statistics:

$$\overline{\mathbf{v}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{v}_j, \qquad \overline{\mathbf{r}}= \frac{1}{N}\sum_{j=1}^{N}(\mathbf{k}_j\odot\mathbf{v}_j)\,. \tag{21}$$

The original pairwise aggregation can then be exactly reformulated as

$$\mathbf{z}_i = \mathbf{q}_i\odot\overline{\mathbf{v}} + \overline{\mathbf{r}} - 2\mathbf{q}_i\odot\overline{\mathbf{r}}. \tag{22}$$

Therefore, the original discrepancy-driven interaction can be evaluated using two global statistics shared across all Queries, without explicitly constructing dense point-to-point discrepancy tensors. This exact factorization preserves the original binary forward computation while reducing the aggregation complexity from $\mathcal{O}(TN^2D)$ to $\mathcal{O}(TND)$. The detailed derivation and associated properties are provided in the ***Appendix***. After discrepancy aggregation, the resulting feature $\mathbf{Z}$ is transformed by a spiking neuron, followed by an output linear, batch normalization, and spiking neuron:

$$\mathbf{Y} = \mathrm{SN}(\mathrm{BN}(\mathrm{SN}(\mathbf{Z})\mathbf{W}))\,. \tag{23}$$

VSSA preserves the temporal axis throughout the attention process, allowing successive viewpoint-induced geometric events to continuously influence neuronal states before subsequent temporal readout, while combining vector-level spike discrepancy modeling for 3D geometric variations with exact binary-spike-induced linearization to achieve efficient global spiking attention without explicitly constructing dense point-to-point attention maps.

5 EXPERIMENTS

We evaluate VGESP on a range of point cloud understanding tasks, including object classification on ModelNet40 Wu et al. (2015) and the PB_T50_RS, OBJ_BG, and OBJ_ONLY variants of ScanObjectNN Uy et al. (2019), part segmentation on ShapeNetPart Yi et al. (2016), and semantic segmentation on S3DIS Armeni et al. (2016) and ScanNetv2 Dai et al. (2017). All experiments are implemented in PyTorch with SpikingJelly Fang et al. (2023) and conducted on two NVIDIA RTX 4090 GPUs with 24 GB memory each. Detailed dataset descriptions, training configurations, mathematical proofs, energy calculation and additional experiments are provided in the *Appendix*.

Table 1: Classification results on ScanObjectNN and ModelNet40. We report overall accuracy (OA, %) on three ScanObjectNN variants (BG: OBJ_BG, ONLY: OBJ_ONLY, RS: PB_T50_RS), and OA on ModelNet40. FLOPs are in Giga (G) and Energy in millijoule (mJ). Bold and underline denote the best and second-best results, respectively; '-' indicates results not reported in the original paper; '*' marks our reproduced results.

Methods	Type	Flops ↓	ModelNet40		ScanObjectNN			
			OA ↑	Energy ↓	BG (OA) ↑	ONLY (OA) ↑	RS (OA) ↑	Energy ↓
ACT Dong et al. (2023)[ICLR]	ANN	-	93.2	-	93.3	91.9	88.9	-
PTv3 Wu et al. (2024b)[CVPR]	ANN	18.9*	**94.5**	86.9*	90.7	88.4	87.9	86.9*
PointDiff Zheng et al. (2024)[CVPR]	ANN	-	94.2*	-	93.3	91.9	87.6	-
PointMamba Liang et al. (2024)[NIPS]	ANN	3.6	93.2	16.6	88.3	87.8	82.5	16.6
PointGPT Chen et al. (2024)[NIPS]	ANN	-	94.0	-	91.6	90.0	86.9	-
PCM Zhang et al. (2025b)[AAAI]	ANN	45.0	93.4	207.0	-	-	88.1	207.0
SIM Bahri et al. (2025)[CVPR]	ANN	3.6	92.7	16.6	92.3	91.4	87.3	16.6
GSPoint Yuan et al. (2026)[AAAI]	ANN	10.6	**94.5**	48.8	90.2*	89.6*	88.1	48.8
QPoint Zhou et al. (2026a)[ICML]	ANN	**0.9**	94.2	**4.2**	**93.8**	**92.6**	**90.7**	**4.2**
P2SResLNetWu et al. (2024a)AAAI	SNN	3.3	88.7	3.0	78.6	80.2	81.0	3.2
SPTWu et al. (2025a)[AAAI]	SNN	14.0	91.4	13.3	82.8	83.4	78.0	14.6
E-3DSNNQiu et al. (2025)[AAAI]	SNN	2.1	91.7	1.8	-	-	83.4	2.6
SPMWu et al. (2025b)[ICCV]	SNN	6.1	92.3	5.4	<u>90.2</u>	<u>89.5</u>	84.2	7.6
SDT Lu et al. (2026b)[ICLR]	SNN	<u>1.5</u>	92.5	<u>1.3</u>	-	-	<u>86.2</u>	2.1
SVL Qiu et al. (2026)[ICML]	SNN	5.7	92.6	5.1	-	-	82.1	-
SpikeNet Zhou et al. (2026b)[ICML]	SNN	1.9	93.4	1.7	88.8*	87.2*	85.5	<u>1.8</u>
VGESP (ours)	SNN	**1.4**	**93.6**	**1.2**	**91.2**	**90.9**	**87.8**	**1.3**

5.1 Comparison with State-of-the-art Methods

We compare VGESP with representative ANN and SNN methods on point cloud understanding tasks. For classification, we report overall accuracy (OA) on ModelNet40 and three ScanObjectNN variants, including OBJ_BG, OBJ_ONLY, and PB_T50_RS. For segmentation, we evaluate category and instance mIoU on ShapeNetPart, and report mean intersection-over-union (mIoU) on S3DIS and ScanNetV2. In addition to recognition performance, we further report FLOPs and estimated energy consumption to evaluate the trade-off between representation capability and energy efficiency.

Shape Classification on ModelNet40. ModelNet40 is a widely used synthetic benchmark for 3D object classification, which evaluates the ability of models to learn 3D object features from point clouds. As shown in Table 1, VGESP achieves an OA of 93.6%, establishing the state-of-the-art performance among SNN-based point cloud methods. Compared with the previous best SNN method SpikeNet (93.4%), VGESP improves the accuracy by 0.2%. This result indicates that geometry-aware temporal representations provide complementary information beyond conventional spike-based feature encoding and further enhance the modeling capability of SNNs for static 3D structures. Meanwhile, VGESP requires only 1.4G FLOPs and 1.2 mJ energy consumption. Compared with the ANN-based QPoint, VGESP achieves comparable classification performance with only a 0.6% accuracy gap, while reducing energy consumption from 4.2 mJ to 1.2 mJ. This result highlights the energy efficiency advantage of spike-driven computation, enabling SNNs to approach the recognition capability of ANN models with lower energy costs while achieving a favorable accuracy-efficiency balance on clean synthetic object shapes.

Shape Classification on ScanObjectNN. ScanObjectNN is a real-world benchmark for evaluating point cloud classification, which contains three commonly used variants: OBJ_ONLY, containing only object points; OBJ_BG, introducing surrounding background points; and PB_T50_RS, further applying perturbations and random rotations to evaluate robustness under complex conditions. As reported in Table 1, VGESP achieves 91.2%, 90.9%, and 87.8% OA on OBJ_BG, OBJ_ONLY, and PB_T50_RS, respectively, achieving the best performance among SNN-based methods across all settings. Compared with the strongest SNN counterparts, VGESP improves by 1.0%, 1.4%, and 1.6%, respectively. The consistent improvements under different testing conditions demonstrate that VGESP can effectively capture geometric variations in complex point clouds and enhance the robustness of SNN-based point cloud understanding. Furthermore, VGESP requires only 1.4G FLOPs and 1.3 mJ energy consumption. Compared with the ANN-based QPoint, VGESP maintains competitive classification performance while reducing energy consumption from 4.2 mJ to 1.3 mJ. This result further demonstrates the energy efficiency advantage of spike-based computation for complex point cloud processing, allowing SNNs to achieve realistic 3D understanding with a more efficient computational paradigm and providing an effective solution for low-power 3D perception.

Table 2: Semantic Segmentation Performance Comparison on S3DIS Dataset.

Method	Type	Flops	Energy	mIoU	ceil.	floor	wall	beam	colu.	wind.	door	table	chair	sofa	book.	board	clut.
PointNet Qi et al. (2017a)[CVPR]	ANN	1.2	5.5	41.1	88.8	97.3	69.8	0.0	3.9	46.3	10.8	59.0	52.6	5.9	40.3	26.4	33.2
PointNet++ Qi et al. (2017b)[NIPS]	ANN	4.9	22.5	53.5	89.4	97.7	75.4	0.0	1.8	58.3	19.5	79.0	69.2	59.1	46.2	58.7	41.6
KPConv Thomas et al. (2019)[ICCV]	ANN	29.7	136.5	67.1	92.8	97.3	82.4	0.0	23.9	58.0	69.0	81.5	91.0	75.4	75.3	66.7	58.9
PointNeXt Qian et al. (2022)[NIPS]	ANN	-	-	70.5	**94.2**	**98.5**	84.4	0.0	37.7	59.3	74.0	83.1	91.6	77.4	77.2	78.8	60.6
PCM Zhang et al. (2025b)[AAAI]	ANN	-	-	63.4	93.3	96.7	80.6	0.0	35.9	57.7	60.0	74.0	87.6	50.1	69.4	63.5	55.9
PointRWKV He et al. (2025)[AAAI]	ANN	-	-	70.5	**94.2**	98.3	86.5	0.0	38.6	**64.5**	76.2	88.2	89.3	65.2	75.6	78.2	61.3
QPoint Zhou et al. (2026a)[ICML]	ANN	9.4	43.2	71.2	90.6	96.9	84.8	0.0	48.9	61.8	75.6	86.8	89.2	77.6	78.4	77.9	59.9
PTv1 Zhao et al. (2021)[ICCV]	ANN	16.7	76.8	70.4	94.0	**98.5**	86.3	0.0	38.0	63.4	74.3	**89.1**	82.4	74.3	**80.2**	76.0	59.3
PTv2 Wu et al. (2022)[NIPS]	ANN	86.9	400.1	71.6	93.0	98.1	**86.7**	0.0	48.0	62.4	76.1	88.3	87.6	77.1	79.2	77.5	59.8
PTv3 Wu et al. (2024b)[CVPR]	ANN	149.3	687.7	**73.6**	92.4	98.3	86.6	0.0	**55.8**	63.7	**77.1**	83.8	**93.3**	**79.1**	79.4	**85.4**	**61.7**
E-3DSNNQiu et al. (2025)[AAAI]	SNN	16.1	14.4	67.4	**95.3**	98.5	82.3	0.0	28.0	55.8	71.5	**81.2**	89.8	69.2	**76.4**	67.0	**61.6**
SpikeNet Zhou et al. (2026b)[ICML]	SNN	11.3	10.2	68.9	90.1	92.2	83.5	0.0	**47.6**	60.5	**75.2**	80.1	86.3	70.3	72.4	80.6	57.3
SDT Lu et al. (2026b)[ICLR]	SNN	8.1	7.3	69.6	93.8	98.5	84.8	0.0	42.1	57.0	69.7	77.6	91.3	76.1	74.1	79.3	59.9
VGESP (ours)	SNN	**3.4**	**3.1**	**71.2**	94.2	**98.6**	**86.3**	0.0	44.5	**60.6**	72.0	79.8	**92.4**	**78.3**	75.5	**81.4**	61.5

Table 3: Part Segmentation Performance Comparison on ShapeNet Part Dataset.

Methods	Type	Cat.mIoU	Ins.mIoU	Energy
PointMamba Liang et al. (2024)[NIPS]	ANN	84.4	86.0	65.8
PointGPT Chen et al. (2024)[NeurIPS]	ANN	84.1	86.2	**26.2**
PCM Zhang et al. (2025b)[AAAI]	ANN	**85.3**	87.0	207.0
SIM Bahri et al. (2025)[CVPR]	ANN	84.1	85.9	75.8
QPoint Zhou et al. (2026a)[ICML]	ANN	85.1	87.0	43.2
PTv1 Zhao et al. (2021)[ICCV]	ANN	83.6	86.6	127.0
PTv2 Wu et al. (2022)[NIPS]	ANN	84.0	86.7	-
PTv3 Wu et al. (2024b)[CVPR]	ANN	84.8	**87.1**	-
E-3DSNNQiu et al. (2025)[AAAI]	SNN	81.7	83.8	8.8
SDT Lu et al. (2026b)[ICLR]	SNN	83.7	85.1	4.7
SpikeNet Zhou et al. (2026b)[ICML]	SNN	83.9	85.4	5.4
VGESP (ours)	SNN	**84.1**	**86.0**	**3.4**

Table 4: Semantic Segmentation Performance Comparison on ScanNet V2 Dataset.

Methods	Type	Val.mIoU	Test.mIoU	Energy
OctFormer Wang (2023)[TOG]	ANN	74.5	-	259.9
Swin3D Yang et al. (2023)[ICCV]	ANN	76.4	-	40.5
PointMamba Liang et al. (2024)[NIPS]	ANN	74.6	-	253.5
OneFormer3D Kolo et al. (2024)[CVPR]	ANN	-	76.6	-
PmabaLi et al. (2025)[AAAI]	ANN	-	77.6	**22.1**
CamPoint Zhang et al. (2025a)[CVPR]	ANN	**77.7**	77.5	49.7
PTv1 Zhao et al. (2021)[ICCV]	ANN	70.6	-	-
PTv2 Wu et al. (2022)[NIPS]	ANN	75.4	74.2	59.8
PTv3 Wu et al. (2024b)[CVPR]	ANN	77.5	**77.9**	-
E-3DSNNQiu et al. (2025)[AAAI]	SNN	68.2	69.5	-
S3DNetLu et al. (2026a)[NN]	SNN	68.7	70.2	-
SpikeNetZhou et al. (2026b)[ICML]	SNN	69.7	71.2	6.7
VGESP (ours)	SNN	**71.1**	**72.0**	**3.1**

Semantic Segmentation on S3DIS. S3DIS is a widely used benchmark for indoor semantic segmentation, containing large-scale point cloud scenes with multiple semantic categories. As shown in Table 2, VGESP achieves 71.2% mIoU on S3DIS, obtaining the best performance among SNN-based methods. Compared with the previous best SNN method SDT (69.6% mIoU), VGESP improves by 1.6%. The performance gain indicates that VGESP can better adapt to semantic variations in complex indoor scenes. In terms of efficiency, VGESP requires only 3.4G FLOPs and 3.1 mJ energy consumption. Compared with the ANN-based QPoint (43.2 mJ), VGESP achieves similar segmentation accuracy with substantially lower energy consumption, demonstrating the efficiency advantage of spike-based computation for complex 3D scenes. Additionally, Figure 2 shows the qualitative segmentation results of VGESP on S3DIS, where the predictions exhibit consistent spatial layouts compared with the ground-truth labels, further demonstrating its effectiveness in indoor scene understanding with reliable prediction consistency.

Part Segmentation on ShapeNetPart. ShapeNetPart is a widely used benchmark for 3D part segmentation, which evaluates the ability of models to understand fine-grained object structures. As shown in Table 3, VGESP achieves 84.1% Cat.mIoU and 86.0% Ins.mIoU on ShapeNetPart, obtaining the best performance among SNN-based methods. Compared with the previous best SNN method SpikeNet (83.9% Cat.mIoU and 85.4% Ins.mIoU), VGESP improves by 0.2% and 0.6%, respectively. These improvements highlight the effectiveness of VGESP in modeling fine-grained object structures. Meanwhile, VGESP achieves an energy consumption of only 3.4 mJ. Compared with the ANN-based QPoint (43.2 mJ), VGESP maintains competitive segmentation performance while significantly reducing energy consumption. This result demonstrates the favorable energy efficiency of VGESP for efficient fine-grained 3D analysis tasks.

Semantic Segmentation on ScanNet V2. ScanNetV2 is a large-scale real-world indoor scanning benchmark for evaluating semantic segmentation in realistic scenes. As shown in Table 4, VGESP achieves 71.1% Val.mIoU and 72.0% Test.mIoU on ScanNetV2, obtaining the best performance among SNN-based methods. Compared with the previous best SNN method SpikeNet (69.7% Val.mIoU and 71.2% Test.mIoU), VGESP improves by 1.4% and 0.8%, respectively. This result further verifies the generalization capability of VGESP on real-world scanned scenes. Furthermore, VGESP consumes only 3.1 mJ of theoretical inference energy, which is substantially lower than the ANN-based CamPoint (49.7 mJ). Combined with its performance advantage over existing SNN methods, VGESP demonstrates a favorable balance between segmentation accuracy and energy efficiency on large-scale real-world scenes. Figure 2 further presents the segmentation results of VGESP on ScanNetV2, where the model preserves accurate semantic predictions and clear object-level boundaries in real-world scanned scenes.

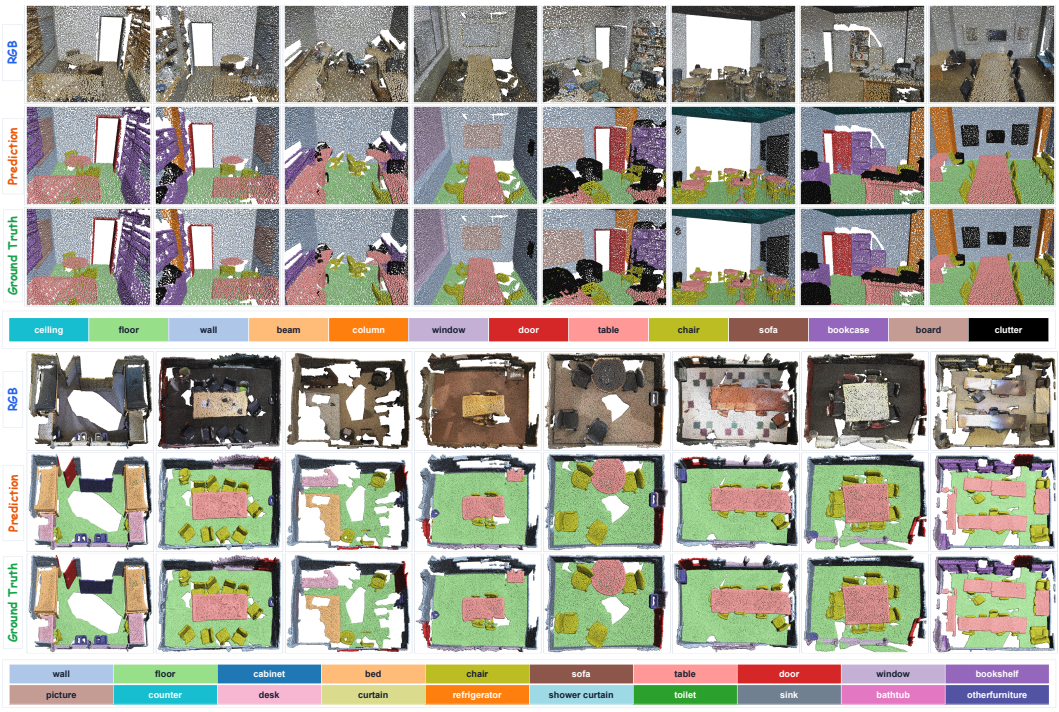

Figure 2: Qualitative semantic segmentation results. The upper parts show results on S3DIS and lower parts show results on ScanNetV2, respectively. For each example, the columns from left to right represent the input RGB point cloud, ground-truth labels, and predicted segmentation results.

5.2 Ablation Studies

Model Efficiency. Model size and runtime efficiency are important factors for evaluating the practicality of SNN-based methods. To evaluate the computational efficiency of VGESP, we compare the parameter count as well as the training and testing time of different methods. The evaluation considers both model size and runtime cost. As shown in Table 5, VGESP contains only 9.7M parameters, making it the smallest model among all compared methods. It also achieves the shortest testing time of 1.1 h. Compared with the SNN-based SpikeNet, VGESP reduces the parameter count from 18.2M to 9.7M and shortens the testing time from 1.4 h to 1.1 h. Its training time is also reduced from 113.5 h to 103.3 h. Compared with the ANN-based PointMamba, VGESP reduces the parameter count from 32.1M to 9.7M and the testing time from 1.9 h to 1.1 h. Although PointMamba requires less training time, VGESP shows clear advantages in model size and inference efficiency. These results highlight the computational efficiency of VGESP with a compact model design.

Architecture Ablation. To further analyze the sources of performance gains in VGESP, we conduct ablation studies on its core components on S3DIS. The baseline directly encodes the static input point cloud into spikes and uses only simple Spiking MLP layers for feature learning, without VGE or any global attention module. Thus, the baseline mainly relies on static spatial geometric features. Based on this baseline, we introduce VGE and VSSA separately to evaluate their individual contributions. We further combine VGE with standard SSA as a reference for comparison with VSSA. The full VGESP integrates both VGE and VSSA. As shown in Table 6, the baseline achieves 66.3% mIoU. Adding VGE increases the mIoU to 68.4%, yielding a 2.1% gain. Compared with direct static spike encoding, VGE organizes visibility and depth changes across consecutive virtual viewpoints into explicit temporal geometric events, validating the effectiveness of introducing temporal geometric representations for static point clouds. Introducing VSSA alone reaches 69.1% mIoU, improving the baseline by 2.8%. Combining VGE with standard SSA achieves 70.4% mIoU. Replacing SSA with VSSA allows the full VGESP to reach 91.0% OA, 77.1% mAcc, and 71.2% mIoU. Compared with VGE + SSA, the three metrics improve by 0.7%, 0.9%, and 0.8%, respectively, while the full model improves mIoU by 4.9% over the baseline. These results show that VGE and VSSA are complementary, jointly enhancing temporal geometric representation and global feature interaction.

Table 5: Model Efficiency on ScanNetV2.

Methods	P(M)	Test(h)	Train(h)
OctFormer Wang (2023)	44.1	2.4	109.8
PointMamba Liang et al. (2024)	32.1	1.9	**76.7**
PTv3 Wu et al. (2024b)	46.2	1.8	116.8
E-3DSNNQiu et al. (2025)	13.6	1.3	110.4
SpikeNet Zhou et al. (2026b)	18.2	1.4	113.5
VGESP (ours)	**9.7**	**1.1**	103.3

Table 6: Analysis of Architecture Variants on S3DIS.

Methods	OA(%)	mAcc(%)	mIoU(%)
Baseline	86.9	73.1	66.3
+ VGE	88.6	75.0	68.4
+ VSSA	89.6	75.3	69.1
+ VGE + SSA	90.3	76.2	70.4
Full VGESP	**91.0**	**77.1**	**71.2**

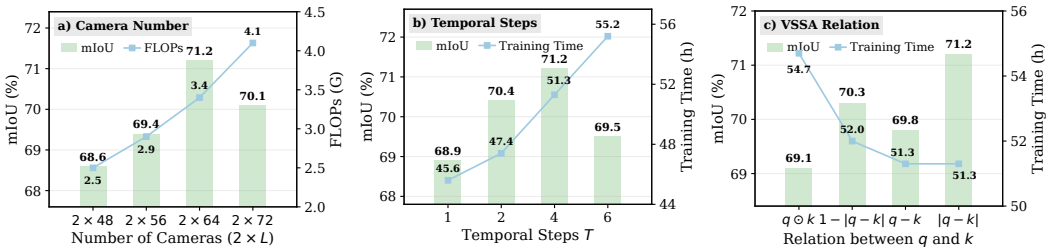

Figure 3: Ablation analysis of VGESP on camera number, temporal steps, and VSSA relations on S3DIS.

Number of cameras. To analyze the effect of the number of virtual cameras on geometric event modeling in VGE, we vary the number of anchors L. Each anchor defines two opposite virtual viewing directions, resulting in $2L$ cameras in total. As shown in Figure. 3(a), increasing the camera number from 2×48 to 2×64 improves the mIoU from 68.6% to 71.2%, indicating that richer viewpoints provide more informative geometric events. More viewpoints can capture richer geometric variations. Further increasing the camera number to 2×72 decreases the mIoU to 70.1%, while the FLOPs increase from 3.4G to 4.1G, suggesting that excessive viewpoints may introduce repetitive geometric responses and additional computation.

Temporal event intervals. In VGE, the virtual camera produces $T + 1$ consecutive viewpoint states along a short trajectory, from which T geometric event intervals are generated according to visibility and depth changes between adjacent viewpoints. We further vary T to study the effect of event sequence length. As shown in Figure. 3(b), increasing T from 1 to 4 improves the mIoU from 68.9% to 71.2%, showing that moderate viewpoint transitions provide richer geometric temporal information. Further increasing T from 4 to 6 decreases the mIoU to 71.2% and 69.5%, while the training time rises to 51.3 h and 55.2 h, respectively. Longer trajectories may introduce redundant events and additional computation along the temporal dimension.

VSSA relations analysis. To examine the effectiveness of vector discrepancy modeling in VSSA, we compare four different q-k relations. As shown in Figure. 3(c), the element-wise product $q \odot k$ achieves 69.1% mIoU, while the XNOR form $1 - |q - k|$ and signed difference $q - k$ obtain 70.3% and 69.8%, respectively. In contrast, the absolute difference $|q - k|$ used in VSSA achieves the best mIoU of 71.2%. The relations $q \odot k$ and $1 - |q - k|$ mainly characterize common activation or similarity, while $q - k$ preserves the signed direction of the difference. In comparison, $|q - k|$ directly captures discrepancies between binary spike responses, which better matches the modeling objective of VSSA. Moreover, $|q - k|$ requires 51.3 h of training, identical to $q - k$ and lower than the other two relations, without introducing additional training cost.

6 Conclusion

We propose VGESP, a spiking neural network framework for efficient point cloud understanding. To address the lack of explicit temporal information in static point clouds, we introduce VGE, which converts spatial geometry into ordered sparse events through controlled virtual viewpoint changes, providing explicit geometric temporal features for SNNs. We further propose VSSA, which performs global interaction through vector-level discrepancies between binary spike features and avoids dense point-to-point attention maps through equivalent reformulation. Experiments on classification and segmentation benchmarks show that VGESP achieves the best performance among SNN-based methods while maintaining low energy consumption and high computational efficiency, providing an effective solution for high-performance and low-power 3D point cloud understanding.

REFERENCES

Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1534–1543, 2016.

Ali Bahri, Moslem Yazdanpanah, Mehrdad Noori, Sahar Dastani, Milad Cheraghalikhani, Gustavo Adolfo Vargas Hakim, David Osowiechi, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Spectral informed mamba for robust point cloud processing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11799–11809, 2025.

Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. *Advances in Neural Information Processing Systems*, 36, 2024.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8Oun8ZUVe8N.

Wei Fang, Yanqi Chen, Jianhao Ding, Zhaofei Yu, Timothée Masquelier, Ding Chen, Liwei Huang, Huihui Zhou, Guoqi Li, and Yonghong Tian. Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9(40):eadi1480, 2023.

Seongheon Ha, Yeogyeong Kim, and Jinsun Park. Interdimensional knowledge transfer for semantic segmentation on lidar point clouds. *IEEE Robotics and Automation Letters*, 2024.

Qingdong He, Jiangning Zhang, Jinlong Peng, Haoyang He, Xiangtai Li, Yabiao Wang, and Chengjie Wang. Pointrwkv: Efficient rwkv-like model for hierarchical point cloud learning. *arXiv preprint arXiv:2405.15214*, 2025.

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.

Maxim Kolo, diazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20943–20953. IEEE, 2024.

Souvik Kundu, Massoud Pedram, and Peter A Beerel. Hire-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5209–5218, 2021.

Zhuoyuan Li, Yubo Ai, Jiahao Lu, ChuXin Wang, Jiacheng Deng, Hanzhi Chang, Yanzhe Liang, Wenfei Yang, Shifeng Zhang, and Tianzhu Zhang. Pamba: Enhancing global interaction in point clouds via state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5092–5100, 2025.

Dingkang Liang, Xin Zhou, Xinyu Wang, Xingkui Zhu, Wei Xu, Zhikang Zou, Xiaoqing Ye, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024.

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*, 195:108190, 2026a. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2025.108190. URL https://www.sciencedirect.com/science/article/pii/S0893608025010706.

Yijie Lu, Zhiyi Pan, Renrui Zhang, Yanhao Jia, Ronggang Wang, and Zhaokun Zhou. Spiking discrepancy transformer for point cloud analysis. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=7Brnh0aNFn.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022.

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

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.

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022.

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*, pp. 20086–20094, 2025.

Xuerui Qiu, Shaowei Gu, Peixi Wu, JiaKui Hu, Yaozhi Wen, Yuqi Pan, Xinhao Luo, Bo XU, and Guoqi Li. SVL: Empowering spiking neural networks for efficient 3d open-world understanding. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Ai3a79cTvr.

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, 2024.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE/CVF ICCV*, pp. 945–953, 2015.

Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6411–6420, 2019.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1588–1597, 2019.

Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions on Graphics (TOG)*, 42(4):1–11, 2023.

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*, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26393–26403, 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*, pp. 6092–6099, 2024a.

Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022.

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 *CVPR*, 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.

Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding, 2023.

Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

Shangbo Yuan, Jie Xu, Ping Hu, Xiaofeng Zhu, and Na Zhao. Graph smoothing for enhanced local geometry learning in point cloud analysis, 2026. URL https://arxiv.org/abs/2601.11102.

Jianhui Zhang, Yizhi Luo, Zicheng Zhang, Xuecheng Nie, and Bonan Li. Campoint: Boosting point cloud segmentation with virtual camera. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 11822–11832, June 2025a.

Tao Zhang, Haobo Yuan, Lu Qi, Jiangning Zhang, Qianyu Zhou, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Point cloud mamba: Point cloud learning via state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10121–10130, 2025b.

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.

Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22935–22945, 2024.

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

Zhiming Zhou, Quanxiao Zhang, Qiaoyun Wu, and Cheng Yi. Spikepoint: Efficient 3d point cloud classification with point-to-pixel conversion and spike-transresnet. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pp. 425–439. Springer, 2025.

Zhiming Zhou, Yong He, Chaoxu Mu, Qiaoyun Wu, and Ajmal Saeed Mian. QPoint: End-to-end lightweight point cloud processing via robust quaternion feature learning. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=xxifbmZe0b.

Zhiming Zhou, Yong He, Chaoxu Mu, Qiaoyun Wu, and Ajmal Saeed Mian. Spikenet: Sparse spike-driven mask vector transformer for energy-efficient and stable spiking point cloud processing. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=7BpcmBjQL0.

Zhiming Zhou, Yong He, Qiaoyun Wu, Chaoxu Mu, and Ajmal Saeed Mian. 3DSMT: A hybrid spiking mamba-transformer for point cloud analysis. In *The Fourteenth International Conference on Learning Representations*, 2026c.

A APPENDIX

The supplementary material includes sections that strengthen the main manuscript:

A1. Theoretical Analysis.

A2. Dataset Introduction.

A3. Experiment Details.

A4. Energy Calculation.

A5. More Experiment.

A1 THEORETICAL ANALYSIS

A1.1 GEOMETRIC INTERPRETATION AND PROPERTIES OF VIEWPOINT-INDUCED GEOMETRIC EVENTS

To further clarify the geometric meaning of Viewpoint-Induced Geometric Events (VGE), we analyze its event generation process from the perspective of continuous viewpoint variation. We first show that the depth variation between adjacent virtual viewpoints locally approximates the geometric variation rate along the camera trajectory. We then analyze the complementary roles of visibility and depth events, the sparsity of the resulting event representation, and the preservation of pointwise correspondence. These properties show that VGE does not simply attach an artificial temporal dimension to a static point cloud, but converts viewpoint-induced geometric responses into ordered sparse events.

A1.1.1 VIEWPOINT-INDUCED DEPTH VARIATION

For the c-th virtual camera, its position along a short trajectory can be expressed as

$$\mathbf{o}_c(\theta) = \mathbf{q}_c + \mathbf{R}_z(\theta)(\bar{\mathbf{o}}_c - \mathbf{q}_c), \tag{1}$$

where $\bar{\mathbf{o}}_c$ denotes the reference camera position, $\mathbf{q}_c$ is the observation target, and θ parameterizes the camera trajectory. Sampling the trajectory at $T+1$ consecutive viewpoint states produces $\{\theta_0, \theta_1, \ldots, \theta_T\}$, from which T event intervals are generated between adjacent viewpoints. For point $\mathbf{p}_i$, let $d_{i,c}(\theta)$ denote its depth under camera c at viewpoint θ. When the point remains visible in two adjacent viewpoints and its depth is nonzero, we define

$$g_{i,c}(\theta) = \log |d_{i,c}(\theta)|. \tag{2}$$

The log-depth variation between two adjacent viewpoints is

$$\Delta_{i,c}^t = g_{i,c}(\theta_t) - g_{i,c}(\theta_{t-1}). \tag{3}$$

Let

$$\Delta\theta_t = \theta_t - \theta_{t-1}. \tag{4}$$

When the viewpoint displacement is sufficiently small, a first-order Taylor expansion of $g_{i,c}(\theta_t)$ around θ_{t-1} gives

$$g_{i,c}(\theta_t) = g_{i,c}(\theta_{t-1}) + g'_{i,c}(\theta_{t-1})\Delta\theta_t + \mathcal{O}(\Delta\theta_t^2). \tag{5}$$

Therefore,

$$\Delta_{i,c}^t = \left.\frac{\partial \log |d_{i,c}(\theta)|}{\partial\theta}\right|_{\theta=\theta_{t-1}} \Delta\theta_t + \mathcal{O}(\Delta\theta_t^2). \tag{6}$$

For sufficiently small $\Delta\theta_t$,

$$|\Delta_{i,c}^t| \approx \left|\frac{\partial \log |d_{i,c}|}{\partial\theta}\right| |\Delta\theta_t|. \tag{7}$$

The depth event in VGE is defined as

$$e_{\mathrm{dep},i,c}^t = (v_{i,c}^{t-1} \wedge v_{i,c}^t) \wedge \mathbb{I}\left[|\Delta_{i,c}^t| > \tau\right], \tag{8}$$

where $v_{i,c}^t \in \{0,1\}$ denotes the visibility state and τ is the depth-change threshold. Using equation 7, the depth-event condition can be approximately interpreted as

$$\left|\frac{\partial \log |d_{i,c}|}{\partial \theta}\right| > \frac{\tau}{|\Delta\theta_t|}. \tag{9}$$

Thus, the depth event responds to points whose geometry exhibits significant viewpoint sensitivity along the virtual camera trajectory. Rather than repeatedly encoding static coordinates, VGE captures geometric responses induced by controlled viewpoint changes. The logarithmic depth difference also has an intuitive interpretation. For a small depth perturbation Δd with $d > 0$,

$$\log(d + \Delta d) - \log d \approx \frac{\Delta d}{d}. \tag{10}$$

Hence, the log-depth difference approximately measures relative rather than absolute depth variation, reducing the influence of depth scale on event generation.

A1.1.2 Complementarity of Visibility and Depth Events

Continuous depth variation alone cannot fully characterize viewpoint-induced geometric changes. As the virtual camera moves, occlusion relationships may cause points to appear or disappear, producing discrete visibility transitions. VGE detects these transitions using

$$e_{\mathrm{vis},i,c}^t = v_{i,c}^{t-1} \oplus v_{i,c}^t. \tag{11}$$

Since $v_{i,c}^t \in \{0,1\}$,

$$v_{i,c}^{t-1} \oplus v_{i,c}^t = |v_{i,c}^t - v_{i,c}^{t-1}|. \tag{12}$$

Therefore, the visibility event represents a discrete change in visibility state between adjacent viewpoints. Whenever a point changes from visible to invisible or from invisible to visible, $e_{\mathrm{vis},i,c}^t = 1$. In contrast, the depth event is activated only when the point remains visible in both adjacent viewpoints and exhibits a sufficiently large depth variation. The two event types therefore capture complementary geometric responses:

$$\text{VGE} = \text{Visibility Discontinuity} + \text{Continuous Depth Variation}. \tag{13}$$

The final event is obtained by

$$e_{i,c}^t = e_{\mathrm{vis},i,c}^t \vee e_{\mathrm{dep},i,c}^t. \tag{14}$$

The visibility event captures discrete geometric changes mainly associated with occlusion and appearance transitions, whereas the depth event complements it with continuous viewpoint-dependent geometric variation in regions that remain visible.

A1.1.3 Event Sparsity under Geometric Stability

An important property of VGE is that not every point generates an event at every temporal interval. Suppose a point maintains the same visibility state between two adjacent viewpoints,

$$v_{i,c}^{t-1} = v_{i,c}^t, \tag{15}$$

and its log-depth variation satisfies

$$|\log |d_{i,c}^t| - \log |d_{i,c}^{t-1}|| \le \tau. \tag{16}$$

Under these conditions,

$$e_{\mathrm{vis},i,c}^t = 0, \qquad e_{\mathrm{dep},i,c}^t = 0, \tag{17}$$

and therefore

$$e_{i,c}^t = 0. \tag{18}$$

Let

$$\mathcal{S}_{t,c} = \{i \mid e_{i,c}^t = 1\} \tag{19}$$

denote the set of points generating events at interval t under camera c. Then

$$\mathcal{S}_{t,c} \subseteq \{i \mid v_{i,c}^{t-1} \neq v_{i,c}^t\} \cup \{i \mid |\Delta_{i,c}^t| > \tau\}. \tag{20}$$

Thus, only points undergoing visibility transitions or significant depth variations generate events, while geometrically stable points remain inactive. This selective activation converts continuous viewpoint changes into sparse binary events and emphasizes regions with informative geometric variation.

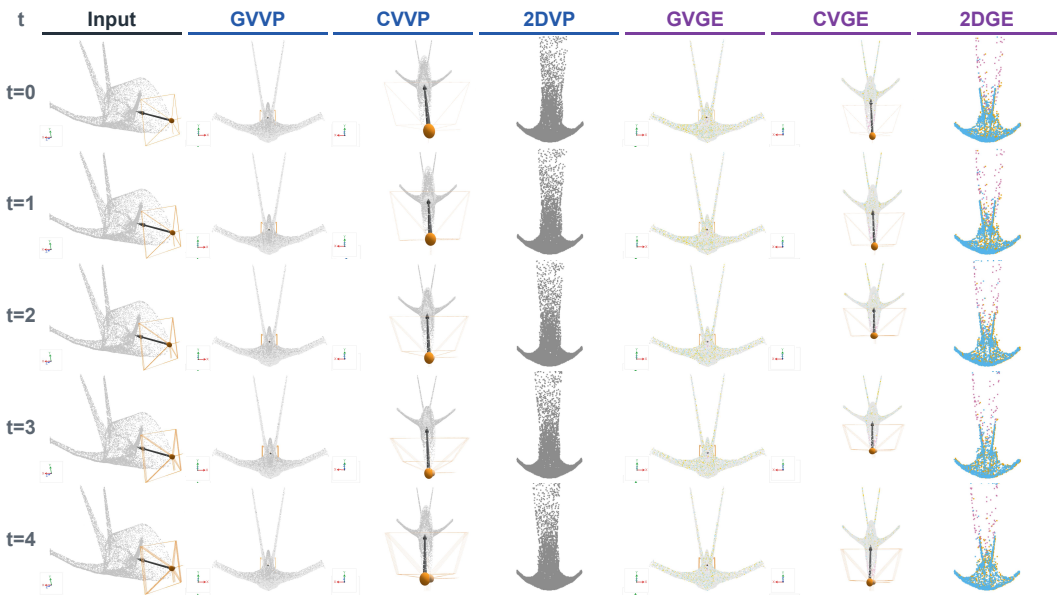

Figure A1: Visualization of Semantic Segmentation on ScanNetV2.

A1.1.4 Ordered Event Representation and Point-Wise Correspondence

VGE constructs T event intervals from $T+1$ consecutive viewpoint states. For each interval t, the event matrix is

$$\mathbf{E}_t = [e_{i,c}^t] \in \{0,1\}^{N\times C}, \tag{21}$$

where N is the number of points and $C = 2L$ denotes the number of virtual cameras generated from L anchors with two opposite viewing directions. Stacking all event matrices along the temporal dimension gives

$$\mathbf{F}_{\mathrm{VGE}} = \mathrm{Stack}_{t=1}^{T}(\mathbf{E}_t) \in \{0,1\}^{T\times N\times C}. \tag{22}$$

VGE changes only the event representation associated with each point and does not resample or reorder the original point set. Therefore, for every original point $\mathbf{p}_i$,

$$\mathbf{p}_i \longleftrightarrow \mathbf{F}_{\mathrm{VGE}}[:, i, :]. \tag{23}$$

This point-wise correspondence allows the temporal event representation to be directly combined with subsequent local neighborhood modeling and point-level prediction. The temporal dimension in $\mathbf{F}_{\mathrm{VGE}}$ is also explicitly ordered by the virtual camera trajectory. Each event interval is determined by the geometric change between two consecutive viewpoint states. Therefore, VGE differs from an unordered stacking of independent multi-view features: it encodes geometric variation along an ordered viewpoint trajectory.

A1.1.5 Visualization

To intuitively analyze the viewpoint-dependent geometric changes induced by virtual camera motion, we provide the visualization shown in Figure A1. The current camera is highlighted by an orange frustum, while the other rotated cameras are shown in light orange. We first select anchors using farthest point sampling and rotate the virtual cameras over a short trajectory around the global z-axis, allowing the static point cloud to exhibit dynamic observations across consecutive viewpoints. At different time steps, we visualize the visible points in the global view, camera view, and 2D projection (GVVP, CVVP, and 2DVP), together with the corresponding geometric event points (GVGE, CVGE, and 2DGE). Blue, orange, purple, and green denote transitions from invisible to visible, visible to invisible, increasing depth, and decreasing depth, respectively. For adjacent time steps $t \rightarrow t+1$, VGE module detects appearance, disappearance, moving-away, and approaching events, thereby converting static geometry into a temporally ordered event sequence. These dynamic events further characterize point responses to viewpoint changes, enabling the network to distinguish spatially nearby points with different viewpoint responses and providing temporally ordered geometric context for subsequent feature aggregation.

A1.2 Mathematical Derivation of Vector Subtractive Spiking Attention

To further clarify the mechanism and computational efficiency of Vector Subtractive Spiking Attention (VSSA), we provide a detailed mathematical derivation of its vector-level spike discrepancy and exact factorization. Specifically, we first show that the absolute difference between binary Query and Key features is equivalent to the XOR operation, and then demonstrate that the resulting pairwise interaction can be exactly reformulated without explicitly constructing dense point-to-point relations. This reformulation reduces the computational complexity with respect to the number of points from quadratic to linear.

A1.2.1 Vector-Level Spike Discrepancy

For clarity, we omit the batch, temporal, and head indices in the following derivation and consider a single time step and attention head. Denote the Query representation of point i, and the Key and Value representations of point j, as

$$\mathbf{q}_i, \mathbf{k}_j, \mathbf{v}_j \in \mathbb{R}^D. \tag{24}$$

Since Query and Key are generated by spiking neurons,

$$q_{id}, k_{jd} \in \{0, 1\}, \tag{25}$$

where d denotes the feature channel. VSSA defines the Query-Key relation using the element-wise absolute difference

$$\mathbf{A}_{ij} = |\mathbf{q}_i - \mathbf{k}_j|. \tag{26}$$

Specifically,

$$|\mathbf{q}_i - \mathbf{k}_j| = [|q_{i1} - k_{j1}|, \ldots, |q_{iD} - k_{jD}|], \tag{27}$$

and therefore $\mathbf{A}_{ij} \in \mathbb{R}^D$. It is important to distinguish this vector discrepancy from the scalar L_1 distance

$$\|\mathbf{q}_i - \mathbf{k}_j\|_1 = \sum_{d=1}^{D} |q_{id} - k_{jd}|. \tag{28}$$

The scalar L_1 distance collapses all feature-channel discrepancies into a single value, whereas VSSA preserves the complete discrepancy vector. Thus, VSSA retains not only whether two point representations differ, but also where these discrepancies occur along the feature channels. For example, consider

$$\mathbf{q}_i = [1, 0, 1, 0], \qquad \mathbf{k}_j = [1, 1, 0, 0]. \tag{29}$$

Their signed difference is

$$\mathbf{q}_i - \mathbf{k}_j = [0, -1, 1, 0], \tag{30}$$

while the element-wise absolute difference is

$$|\mathbf{q}_i - \mathbf{k}_j| = [0, 1, 1, 0]. \tag{31}$$

In contrast, the corresponding scalar L_1 distance is

$$\|\mathbf{q}_i - \mathbf{k}_j\|_1 = 2. \tag{32}$$

VSSA therefore adopts the vector form to preserve channel-wise spike discrepancy information.

A1.2.2 Equivalence Between Absolute Difference and XOR

For binary spike variables $q, k \in \{0, 1\}$, the absolute difference is exactly equivalent to the XOR operation:

$$|q - k| = q \oplus k. \tag{33}$$

This relation can be verified from all four possible binary states:

(q, k)	$\lvert q - k \rvert$	$q \oplus k$
$(0, 0)$	0	0
$(0, 1)$	1	1
$(1, 0)$	1	1
$(1, 1)$	0	0

(34)

Therefore, the absolute difference in VSSA acts as a binary spike-discrepancy detector: identical firing states produce zero response, whereas different firing states produce one. Importantly, this property relies on the absolute difference rather than plain subtraction. For example, when $q = 0$ and $k = 1$, plain subtraction gives $q - k = -1$, whereas the absolute difference gives $|q - k| = 1$, which correctly represents the spike-state discrepancy. Thus, the core interaction in VSSA is

$$|Q - K| \tag{35}$$

rather than simply $Q - K$. For binary variables, XOR can further be expressed as

$$q \oplus k = q + k - 2qk. \tag{36}$$

Combining the two identities gives

$$|q - k| = q \oplus k = q + k - 2qk. \tag{37}$$

Its vector form is

$$|\mathbf{q}_i - \mathbf{k}_j| = \mathbf{q}_i + \mathbf{k}_j - 2\mathbf{q}_i \odot \mathbf{k}_j. \tag{38}$$

This binary identity provides the mathematical basis for the exact factorization of VSSA.

A1.2.3 Discrepancy-Gated Value Aggregation

Based on the vector discrepancy defined above, VSSA computes the output associated with Query point i as

$$\mathbf{o}_i = \frac{1}{N} \sum_{j=1}^{N} \left[|\mathbf{q}_i - \mathbf{k}_j| \odot \mathbf{v}_j \right]. \tag{39}$$

Here, $\odot$ denotes element-wise multiplication, and $1/N$ performs uniform aggregation over all N points. For the d-th feature channel,

$$o_{id} = \frac{1}{N} \sum_{j=1}^{N} |q_{id} - k_{jd}| v_{jd}. \tag{40}$$

Using the XOR equivalence, this becomes

$$o_{id} = \frac{1}{N} \sum_{j=1}^{N} (q_{id} \oplus k_{jd}) v_{jd}. \tag{41}$$

Equivalently,

$$o_{id} = \mathbb{E}_j[(q_{id} \oplus k_{jd}) v_{jd}], \tag{42}$$

where

$$\mathbb{E}_j[f(j)] = \frac{1}{N} \sum_{j=1}^{N} f(j). \tag{43}$$

This formulation shows that each output channel aggregates the average Value response associated with Query-Key spike discrepancies at the same feature channel. When $q_{id} = 0$, we have $0 \oplus k_{jd} = k_{jd}$, and therefore

$$o_{id} = \mathbb{E}_j[k_{jd} v_{jd}]. \tag{44}$$

Thus, when the Query does not fire, Value responses from firing Keys contribute to the output. When $q_{id} = 1$, we have $1 \oplus k_{jd} = 1 - k_{jd}$, yielding

$$o_{id} = \mathbb{E}_j[(1 - k_{jd}) v_{jd}]. \tag{45}$$

In this case, Value responses from non-firing Keys contribute to the output. Hence, VSSA consistently aggregates information associated with spike-state discrepancies rather than reinforcing Query-Key agreement.

A1.2.4 EXACT FACTORIZATION

Substituting equation 38 into the original VSSA formulation gives

$$\mathbf{o}_i = \frac{1}{N}\sum_{j=1}^{N}\left[(\mathbf{q}_i + \mathbf{k}_j - 2\mathbf{q}_i \odot \mathbf{k}_j) \odot \mathbf{v}_j\right]. \tag{46}$$

Expanding the element-wise products yields

$$\mathbf{o}_i = \frac{1}{N}\sum_{j=1}^{N}\mathbf{q}_i \odot \mathbf{v}_j + \frac{1}{N}\sum_{j=1}^{N}\mathbf{k}_j \odot \mathbf{v}_j - \frac{2}{N}\sum_{j=1}^{N}\mathbf{q}_i \odot \mathbf{k}_j \odot \mathbf{v}_j. \tag{47}$$

For a fixed Query point i, $\mathbf{q}_i$ is independent of the summation index j. Therefore,

$$\mathbf{o}_i = \mathbf{q}_i \odot \left(\frac{1}{N}\sum_{j=1}^{N}\mathbf{v}_j\right) + \left(\frac{1}{N}\sum_{j=1}^{N}\mathbf{k}_j \odot \mathbf{v}_j\right) - 2\mathbf{q}_i \odot \left(\frac{1}{N}\sum_{j=1}^{N}\mathbf{k}_j \odot \mathbf{v}_j\right). \tag{48}$$

We define two global statistics:

$$\overline{\mathbf{v}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{v}_j, \qquad \overline{\mathbf{r}} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{k}_j \odot \mathbf{v}_j. \tag{49}$$

The VSSA output can then be written as

$$\mathbf{o}_i = \mathbf{q}_i \odot \overline{\mathbf{v}} + \overline{\mathbf{r}} - 2\mathbf{q}_i \odot \overline{\mathbf{r}}. \tag{50}$$

equation 50 is exactly equivalent to the original pairwise aggregation in equation 39. No approximation, sampling, or information truncation is introduced. For all Query points, the computation can be expressed in vectorized form as

$$\mathbf{O} = \mathbf{Q} \odot \overline{\mathbf{v}} + \overline{\mathbf{r}} - 2\mathbf{Q} \odot \overline{\mathbf{r}}, \tag{51}$$

where $\overline{\mathbf{v}}$ and $\overline{\mathbf{r}}$ are broadcast along the point dimension. The first term $\mathbf{q}_i \odot \overline{\mathbf{v}}$ represents the global Value response conditioned on the current Query state. The second term $\overline{\mathbf{r}}$ represents globally aggregated Key-Value co-activation, while the final term $-2\mathbf{q}_i \odot \overline{\mathbf{r}}$ removes the overlapping contribution when Query and Key are simultaneously active. For example, when $q = k = 1$,

$$q + k - 2qk = 1 + 1 - 2 = 0, \tag{52}$$

which exactly agrees with $|1 - 1| = 0$. Therefore, the factorized formulation strictly preserves the original spike-discrepancy relation.

A1.2.5 COMPLEXITY ANALYSIS

A direct implementation of equation 39 evaluates $|\mathbf{q}_i - \mathbf{k}_j|$ for every Query-Key pair. There are N Query points and N Key points, resulting in N^2 point pairs. Since each pair contains D feature channels, the direct pairwise implementation requires

$$\mathcal{O}(N^2 D) \tag{53}$$

operations. If the complete pairwise discrepancy tensor is explicitly materialized, its size is $N \times N \times D$, resulting in an intermediate memory complexity of

$$\mathcal{O}(N^2 D). \tag{54}$$

In contrast, the factorized implementation computes $\overline{\mathbf{v}}$ and $\overline{\mathbf{r}}$ in $\mathcal{O}(ND)$ operations, respectively. Once these two global statistics are obtained, reconstructing the outputs for all N Query points also requires $\mathcal{O}(ND)$ operations. Therefore,

$$\mathcal{O}(ND) + \mathcal{O}(ND) + \mathcal{O}(ND) = \mathcal{O}(ND). \tag{55}$$

The computational complexity is therefore reduced from

$$\mathcal{O}(N^2 D) \to \mathcal{O}(ND). \tag{56}$$

The factorized formulation also avoids explicitly constructing the $N \times N \times D$ discrepancy tensor. Considering T time steps and H attention heads, the corresponding complexity becomes

$$\mathcal{O}(THN^2D) \rightarrow \mathcal{O}(THND), \tag{57}$$

where D denotes the feature dimension of each attention head and the batch dimension is omitted for clarity. Most importantly, this reduction is not achieved by approximating the original pairwise interaction. Instead, it follows directly from the binary spike identity

$$|Q - K| = Q \oplus K = Q + K - 2QK. \tag{58}$$

Therefore, VSSA preserves the original vector-discrepancy aggregation exactly while eliminating explicit dense point-to-point interactions.

A2 Dataset Introduction

ModelNet40 is a widely used benchmark for 3D object classification, consisting of 12,311 CAD models from 40 object categories, such as chairs, tables, airplanes, and cars. The standard split contains 9,843 models for training and 2,468 models for testing. Since the objects are collected from carefully constructed CAD models, the point clouds generally exhibit complete shapes, clean geometric structures, and limited background interference. Compared with real-world scanned datasets, ModelNet40 provides a relatively controlled setting for evaluating the ability of a model to capture global object shape and structural characteristics.

ScanObjectNN is a challenging real-world point cloud classification benchmark constructed from indoor scenes. It contains approximately 15,000 objects from 15 categories and introduces various realistic factors, including background clutter, occlusion, incomplete geometry, scanning noise, and arbitrary object orientations. Unlike synthetic datasets such as ModelNet40, ScanObjectNN retains many imperfections inherent in real-world 3D acquisition, making it more suitable for evaluating model robustness under complex geometric conditions. We conduct experiments on three commonly used variants, including OBJ_ONLY, which retains only object points, OBJ_BG, which additionally includes background points, and PB_T50_RS, which introduces stronger perturbations.

ShapeNetPart is a large-scale benchmark for fine-grained 3D part segmentation. It contains 16,881 shapes from 16 object categories and provides annotations for 50 semantic part labels. Different object categories contain different part structures; for example, chairs can be decomposed into seats and armrests, while airplanes contain components such as wings and tails. This category-dependent part composition requires a model not only to understand the overall object structure but also to distinguish subtle local geometric differences between neighboring parts. Compared with object classification, ShapeNetPart therefore places greater emphasis on local geometry, point-wise feature discrimination, and the interaction between local structures and global object context, making it an important benchmark for evaluating fine-grained point cloud understanding.

S3DIS is a large-scale indoor point cloud dataset designed for semantic segmentation of real-world 3D scenes. It contains 3D scans collected from six indoor areas, covering 272 rooms with diverse environments such as offices, conference rooms, hallways, storage rooms, and open spaces. Each point is assigned to one of 13 semantic categories. S3DIS contains substantially larger scenes and more complex spatial relationships between different objects. Several semantic categories also exhibit similar local geometric structures, while their spatial distributions and contextual relationships can differ significantly. Following the widely used evaluation protocol, we adopt Area 5 as the held-out test area and use the remaining areas for training, providing a challenging setting for evaluating scene-level geometric representation and generalization across indoor environments.

ScanNetV2 is a large-scale RGB-D indoor scene dataset containing 1,513 reconstructed scans from hundreds of real-world indoor environments. The dataset covers diverse scene types and provides dense 3D geometry together with semantic annotations. The standard semantic segmentation benchmark focuses on 20 semantic categories, including both large structural elements and common indoor objects. Compared with S3DIS, ScanNetV2 contains more diverse scenes and more pronounced real-world acquisition artifacts, such as incomplete surfaces, irregular point densities, occlusions, reconstruction noise, and missing geometry. Its training and validation sets contain 1,201 and 312 scans, respectively, with an additional benchmark test set used for online evaluation. These characteristics make ScanNetV2 a particularly challenging benchmark for assessing point cloud models under large-scale and complex real-world scene conditions.

Table A1: Training configurations for different point cloud understanding tasks.

Parameter	ModelNet40	ScanObjectNN	ShapeNetPart	S3DIS	ScanNetV2
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Learning Rate	5×10^{-4}	5×10^{-4}	2×10^{-3}	1×10^{-3}	1×10^{-3}
Weight Decay	5×10^{-2}	5×10^{-2}	5×10^{-2}	5×10^{-2}	5×10^{-2}
Batch Size	32	32	32	8	2
Training Epochs	300	300	300	100	200
Warm-up Epochs	30	30	30	10	20
Input / Max Points	1024	1024	2048	30000	80000
Spike Threshold	1.0	1.0	1.0	1.0	1.0

A3 Experiment Details

Classification. For point cloud classification, we follow the training configurations of CamPoint. On both ModelNet40 and ScanObjectNN, the models are trained for 300 epochs using AdamW with a cosine learning rate schedule and 30 warm-up epochs. The initial learning rate is set to 5×10^{-4} with a weight decay of 5×10^{-2} and a batch size of 32. We sample 1024 points from each input point cloud. For all spiking neurons, the firing threshold is fixed to 1.0.

Segmentation. For point cloud segmentation, we also follow the corresponding CamPoint settings. On ShapeNetPart, the model is trained for 300 epochs with an initial learning rate of 2×10^{-3}, a batch size of 32, and 30 warm-up epochs, using 2048 input points. For S3DIS and ScanNetV2, the initial learning rate is set to 1×10^{-3}, with batch sizes of 8 and 2 and training schedules of 100 and 200 epochs, respectively. Their maximum input point numbers are set to 30,000 and 80,000. AdamW with cosine learning rate scheduling and a weight decay of 5×10^{-2} is used for all segmentation tasks, and the spike threshold is consistently set to 1.0.

A4 Energy Calculation

One important advantage of spiking neural networks lies in their sparse and event-driven computation. Unlike the multiply-accumulate (MAC) operations widely used in artificial neural networks, binary spikes can convert a large number of multiplications into spike-triggered accumulation (AC) operations, thereby reducing the theoretical energy consumption. Following the energy estimation protocol commonly adopted in existing point cloud SNN methods, we estimate the theoretical inference energy of VGESP based on the actual firing rate of each spiking layer. For the l-th spike-driven layer, the effective number of synaptic operations (SOPs) is defined as

$$\mathrm{SOPs}_l = fr_l \times T \times \mathrm{FLOPs}(l), \qquad (59)$$

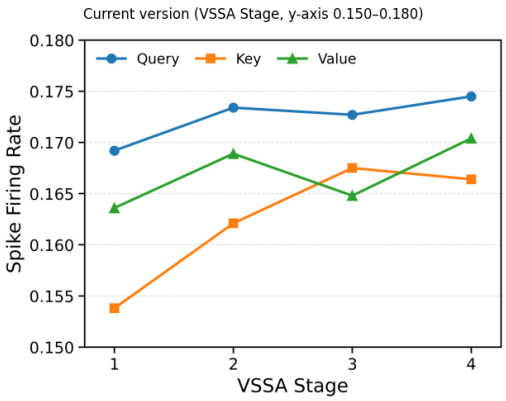

Figure A2: The firing rates of Q, K, V.

where fr_l denotes the average firing rate of the l-th spiking layer, T is the number of SNN timesteps, and $\mathrm{FLOPs}(l)$ denotes the number of operations of the corresponding layer. The firing rate is measured from binary spike activations over the test samples, such that AC operations are triggered only at activated spike positions. As shown in Figure A2, the firing rates of Query, Key, and Value across different VSSA blocks remain at relatively low levels, indicating highly sparse spike activity in VGESP. This sparsity reduces the number of effective AC operations and consequently lowers the theoretical inference energy. Since the main body of VGESP is based on spike-driven computation and only the task-specific prediction heads contain a small number of floating-point MAC

operations, the theoretical inference energy is estimated as

$$E_{\mathrm{VGESP}} = E_{\mathrm{MAC}} \times \mathrm{FLOPs}_{\mathrm{head}} + E_{\mathrm{AC}} \sum_{l=1}^{L} \mathrm{SOPs}_l, \tag{60}$$

where $\mathrm{FLOPs}_{\mathrm{head}}$ denotes the floating-point operations in the task-specific prediction head, and L is the number of spike-driven layers. For computation layers receiving binary spikes, their operations are converted into AC operations according to the corresponding firing rates. Following the theoretical energy setting widely adopted in SNN studies under 45 nm technology with 32-bit floating-point operations (Kundu et al., 2021; Horowitz, 2014), we use

$$E_{\mathrm{MAC}} = 4.6\ \mathrm{pJ}, \qquad E_{\mathrm{AC}} = 0.9\ \mathrm{pJ}. \tag{61}$$

Combined with the sparse binary geometric events generated by VGE and the sparse spike activations within the network, VGESP performs feature propagation primarily through low-cost AC operations, thereby maintaining low theoretical inference energy consumption.

A5 More Experiment

A5.1 Effect of Spiking Neuron Models.

We first compare three spiking neuron models, including EIF, LIF, and IF, to evaluate their influence on VGESP. As shown in Table A2, EIF and LIF achieve 70.2% and 70.8% mIoU, respectively, while IF obtains the best performance with 71.2% mIoU and 91.0% OA. Compared with LIF and EIF, IF improves the mIoU by 0.4% and 1.0%, respectively. These results indicate that simple integrate-and-fire dynamics are already effective for processing the sparse geometric events generated by VGE and the spike features within the network, while more complex leakage or exponential membrane dynamics do not provide additional benefits.

A5.2 Effect of Rotation axis in VGE Module.

The rotation direction of the virtual camera directly affects the geometric variations induced by consecutive viewpoints. We therefore compare camera rotations around the x-, y-, and z-axes. As shown in Table A3, rotations around the x- and y-axes achieve 69.9% and 70.6% mIoU, respectively, whereas rotation around the z-axis obtains the best result of 71.2% mIoU with 91.0% OA. Indoor scenes typically exhibit clear vertical structures and horizontal spatial layouts. Rotating around the z-axis preserves the vertical geometric relationships while introducing richer horizontal viewpoint variations, which facilitates the generation of more discriminative visibility and depth events.

A5.3 Effect of Block Depth.

We further vary the block depth M to investigate its influence on global feature interaction. Here, M denotes the number of stacked blocks in each stage. The whole network contains four stages, and the same M is used for all stages. As shown in Table A4, When $M = 1$, the model achieves 91.0% OA and 71.2% mIoU. Increasing M to 2 maintains the mIoU at 71.2%, while the OA slightly decreases to 90.8%. As M further increases to 3 and 4, the mIoU decreases to 70.8% and 70.1%, respectively. This suggests that a shallow block configuration is sufficient for effective feature interaction, while further increasing the depth may introduce redundant global information propagation and additional model complexity.

A5.4 Effect of Camera rotation angle.

The magnitude of viewpoint variation is another important factor affecting geometric event generation in VGE. As shown in Table A5, increasing the rotation angle from 1 to 2 improves the mIoU from 69.6% to 71.2% and the OA from 89.7% to 91.0%. Further increasing the angle to 3 and 4 reduces the mIoU to 70.8% and 69.9%, respectively. A small rotation angle induces limited viewpoint changes and may not sufficiently activate geometric events, whereas an excessively large angle can cause more drastic visibility and depth variations, introducing redundant or unstable event responses. Therefore, a moderate viewpoint variation provides a better balance between event diversity and stability.

Table A2: Effect of different spiking neuron models on S3DIS.

Neuron	OA(%)	mIoU(%)
EIF	90.1	70.2
LIF	90.5	70.8
IF	**91.0**	**71.2**

Table A3: Effect of different rotation axes in VGE on S3DIS.

Axis	OA(%)	mIoU(%)
x	89.8	69.9
y	90.5	70.6
z	**91.0**	**71.2**

Table A4: Effect of the number of stacked blocks M on S3DIS.

M	OA(%)	mIoU(%)
1	**91.0**	**71.2**
2	90.8	71.2
3	90.5	70.8
4	89.9	70.1

Table A5: Effect of the camera rotation angle in VGE on S3DIS.

Angle	OA(%)	mIoU(%)
1	89.7	69.6
2	**91.0**	**71.2**
3	90.6	70.8
4	90.0	69.9

A5.5 Effect of Geometric Event Components.

To investigate the contribution of different geometric event components in VGE, we evaluate the performance of visibility events and depth events separately. As shown in Table A6, removing both event components reduces the performance to 89.6% OA and 69.1% mIoU, demonstrating the effectiveness of introducing viewpoint-induced geometric events. Using visibility events alone improves the mIoU to 70.8%, while depth events alone achieve 70.4% mIoU, indicating that both types of geometric changes provide useful temporal cues. Combining visibility and depth events achieves the best performance with 91.0% OA and 71.2% mIoU, validating their complementary roles in capturing discrete visibility transitions and continuous depth variations.

A5.6 Effect of Camera rotation angle.

We further analyze the sensitivity of the depth-change threshold τ in VGE. As shown in Table A7, VGE maintains stable performance under different threshold settings, indicating the robustness of the proposed depth event generation. A smaller threshold may introduce insignificant depth fluctuations, while a larger threshold can suppress informative geometric variations. The default setting of $\tau = 0.02$ achieves the best performance by providing an appropriate balance between event sensitivity and noise suppression.

A6 LLM Usage Statement

In accordance with the relevant ICLR policy regarding the use of Large Language Models (LLMs), the authors hereby declare that the core academic content of this research, including the formulation of research ideas, methodology design, experimental implementation and data analysis, derivation of results, and formulation of conclusions, was independently completed by the authors.

During the paper writing process, AI tools were used only for language polishing and expression refinement of selected parts of the manuscript to improve grammatical accuracy, readability, and fluency. AI tools were not involved in the conception of the research, methodology design, experimental execution, result analysis, or formulation of scientific conclusions. All AI-assisted revisions were further reviewed and verified by the authors.

The authors take full responsibility for the authenticity, accuracy, originality, and academic integrity of all content in this paper. The use of AI tools was strictly limited to language-level assistance and did not affect the independence or scientific integrity of the research.

Table A6: Effect of different geometric event components in VGE on S3DIS.

Visibility	Depth	OA(%)	mIoU(%)
		89.6	69.1
✓		90.7	70.8
	✓	90.3	70.4
✓	✓	**91.0**	**71.2**

Table A7: Effect of depth-change threshold τ in VGE on S3DIS.

τ	OA(%)	mIoU(%)
0.01	90.4	70.8
0.02	**91.0**	**71.2**
0.03	90.6	-70.9
0.04	90.3	-70.5

A7 Ethics Statement

We confirm that all authors have read and comply with the ICLR Code of Ethics. Our work does not involve human subjects, sensitive data collection or release, potentially harmful methodologies or applications, conflicts of interest, discrimination/bias/fairness concerns, privacy/security issues, legal compliance challenges, or research integrity matters that require explicit addressing.

A8 Reproducibility Statement

To ensure the reproducibility of our results, we will make our implementation code available as supplementary material. The datasets used (e.g., ModelNet40, ScanObjectNN, ShapeNetPart for point cloud tasks) are publicly accessible, with relevant details provided in the main text. Key experimental settings (including hyperparameters like SNN threshold and timestep optimized in our analysis) and procedures are thoroughly documented in the manuscript. Any theoretical derivations supporting our work are included in the appendix.