MAG-SSM: MOTION-ADAPTIVE GEOMETRIC STATE-SPACE MODELING FOR 4D POINT CLOUD VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

4D point cloud videos naturally represent 3D geometry and temporal evolution, but their irregular and evolving structures challenge adaptive local modeling and coherent long-range propagation. Existing methods often rely on fixed local aggregation and predefined sequence organization, limiting their ability to jointly adapt receptive fields, token organization, and local geometric modeling to evolving 4D structures. Our core insight is that effective state-space modeling of dynamic point clouds requires geometric adaptation to be coordinated throughout the propagation pipeline, rather than confined to local feature extraction. To this end, we propose MAG-SSM, a Motion-Adaptive Geometric State-Space Model for 4D point cloud video understanding that coordinates geometric adaptation across local representation, sequence organization, and geometric recovery. Specifically, Motion-Adaptive Spatio-Temporal Sampling (MSTS) adapts cross-frame local representations through bounded anchor relocation. Motion-Adaptive Semantic Routing Scanning (MSR-Scanning) forms priority-aware sequences from motion-adaptive representations, while Offset-Guided Local Geometric Fusion (OGLF) restores local geometric relations with aligned offset priors. Experiments on MSR-Action3D, Synthia 4D, and HOI4D demonstrate consistent improvements in action recognition, 4D semantic segmentation, and action segmentation with favorable computational efficiency.

1 INTRODUCTION

4D point cloud videos encode 3D geometry and temporal evolution, providing a natural representation for dynamic-scene understanding, human motion analysis, robotic perception, and autonomous driving (Huang et al., 2026; Jing et al., 2024; Liu et al., 2025; Lv et al., 2025; Gong et al., 2026; Ben-Shabat et al., 2024; Xu et al., 2024). Unlike grid-structured RGB videos, point cloud videos are spatially irregular and unordered yet temporally continuous, requiring models to capture evolving geometric structures while maintaining long-range temporal interactions.

Existing 4D point cloud video understanding methods mainly follow convolutional or Transformer-based paradigms. Convolutional methods such as MeteorNet and PSTNet construct local spatio-temporal neighborhoods but have limited receptive fields (Liu et al., 2019; Fan et al., 2021b), while Transformer-based methods such as P4Transformer, PST-Transformer, PPTr, and X4D-SceneFormer capture interactions through self-attention at increasing computational and memory cost (Fan et al., 2021a; 2023; Wen et al., 2022; Jing et al., 2024). Recently, Selective State Space Models (SSMs) provide a more efficient alternative for long-sequence modeling (Gu & Dao, 2024; Zhang et al., 2025; Li et al., 2025b). For 4D point cloud videos, Mamba4D performs disentangled spatial and temporal state modeling (Liu et al., 2025), while UST-SSM constructs semantic token sequences and restores serialization-weakened local structures (Li et al., 2025a). Despite their gratifying progress, existing SSM-based 4D methods still exhibit a fundamental mismatch between evolving 4D geometry and largely static geometric organization during state propagation.

This mismatch manifests throughout the state-propagation pipeline of 4D point cloud video understanding. Our motivations lie in three aspects. *Firstly*, local features are typically aggregated around fixed or stride-based anchors, making the receptive fields less responsive to motion-induced geomet-

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.

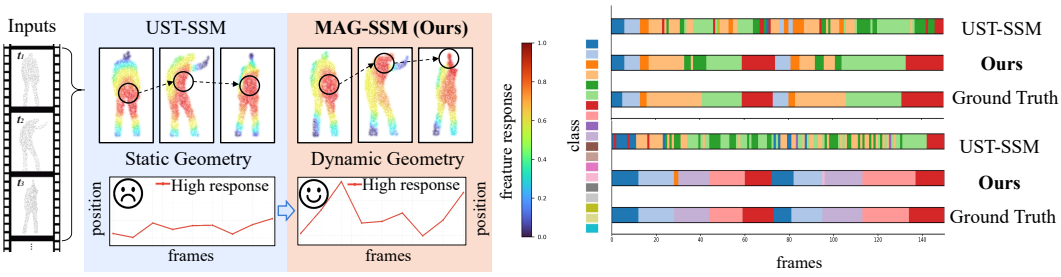


Figure 1: Motivation of MAG-SSM. (a) Static vs. Dynamic geometric modeling: UST-SSM (Li et al., 2025a), as a representative baseline, largely relies on static geometric organization, which may limit adaptation to evolving structures. In contrast, MAG-SSM adapts geometric modeling to temporal evolution, producing responses that better follow changing structures across frames. Heatmaps visualize model responses, while trajectories are schematic. (b) Action-segmentation comparison: MAG-SSM produces fewer fragmented segments and spurious label transitions.

ric variations; as illustrated in Figure 1(a), high-response regions may remain relatively static across frames rather than following the evolving geometry. *Secondly*, predefined scan paths or motion-agnostic semantic sequences organize tokens without explicitly considering task-relevant saliency in motion-adaptive representations, which limits the propagation of dynamically informative context during state-space modeling. *Thirdly*, although local aggregation can partially alleviate the topology degradation caused by 1D serialization (Yang et al., 2025; Li et al., 2026; 2025a), existing approaches do not explicitly account for receptive-field adaptation when reconstructing local geometric relations. These limitations may make it harder to maintain coherent representations across dynamic observations, potentially resulting in fragmented temporal predictions and spurious label transitions, as illustrated in Figure 1(b).

Motivated by these observations, we revisit state-space modeling of 4D point cloud videos from the perspective of dynamic geometric adaptation. Our core insight is that evolving geometry should not only determine how local observations are constructed, but should also be consistently reflected in how these representations are organized and propagated through the state-space model. In other words, geometric adaptation should extend throughout the state-propagation pipeline, from local representation to sequence organization and subsequent geometric recovery.

Guided by this insight, we propose MAG-SSM, a Motion-Adaptive Geometric State-Space Model for 4D point cloud video understanding. MAG-SSM adopts motion-adaptive geometric modeling to adjust local representations to evolving 4D structures and further extends this adaptation to sequence organization and local geometric modeling. MAG-SSM comprises three core components. *Firstly*, Motion-Adaptive Spatio-Temporal Sampling (MSTS) predicts bounded 3D anchor offsets from positional cues and encoded spatio-temporal context, and retrieves cross-frame neighborhoods around relocated anchors to adapt local receptive fields to evolving geometry. *Secondly*, Motion-Adaptive Semantic Routing Scanning (MSR-Scanning) organizes MSTS-enhanced tokens into learnable semantic groups and prioritizes their serialization according to task-relevant saliency, promoting coherent propagation of important representations. *Thirdly*, Offset-Guided Local Geometric Fusion (OGLF) reconstructs local geometric relations over reordered anchors and incorporates aligned offsets as geometric priors before state-space updates, allowing local modeling to remain consistent with receptive-field variations. Our main contributions are summarized as follows:

- We revisit 4D point cloud state-space modeling from the perspective of dynamic geometric adaptation. We identify a fundamental mismatch between evolving 4D geometry and the static geometric organization of existing SSMs.
- We propose a novel MAG-SSM framework to coordinate geometric adaptation throughout state propagation. MSTS adapts local receptive fields, MSR-Scanning constructs priority-aware sequences, and OGLF restores local geometry with adaptive offset priors.
- Extensive experiments across action recognition, 4D semantic segmentation, and action segmentation demonstrate consistent performance improvements while maintaining a favorable balance between effectiveness and efficiency across multiple benchmarks.

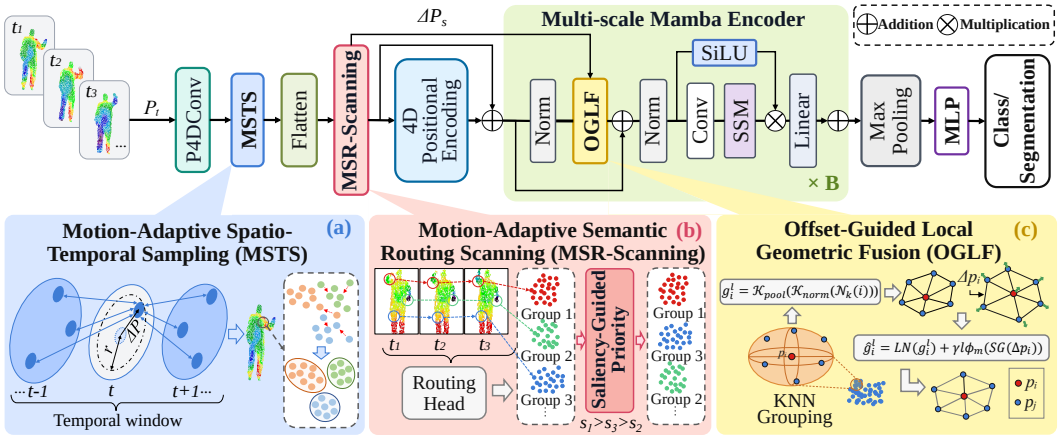


Figure 2: Overview of MAG-SSM. (a) MSTS: spatio-temporal anchor relocation and cross-frame sampling. (b) MSR-Scanning: saliency-guided enhancement and serialization of motion-adaptive tokens. (c) OGLF: local geometric reconstruction with aligned offset priors.

2 Related Work

State Space Models. State Space Models provide an alternative to attention-based long-sequence modeling. Building on S4 (Gu et al., 2022), Mamba (Gu & Dao, 2024) introduces input-dependent state selection and efficient scanning, extended to visual modeling through VMamba (Liu et al., 2024) and VideoMamba (Li et al., 2024). For point clouds, PointMamba (Liang et al., 2024), Point Cloud Mamba (Zhang et al., 2025), and Mamba3D (Han et al., 2024) explore serialization, representation, and geometric modeling. FlowMamba (Lin et al., 2025) designs sequence organization for scene flow estimation, whereas MSR-Scanning forms priority-aware semantic sequences from motion-adaptive representations for 4D understanding. For point cloud videos, Mamba4D (Liu et al., 2025) disentangles spatiotemporal state modeling, UST-SSM (Li et al., 2025a) combines semantic sequences with recovery, and PST-Mamba (Song et al., 2025) introduces spatio-temporal selective state fusion. In contrast, MAG-SSM coordinates geometric adaptation across local representation, priority-aware sequence organization, and local geometric recovery during state propagation.

4D Point Cloud Video Understanding. Early 4D point cloud methods such as MeteorNet (Liu et al., 2019) and PSTNet (Fan et al., 2021b) model local spatio-temporal relations through cross-frame aggregation and convolution. Transformer-based approaches, including P4Transformer (Fan et al., 2021a), PST-Transformer (Fan et al., 2023), PPTr (Wen et al., 2022), and X4D-SceneFormer (Jing et al., 2024), further capture long-range interactions, while PL2-Transformer (Zuo et al., 2025) emphasizes long-term locality-aware modeling. Recent studies explore improved network design, human-centric representation, cross-frame adaptation, self-supervised learning, and graph or spectral modeling (Wang et al., 2025; Ben-Shabat et al., 2024; Xu et al., 2024; Lv et al., 2025; Liu et al., 2023; Shen et al., 2023a;b; Zhang et al., 2023). Recent work also studies egocentric dynamic-scene reasoning and adaptive segmentation (Huang et al., 2026; Gong et al., 2026). MAG-SSM differs by coordinating geometric adaptation across local representation, sequence organization, and geometric recovery during state propagation rather than solely extending interaction range or feature mixing.

3 Methodology

This section presents MAG-SSM, illustrated in Figure 2. Given a T-frame 4D point cloud video, Point 4D Convolution extracts initial spatio-temporal features. As shown in Figure 2(a), Motion-Adaptive Spatio-Temporal Sampling (MSTS) predicts bounded anchor offsets and retrieves cross-frame neighborhoods around relocated anchors to produce motion-adaptive representations F_t^d. After flattening F_t^d across frames, Motion-Adaptive Semantic Routing Scanning (MSR-Scanning) organizes motion-adaptive representations into priority-aware semantic sequences. The permutation reorders token features, original and relocated anchors, and temporal indices, yielding E_s, P_s, $\widetilde{P}_s$, and ν_s, with $\Delta P_s = \widetilde{P}_s - P_s$. The reordered sequence is processed by a Mamba encoder. As shown

in Figure 2(c), Offset-Guided Local Geometric Fusion (OGLF) reconstructs geometric relations over reordered anchors and incorporates aligned offsets as priors before each state-space update.

3.1 Motion-Adaptive Spatio-Temporal Sampling

Point-based 4D operators commonly construct local spatio-temporal neighborhoods around fixed anchors, limiting their adaptation to evolving geometric variations. MSTS instead conditions anchor relocation on encoded spatio-temporal context, allowing the spatial support of each receptive field to adapt before cross-frame aggregation.

Adaptive Anchor Offset Prediction. For frame t, farthest point sampling selects M anchor indices $I_t = \mathrm{FPS}(P_t^0, M)$, where $I_t \in \{1, \dots, N_0\}^M$. The corresponding coordinates and features are $A_t = P_t^0[I_t] \in \mathbb{R}^{M\times 3}$ and $H_t = F_t^0[I_t] \in \mathbb{R}^{M\times C_0}$. We construct the position-aware center context as $\bar{H}_t = H_t + \phi_{\mathrm{pos}}(A_t)$, where ϕ_{pos} projects 3D coordinates into the feature space. Since F_t^0 is produced by P4DConv, $\bar{H}_t$ combines encoded spatio-temporal context with explicit positional cues for context-conditioned anchor relocation.

Let $\mathcal{N}_i$ denote the K_{A} nearest sampled anchors to $a_{t,i}$ in frame t. For each neighbor, we normalize the center-relative feature residual and concatenate the position-aware center context:

$$v_{i,j} = \left[\frac{\bar{h}_{t,j} - \bar{h}_{t,i}}{\max(\sigma_t, \epsilon)} \,\middle\|\, \bar{h}_{t,i} \right], \qquad j \in \mathcal{N}_i \tag{1}$$

where $\bar{h}_{t,i}$ and $\bar{h}_{t,j}$ are rows of $\bar{H}_t$, σ_t is a shared scalar standard deviation computed from the local anchor-feature residuals at frame t, and $\epsilon = 10^{-5}$. The descriptor combines normalized position-aware feature contrast with center context, while explicit geometric distances are incorporated during the subsequent aggregation.

A shared affine transformation is applied to each neighbor, followed by permutation-invariant geometry-aware pooling:

$$o_{t,i} = \phi_{\mathrm{off}} \left(\mathcal{A}_{\mathrm{geo}} \left(\{ w_{i,j} \phi_{\mathrm{aff}}(v_{i,j}) \}_{j\in\mathcal{N}_i} \right) \right) \in \mathbb{R}^3 \tag{2}$$

where $w_{i,j} = \exp(-\|a_{t,j} - a_{t,i}\|_2)$ weights neighboring anchors according to their geometric distance, and $\mathcal{A}_{\mathrm{geo}}$ denotes channel-wise exponential soft pooling over the weighted responses. The distances are computed directly from the anchor coordinates used for neighborhood construction, without an additional bandwidth parameter. This operation is independent of neighbor ordering while assigning larger weights to spatially closer anchors.

To restrict excessive anchor displacement, the predicted offset is constrained componentwise according to the neighborhood radius:

$$\Delta a_{t,i} = \mathrm{clip}\left(o_{t,i}, -\eta r, \eta r\right), \qquad \tilde{a}_{t,i} = a_{t,i} + \Delta a_{t,i} \tag{3}$$

where r and η denote the neighborhood radius and offset constraint factor. The componentwise constraint guarantees $|\Delta a_{t,i}^{(d)}| \le \eta r$ along each spatial dimension d. The resulting offset represents the learned spatial adjustment of the receptive field under the current spatio-temporal context.

4D Spatio-temporal Aggregation. We perform cross-frame neighbor retrieval within the temporal window $\Omega_t = \{t-m, \dots, t+m\}$, where m denotes the temporal radius. For anchor i in frame t, we retrieve K_{M} neighboring points within radius r from P_τ^0 around the relocated anchor $\tilde{a}_{t,i}$ for each frame $\tau \in \Omega_t$. Let $q_{\tau,j}$ and $f_{\tau,j}^0$ denote the coordinate and feature of the j-th retrieved point in frame τ. Its relative 4D displacement to $\tilde{a}_{t,i}$ is:

$$d_{\tau,j} = \left[q_{\tau,j} - \tilde{a}_{t,i} \,\middle\|\, \tau - t \right] \in \mathbb{R}^4 \tag{4}$$

The spatial term describes each neighbor relative to the relocated anchor, while the signed temporal term indicates its temporal position relative to the center frame. The descriptor therefore jointly encodes spatial relations and temporal direction within the local window.

The neighbor feature and its 4D displacement are projected into the same C-dimensional space and fused by a shared pointwise mapping:

$$g_{\tau,j} = \phi_g \left(\phi_f(f_{\tau,j}^0) + \phi_d(d_{\tau,j}) \right) \tag{5}$$

where ϕ_f and ϕ_d denote feature and displacement projections, respectively, and ϕ_g denotes the shared pointwise fusion mapping. MSTS then performs spatial aggregation within each frame and temporal aggregation across the window:

$$f_{t,i}^{d} = \mathcal{A}_{\text{temp}}\left(\left\{\mathcal{A}_{\text{sp}}\left(\{g_{\tau,j}\}_{j=1}^{K_{\text{M}}}\right)\right\}_{\tau\in\Omega_t}\right) \tag{6}$$

where $\mathcal{A}_{\text{sp}}$ and $\mathcal{A}_{\text{temp}}$ denote spatial and temporal aggregation, both implemented by max pooling. Spatial aggregation is permutation invariant, while the relocated query anchor and signed temporal coordinates enable $f_{t,i}^{d}$ to capture local cross-frame evolution.

3.2 Motion-Adaptive Semantic Routing Scanning

MSR-Scanning organizes MSTS-enhanced motion-adaptive representations into priority-aware semantic sequences according to learned group saliency. The learned saliency reflects group importance within the adapted representations rather than physical motion magnitude.

Semantic Group Routing. Let T_d denote the temporal length of the MSTS output. Flattening MSTS outputs yields $L = T_d M$ tokens $Z = [z_i]_{i=1}^{L} \in \mathbb{R}^{L\times C}$, where $z_i = f_{t,a}^{d}$ for $i = (t-1)M + a$. Each token is associated with its original MSTS output-frame index ν_i, which is retained throughout subsequent routing and serialization. A routing head maps each token to logits $B_i = \phi_r(z_i) \in \mathbb{R}^{K_s}$ over K_s semantic groups. Gumbel-Softmax first estimates a soft assignment $\widetilde{\Pi}_i$, which is converted into a discrete routing decision Π_i through a straight-through estimator:

$$\begin{aligned}\widetilde{\Pi}_i &= \text{GumbelSoftmax}(B_i;\tau_g)\\ \Pi_i &= \text{onehot}\left(\arg\max_k \widetilde{\Pi}_{i,k}\right) + \widetilde{\Pi}_i - \text{SG}(\widetilde{\Pi}_i)\end{aligned} \tag{7}$$

where τ_g is the temperature and $\text{SG}(\cdot)$ denotes the stop-gradient operator. This design maintains discrete assignments in the forward pass while preserving gradients through the soft assignments $\widetilde{\Pi}_i$. During inference, each token is assigned to the group with the maximum routing logit.

Group Saliency Enhancement. For each semantic group k, we compute a stop-gradient prototype $c_k = \sum_i \Pi_{i,k}\,\text{SG}(z_i)/(\sum_i \Pi_{i,k} + \epsilon)$ and obtain a task-relevant group saliency score $s_k = \phi_s(c_k) \in (0,1)$. Because c_k is computed from MSTS-enhanced representations, s_k measures task-relevant group importance after receptive-field adaptation and is not interpreted as physical motion magnitude. Each token inherits the score of its assigned group, $u_i = s_{\ell_i}$, yielding:

$$x_i = \left[z_i \odot (1+\lambda u_i) \,\middle\|\, \Pi_i E_p\right] \in \mathbb{R}^{D} \tag{8}$$

where ℓ_i is the group of token i, $E_p = E_b E_h \in \mathbb{R}^{K_s\times C_p}$ is a factorized semantic prompt table, λ is the residual factor, and $D = C + C_p$. The residual modulation preserves the original token representation while emphasizing groups with stronger task-relevant responses. Stacking all enhanced tokens gives the sequence $E = [x_i]_{i=1}^{L}$.

Saliency-Guided Priority Locking. Mini-batch group scores may fluctuate during training and produce unstable estimates of semantic importance. We therefore maintain an exponential moving average (EMA) of the group-level saliency scores:

$$\bar{s}_k \leftarrow \beta \bar{s}_k + (1-\beta)\frac{1}{B}\sum_{b=1}^{B} s_{b,k} \tag{9}$$

where β is EMA momentum. The accumulated scores rank semantic groups in descending saliency, yielding priority mapping r_k. Tokens are reordered by r_{ℓ_i} while preserving intra-group order, making high-saliency representations more contiguous. EMA statistics and discrete ranking are updated without gradients, while ϕ_s is optimized by $\mathcal{L}_{\text{task}}$ through Eq. 8, requiring no auxiliary ranking loss.

The original anchors, relocated anchors, and temporal indices are flattened using the same token order as E. The resulting permutation ρ is consistently applied to all aligned quantities: $E_s = E[\rho]$, $P_s = P[\rho]$, $\widetilde{P}_s = \widetilde{P}[\rho]$, and $\nu_s = \boldsymbol{\nu}[\rho]$, with $\Delta P_s = \widetilde{P}_s - P_s$. We construct $Q_s = [P_s \| \nu_s/T_d]$ and, for each sequence subsampling stride r_s, consistently subsample the aligned quantities into their branch-specific sequences. The corresponding encoder input is $U^{0,(r_s)} = E^{(r_s)} + \phi_{\text{4D}}(Q^{(r_s)})$.

3.3 OFFSET-GUIDED LOCAL GEOMETRIC FUSION

Although MSR-Scanning reorganizes motion-adaptive representations for state propagation, 1D serialization may weaken geometric adjacency among reordered anchors. OGLF therefore reconstructs geometry-based local neighborhoods within each branch and incorporates aligned MSTS offsets ΔP_s as geometric priors before each Mamba update. For clarity, we omit the multi-scale branch superscript below.

Geometry-Based Neighborhood Reconstruction. In the l-th encoder block, let $u_i^{l-1} = U^{l-1}[i]$ denote the feature of token i, and let $p_i = P_s[i]$ and $\Delta p_i = \Delta P_s[i]$ denote its original anchor coordinate and aligned offset, respectively. The local neighborhood is defined as

$$\mathcal{N}_K(i) = \mathrm{KNN}\left(p_i, P_s, K\right) \tag{10}$$

where P_s denotes the reordered anchors in the current branch and K is the corresponding neighborhood size. The neighborhood is determined by 3D geometric proximity rather than 1D sequence adjacency, reconnecting spatially adjacent anchors that may become separated after serialization.

Normalized Local Aggregation. The input features are first normalized as $\tilde{u}_i^{l-1} = \mathrm{LN}(u_i^{l-1})$. Given $\mathcal{N}_K(i)$, OGLF applies K-Norm to encode normalized local feature relations:

$$h_i^l = \mathcal{K}_{\mathrm{norm}}\left(p_i, \tilde{u}_i^{l-1}, \left\{(p_j, \tilde{u}_j^{l-1})\right\}_{j\in\mathcal{N}_K(i)}\right) \tag{11}$$

Specifically, K-Norm normalizes neighbor-center feature residuals using a shared local standard deviation, concatenates them with the corresponding center feature, and applies learnable affine scaling. The resulting responses are aggregated by K-Pool:

$$g_i^l = \mathcal{K}_{\mathrm{pool}}\left(h_i^l\right) \tag{12}$$

K-Pool performs channel-wise exponential weighted aggregation over K responses. Together, K-Norm and K-Pool recover geometry-based interactions independently of 1D sequence adjacency.

Gated Offset Injection. The reconstructed neighborhood captures geometry defined by the original anchors, while the aligned offset encodes the corresponding receptive-field adjustment. OGLF incorporates this offset through a gated residual branch:

$$\widehat{g}_i^l = \mathrm{LN}(g_i^l) + \gamma_l \phi_m\left(\mathrm{SG}(\Delta p_i)\right) \tag{13}$$

where ϕ_m projects the 3D offset into the feature space and γ_l is a learnable scalar initialized to zero. The offset serves as a geometric prior describing the direction and magnitude of the receptive-field adjustment. The stop-gradient operation prevents gradients from the OGLF offset branch from propagating back to the MSTS offset predictor.

The enhanced token is defined as $y_i^l = \mathrm{SiLU}(\phi_o(\widehat{g}_i^l))$, and all tokens are stacked into Y^l. OGLF is followed by the Mamba update through residual connections:

$$\begin{aligned} \bar{U}^l &= U^{l-1} + \mathrm{DropPath}(Y^l), \\ U^l &= \bar{U}^l + \mathrm{DropPath}\left(\mathrm{Mamba}_l\left(\mathrm{LN}(\bar{U}^l)\right)\right) \end{aligned} \tag{14}$$

4 EXPERIMENTS

4.1 DATASETS AND EXPERIMENTAL SETTINGS

Datasets. We evaluate MAG-SSM on three 4D point cloud benchmarks covering different prediction granularities. MSR-Action3D (Li et al., 2010) consists of 567 depth videos captured by Kinect v1, spanning 20 action categories. Following P4Transformer (Fan et al., 2021a), we use 270 videos for training and 297 for testing. Synthia 4D (Choy et al., 2019) contains six urban driving sequences for 4D semantic segmentation, and we adopt the data split used by MeteorNet (Liu et al., 2019). HOI4D (Liu et al., 2022) is an egocentric human-object interaction benchmark for action segmentation, containing 2,971 training scenes and 892 testing scenes under the official protocol.

Table 1: Action recognition accuracy (%) on the MSR-Action3D dataset.

Backbone	Method	Venue	Acc		
			16	24	36
CNN	Kinet (Zhong et al., 2022)	CVPR'22	91.92	93.27	–
	PointCMP (Shen et al., 2023b)	CVPR'23	92.26	93.27	–
	3DInAction (Ben-Shabat et al., 2024)	CVPR'24	90.57	92.23	–
	PvNeXt (Wang et al., 2025)	ICLR'25	93.93	94.77	–
Transformer	P4Transformer (Fan et al., 2021a)	CVPR'21	89.56	90.94	82.81
	PPTr (Wen et al., 2022)	ECCV'22	90.31	92.33	–
	PST-Transformer (Fan et al., 2023)	TPAMI'23	91.98	93.73	91.15
	PointCPSC (Sheng et al., 2023)	ICCV'23	92.26	92.68	–
SSM	Mamba4D (Liu et al., 2025)	CVPR'25	–	92.68	93.23
	UST-SSM (Li et al., 2025a)	ICCV'25	93.27	94.77	95.12
	MAG-SSM (Ours)	–	**94.28**	**97.91**	**96.35**

Table 2: Action segmentation results (%) on the HOI4D dataset.

Method	Acc	Edit	F1@10	F1@25	F1@50
P4Transformer+C2P	73.5	76.8	77.2	72.9	62.4
PPTr	77.4	80.1	81.7	78.5	69.5
PPTr+C2P	81.1	**84.0**	85.4	82.5	74.1
PST-Transformer	79.2	59.9	65.4	62.7	54.6
PointNet4D	73.9	73.8	75.3	71.3	61.3
UST-SSM	77.2	71.9	75.8	72.7	63.2
Mamba4D	73.9	69.6	73.7	70.1	59.7
MAG-SSM	**81.8**	82.7	**85.6**	**83.2**	**75.9**

Table 3: Semantic segmentation results on Synthia 4D, measured by mIoU (%).

Method	Venue	Frames	mIoU
MeteorNet-m	ICCV'19	2	81.47
MeteorNet-l	ICCV'19	3	81.80
PSTNet	ICLR'21	3	82.24
P4Transformer	CVPR'21	3	83.16
PST-Transformer	TPAMI'23	3	83.95
Mamba4D	CVPR'25	3	83.35
UST-SSM	ICCV'25	3	83.93
MAG-SSM	–	3	**84.42**

Experimental Settings. All experiments are implemented in PyTorch 2.0.1 with CUDA 11.8 and trained on a single NVIDIA RTX 4090 GPU with 24 GB memory. We use SGD with momentum 0.9 and weight decay 10^{-4} for 50 epochs with a batch size of 16. For MSR-Action3D and Synthia 4D, the initial learning rate is 0.01 and decays by a factor of 0.1 at epochs 20 and 30; for HOI4D, it is initialized to 0.02 and decayed by a factor of 0.5 at epochs 35 and 45. A 10-epoch learning-rate warm-up is adopted, and the MSR-Scanning group priority is fixed at epoch $E_w = 20$. For MSR-Action3D, we use $T \in \{16, 24, 36\}$ with $N = 2{,}048$ points per frame; Synthia 4D uses $T = 3$ with $N = 16{,}384$; HOI4D sequences contain $T = 150$ frames with $N = 2{,}048$ points per frame and are processed using overlapping 24-frame windows with stride 12. Detailed hyperparameters, branch configurations, losses, and exact operator definitions are provided in Appendix B.

4.2 Main Results

3D Action Recognition. As shown in Table 1, MAG-SSM achieves the best performance across all evaluated sequence lengths, peaking at 97.91% with 24 frames. The improvement from 16 to 24 frames suggests that richer temporal context benefits motion-adaptive representation and sequence organization, whereas the slight drop to 96.35% at 36 frames may arise because fixed K_s increases group occupancy and temporal span, potentially separating temporally adjacent tokens after priority-based serialization. Nevertheless, MAG-SSM still exceeds UST-SSM by 1.23 percentage points at 36 frames, demonstrating robust modeling of longer temporal contexts.

4D Action Segmentation We further evaluate MAG-SSM on HOI4D and compare it with P4Transformer (Fan et al., 2021a), PPTr (Wen et al., 2022), their C2P-enhanced variants (Zhang et al., 2023), and the adapted UST-SSM and Mamba4D backbones (Li et al., 2025a; Liu et al.,

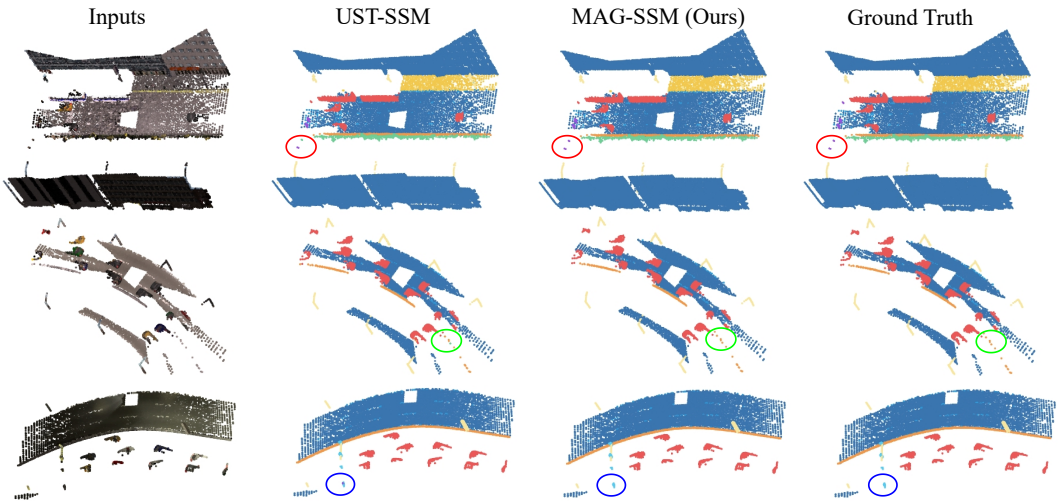
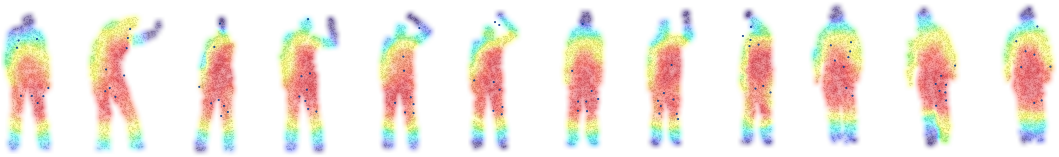
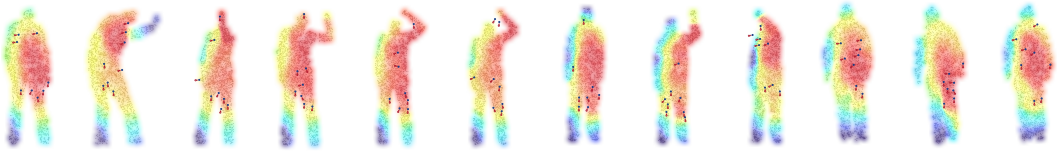


Figure 3: Qualitative comparison with state-of-the-art methods (UST-SSM) on the Synthia 4D dataset. Erroneous estimation results are highlighted with circles.



(a) UST-SSM Responses and Sampling Anchors

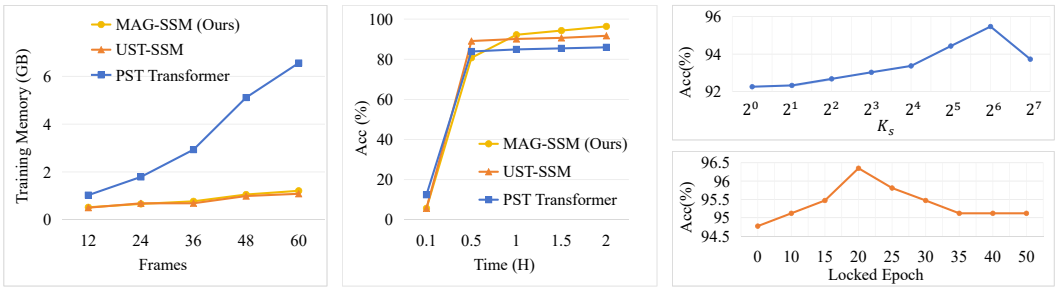


(b) MAG-SSM Responses and Adaptive Anchors

Figure 4: Visualization of feature responses and sampling anchors on a representative MSR-Action3D sequence. (a) UST-SSM responses with the original sampling anchors. (b) MAG-SSM responses with MSTS-adapted anchors.

2025). To ensure a fair comparison, all methods in Table 2 use only 4D point cloud sequences without RGB images or auxiliary 2D encoders, while the SSM-based methods are integrated into the same P4Transformer-based action segmentation framework. MAG-SSM achieves the best Acc and F1 scores while remaining competitive on Edit, demonstrating its effectiveness in improving frame-level recognition accuracy and temporal segment quality. Compared with PPTr+C2P, MAG-SSM improves Acc by 0.7 points and F1@10/25/50 by 0.2/0.7/1.8 points, showing gains across overlap thresholds. It also substantially outperforms the reproduced UST-SSM and Mamba4D baselines, indicating stronger temporal segmentation under the shared framework. Additional sequence-level qualitative comparisons are provided in Appendix C.3.

4D Semantic Segmentation We evaluate MAG-SSM on Synthia 4D against representative convolutional methods (Liu et al., 2019; Fan et al., 2021b), Transformer-based methods (Fan et al., 2021a; 2023), and recent SSM-based approaches (Liu et al., 2025; Li et al., 2025a; Song et al., 2025). As shown in Table 3, MAG-SSM achieves 84.42% mIoU, slightly above PST-Transformer and UST-SSM. Although the margin on Synthia 4D is smaller than those on the action-oriented benchmarks, it still shows that the proposed design transfers to dense point-level semantic understanding. Figure 3 further shows that MAG-SSM produces predictions more consistent with the ground truth, particularly around local structures and semantic boundaries. Compared with UST-SSM, it reduces isolated local errors in the highlighted regions and preserves more spatially coherent semantic layouts. Additional qualitative comparisons illustrating local semantic consistency are provided in Appendix C.2.



(a) (b) (c)

Figure 5: Efficiency and routing analyses on MSR-Action3D. (a) Training memory across sequence lengths (batch size 1). (b) Accuracy versus training time at 36 frames (batch size 16). (c) Effects of K_s (top) and priority-locking epoch (bottom).

Table 4: Component-level ablation study on MSR-Action3D (Acc^1) and HOI4D (Acc^2).

MSTS	MSR-Scanning	OGLF	Acc^1 (%)	Acc^2 (%)
×	×	×	94.77	77.17
✓	×	×	96.17	78.41
×	✓	×	95.47	77.63
✓	✓	×	97.21	79.65
✓	×	✓	96.86	80.31
✓	✓	✓	**97.91**	**81.78**

4.3 Ablation Studies and Analyses

We conduct component-level ablations on MSR-Action3D and HOI4D, while the internal design choices are analyzed on MSR-Action3D. Unless otherwise specified, the latter experiments adopt the 24-frame setting, with UST-SSM as the baseline. As shown in Table 4, MSTS improves the baseline on both datasets, validating motion-adaptive neighborhood construction. Combining it with MSR-Scanning's priority-aware sequence organization further boosts accuracy to 97.21% on MSR-Action3D and 79.65% on HOI4D. Adding OGLF to MSTS raises results from 96.17% to 96.86% and 78.41% to 80.31%, respectively. The MAG-SSM yields the best performance, demonstrating the complementary effects of these three adaptive modules. Further module-specific ablations of MSTS, MSR-Scanning, and OGLF are provided in Appendix D.

Figure 4 shows that MAG-SSM produces more concentrated responses around changing body structures than UST-SSM, while its adaptive anchors follow the evolving geometry across frames, consistent with the role of MSTS in adapting local receptive fields to motion-dependent geometric variations. Figure 5 analyzes the efficiency and routing behavior of MAG-SSM: as shown in Figure 5(a,b), MAG-SSM exhibits memory growth close to UST-SSM and substantially below PST-Transformer, while achieving higher accuracy than UST-SSM under the 36-frame setting with a moderate increase in training time; Figure 5(c) shows that $K_s = 64$ yields the best accuracy of 95.47%, whereas overly fine grouping degrades performance, and locking the group priority at epoch 20 achieves the highest accuracy of 96.35%, supporting a moderate warm-up before priority fixation.

5 Conclusion

We presented MAG-SSM, a motion-adaptive geometric state-space model for 4D point cloud video understanding. MAG-SSM coordinates geometric adaptation across local representation, sequence organization, and geometric recovery within an SSM backbone. MSTS learns bounded offsets to adapt receptive fields to evolving geometry; MSR-Scanning organizes motion-adaptive representations into priority-aware semantic sequences; and OGLF incorporates aligned offsets as geometric priors during local recovery. Experiments across action recognition, 4D semantic segmentation, and action segmentation show consistent improvements over baselines with computational efficiency.

AI USE STATEMENT

In this work, we used generative AI tools, including OpenAI ChatGPT, for readability enhancement, structural organization, code inspection, English grammar checking, and caption refinement. We did not use generative AI to generate experimental data or reported results, and other mandatory-disclosure tasks not applicable to this work were omitted. We have reviewed and verified all AI-assisted work, including technical descriptions, code suggestions, and the interpretation of experimental results. All experiments and reported results were independently produced and validated by the authors. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

The main paper specifies the datasets, evaluation protocols, input configurations, and core training settings used in our experiments. Appendix A provides complete pseudocode for MAG-SSM and its three components, including tensor alignment, multi-scale processing, and the two-stage priority-locking procedure. Appendix B further reports task-specific optimization and evaluation settings, loss functions, inference aggregation, baseline-result provenance, and the controlled HOI4D reproduction protocol. Together, these details specify the model execution, training, and evaluation procedures required for reproducing the reported results.

REFERENCES

Yizhak Ben-Shabat, Oren Shrout, and Stephen Gould. 3DInAction: Understanding human actions in 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19978–19987, June 2024.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084, June 2019.

Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4D transformer networks for spatio-temporal modeling in point cloud videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14204–14213, 2021a.

Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. PSTNet: Point spatio-temporal convolution on point cloud sequences. In *International Conference on Learning Representations*, 2021b.

Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point spatio-temporal transformer networks for point cloud video modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2): 2181–2192, 2023.

Mingrong Gong, Chaoqi Chen, Luyao Tang, Yuxi Wang, and Sergio Escalera. 4D point cloud segmentation via active test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21315–21323, 2026.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3D: Enhancing local features for 3D point cloud analysis via state space model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 4995–5004, 2024.

Junsheng Huang, Shengyu Hao, Bo-Cheng Hu, Hongwei Wang, and Gaoang Wang. Understanding dynamic scenes in ego centric 4D point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5031–5039, 2026.

Linglin Jing, Ying Xue, Xu Yan, Chaoda Zheng, Dong Wang, Ruimao Zhang, Zhigang Wang, Hui Fang, Bin Zhao, and Zhen Li. X4D-SceneFormer: Enhanced scene understanding on 4D point cloud videos through cross-modal knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2670–2678, 2024.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. VideoMamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, pp. 237–255, 2024.

Peiming Li, Ziyi Wang, Yulin Yuan, Hong Liu, Xiangming Meng, Junsong Yuan, and Mengyuan Liu. UST-SSM: Unified spatio-temporal state space models for point cloud video modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6738–6747, 2025a.

Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3D points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 9–14, 2010.

Yinghui Li, Qianyu Zhou, Di Shao, Hao Yang, Ye Zhu, Richard Dazeley, and Xuequan Lu. DA-PointMamba: Domain adaptive point mamba for point cloud completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6653–6661, 2026.

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*, volume 39, pp. 5092–5100, 2025b.

Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. PointMamba: A simple state space model for point cloud analysis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 32653–32677, 2024.

Min Lin, Gangwei Xu, Yun Wang, Xianqi Wang, and Xin Yang. FlowMamba: Learning point cloud scene flow with global motion propagation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5271–5279, 2025.

Jiuming Liu, Jinru Han, Lihao Liu, Angelica I. Aviles-Rivero, Chaokang Jiang, Zhe Liu, and Hesheng Wang. Mamba4D: Efficient 4D point cloud video understanding with disentangled spatial-temporal state space models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17626–17636, 2025.

Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. MeteorNet: Deep learning on dynamic 3D point cloud sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9246–9255, 2019.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual state space model. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103031–103063, 2024.

Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4D egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21013–21022, 2022.

Yunze Liu, Junyu Chen, Zekai Zhang, Jingwei Huang, and Li Yi. LeaF: Learning frames for 4D point cloud sequence understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 604–613, 2023.

Baixuan Lv, Yaohua Zha, Tao Dai, Yuerong Xue, Ke Chen, and Shu-Tao Xia. Adapting pre-trained 3D models for point cloud video understanding via cross-frame spatio-temporal perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12413–12422, June 2025.

Zhiqiang Shen, Xiaoxiao Sheng, Hehe Fan, Longguang Wang, Yulan Guo, Qiong Liu, Hao Wen, and Xi Zhou. Masked spatio-temporal structure prediction for self-supervised learning on point cloud videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16580–16589, 2023a.

Zhiqiang Shen, Xiaoxiao Sheng, Longguang Wang, Yulan Guo, Qiong Liu, and Xi Zhou. PointCMP: Contrastive mask prediction for self-supervised learning on point cloud videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1212–1222, 2023b.

Xiaoxiao Sheng, Zhiqiang Shen, Gang Xiao, Longguang Wang, Yulan Guo, and Hehe Fan. Point contrastive prediction with semantic clustering for self-supervised learning on point cloud videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16515–16524, October 2023.

Shaqing Song, Kuo Tang, and Ying Zhang. Pst-mamba: Spatio-temporal selective state fusion for effective point cloud video understanding with state space models. *Image and Vision Computing*, pp. 105785, 2025.

Jie Wang, Tingfa Xu, Lihe Ding, Xinjie Zhang, Long Bai, and Jianan Li. PVNeXt: Rethinking network design and temporal motion for point cloud video recognition. In *International Conference on Learning Representations*, 2025.

Hao Wen, Yunze Liu, Jingwei Huang, Bo Duan, and Li Yi. Point primitive transformer for long-term 4D point cloud video understanding. In *European Conference on Computer Vision*, pp. 19–35, 2022.

Yiteng Xu, Kecheng Ye, Xiao Han, Yiming Ren, Xinge Zhu, and Yuexin Ma. A unified framework for human-centric point cloud video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1155–1164, June 2024.

Hao Yang, Qianyu Zhou, Haijia Sun, Xiangtai Li, Fengqi Liu, Xuequan Lu, Lizhuang Ma, and Shuicheng Yan. PointDGMamba: Domain generalization of point cloud classification via generalized state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9193–9201, 2025.

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*, volume 39, pp. 10121–10130, 2025.

Zhuoyang Zhang, Yuhao Dong, Yunze Liu, and Li Yi. Complete-to-partial 4D distillation for self-supervised point cloud sequence representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17661–17670, 2023.

Jia-Xing Zhong, Kaichen Zhou, Qingyong Hu, Bing Wang, Niki Trigoni, and Andrew Markham. No pain, big gain: Classify dynamic point cloud sequences with static models by fitting feature-level space-time surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8510–8520, June 2022.

Zhi Zuo, Pan Gao, and Manoranjan Paul. Point long-term locality-aware transformer for point cloud video understanding. In *Proceedings of the 7th ACM International Conference on Multimedia in Asia*, pp. 1–8, 2025.

A Pseudocode of the Overall Framework

Algorithmic Workflow and Implementation Details. Algorithms 1–4 present the complete execution order of MAG-SSM. For each multi-scale branch, a shared temporal index set is used to gather E_s, P_s, Q_s, and ΔP_s, ensuring that token features, positional metadata, and offset priors remain pointwise aligned. In MSTS, both $\mathcal{A}_{\mathrm{sp}}$ and $\mathcal{A}_{\mathrm{temp}}$ are implemented by max pooling. The remaining operations in Algorithms 2–4 follow the corresponding module definitions, while the explicit pseudocode clarifies their execution order and tensor dependencies.

Algorithm 1 Overall Framework of MAG-SSM

Input: Point cloud video $\mathcal{X} = \{P_t^0\}_{t=1}^{T}$
Output: Task prediction $\widehat{y}$

$\{F_t^0\} \leftarrow \mathrm{P4DConv}(\mathcal{X})$
$\{A_t, \widetilde{A}_t, F_t^d\} \leftarrow \mathrm{MSTS}(\{P_t^0, F_t^0\})$ ▷ Algorithm 2
$(E_s, P_s, \widetilde{P}_s, \nu_s) \leftarrow \text{MSR-Scanning}(\{A_t, \widetilde{A}_t, F_t^d\})$ ▷ Algorithm 3
$\Delta P_s \leftarrow \widetilde{P}_s - P_s$, $Q_s \leftarrow [P_s \| \nu_s / T_d]$
for each temporal scale r_s **do**
  $(E^{(r_s)}, P^{(r_s)}, Q^{(r_s)}, \Delta P^{(r_s)}) \leftarrow \mathrm{Subsample}_{r_s}(E_s, P_s, Q_s, \Delta P_s)$
  $U^{0,(r_s)} \leftarrow E^{(r_s)} + \phi_{\mathrm{4D}}(Q^{(r_s)})$
  for each encoder block l **do**
    $U^{l,(r_s)} \leftarrow \mathrm{OGLF}(U^{l-1,(r_s)}, P^{(r_s)}, \Delta P^{(r_s)})$ ▷ Algorithm 4
  end for
  $h^{(r_s)} \leftarrow \mathrm{MaxPool}(U^{D,(r_s)})$
end for
$\widehat{y} \leftarrow \phi_{\mathrm{task}}\big(\mathrm{Concat}(\{h^{(r_s)}\})\big)$
return $\widehat{y}$

Algorithm 2 Motion-Adaptive Spatio-Temporal Sampling

Input: Coordinates and features $\{P_t^0, F_t^0\}_{t=1}^{T}$
Output: $\{A_t, \widetilde{A}_t, F_t^d\}_{t=1}^{T_d}$

for each output frame t **do**
  $I_t \leftarrow \mathrm{FPS}(P_t^0, M),\ A_t \leftarrow P_t^0[I_t],\ H_t \leftarrow F_t^0[I_t]$
  $\widetilde{A}_t \leftarrow A_t + \mathrm{clip}(\mathrm{Offset}(A_t, H_t + \phi_{\mathrm{pos}}(A_t)), -\eta r, \eta r)$
  for each $\tau \in \Omega_t$ **do**
    $\mathcal{J}_{t,\tau} \leftarrow \mathrm{BallQuery}(\widetilde{A}_t, P_\tau^0, r, K_{\mathrm{M}})$
    $D_{t,\tau} \leftarrow [P_\tau^0[\mathcal{J}_{t,\tau}] - \widetilde{A}_t \| \tau - t]$
    $S_{t,\tau} \leftarrow \mathcal{A}_{\mathrm{sp}}\big(\phi_g\big(\phi_f(F_\tau^0[\mathcal{J}_{t,\tau}]) + \phi_d(D_{t,\tau})\big)\big)$
  end for
  $F_t^d \leftarrow \mathcal{A}_{\mathrm{temp}}(\{S_{t,\tau}\}_{\tau \in \Omega_t})$
end for
return $\{A_t, \widetilde{A}_t, F_t^d\}_{t=1}^{T_d}$

Two-Stage Priority Locking. To obtain a stable serialization order, MSR-Scanning employs a two-stage priority-locking strategy with a 20-epoch warm-up. During warm-up ($e \le E_w$, $E_w = 20$), saliency-guided feature modulation is already active, but tokens retain their original frame-major order. This allows the routing and scoring heads to learn meaningful semantic assignments before their statistics are used to determine the permanent scan order. The batch-averaged saliency of group k at epoch e is

$$\widehat{s}_k^{(e)} = \frac{1}{B} \sum_{b=1}^{B} s_{b,k}^{(e)} \tag{15}$$

The batch estimate may fluctuate because routing assignments and feature representations continue to evolve during early training. We therefore accumulate an exponential moving average:

$$\bar{s}_k^{(e)} = \beta \bar{s}_k^{(e-1)} + (1-\beta)\widehat{s}_k^{(e)} \tag{16}$$

The EMA suppresses short-term batch variations and provides a more stable estimate of the relative importance of each semantic group.

At the end of epoch E_w, groups are ranked in descending order of their accumulated saliency, and the resulting ranking is converted into a fixed priority map:

$$\begin{gathered} \pi = \mathrm{Argsort}_{k \in \{1,\dots,K_s\}} \left(-\bar{s}_k^{(E_w)} \right) \\ \mathcal{M}_{\mathrm{lock}}(\pi_j) = j \end{gathered} \tag{17}$$

Algorithm 3 Motion-Adaptive Semantic Routing Scanning

Input: MSTS outputs $\{A_t, \widetilde{A}_t, F_t^d\}_{t=1}^{T_d}$, epoch e, locking epoch $E_w = 20$
Output: Serialized features and aligned metadata

Flatten aligned quantities into Z, P, $\widetilde{P}$, and ν
$\Pi \leftarrow \mathrm{ST\text{-}GumbelSoftmax}(\phi_r(Z))$ ▷ Discrete semantic routing
$C \leftarrow \mathrm{GroupPrototype}(\mathrm{SG}(Z), \Pi)$, $s \leftarrow \phi_s(C)$
$E \leftarrow [Z \odot (1 + \lambda s_\ell) \| \Pi E_p]$ ▷ Saliency-guided enhancement
if $e \le E_w$ **then**
    $\bar{s}^{(e)} \leftarrow \beta \bar{s}^{(e-1)} + (1-\beta)\,\mathrm{MeanBatch}(s^{(e)})$, $\rho \leftarrow (1, \ldots, L)$
    if the current epoch ends and $e = E_w$ **then**
        $\pi \leftarrow \mathrm{Argsort}(-\bar{s}^{(E_w)})$, $\mathcal{M}_{\mathrm{lock}}(\pi_j) \leftarrow j$ ▷ Eq. equation 17, lock EMA
    end if
else
    $R \leftarrow \mathcal{M}_{\mathrm{lock}}(\ell)$, $\rho \leftarrow \mathrm{StableArgsort}(R)$ ▷ Eq. equation 18, preserve order
end if
return $(E_s, P_s, \widetilde{P}_s, \nu_s) \leftarrow (E, P, \widetilde{P}, \nu)[\rho]$

Algorithm 4 Offset-Guided Local Geometric Fusion

Input: Tokens U^{l-1}, anchors P_s, offsets ΔP_s
Output: Updated tokens U^l

$\mathcal{N}_K \leftarrow \mathrm{KNN}(P_s, P_s, K)$ ▷ Geometry-based neighborhood recovery
$G^l \leftarrow \mathcal{K}_{\mathrm{pool}}\big(\mathcal{K}_{\mathrm{norm}}(P_s, \mathrm{LN}(U^{l-1}), \mathcal{N}_K)\big)$ ▷ Normalized local aggregation
$\widehat{G}^l \leftarrow \mathrm{LN}(G^l) + \gamma_l \phi_m(\mathrm{SG}(\Delta P_s))$ ▷ Gated offset-prior injection
$\bar{U}^l \leftarrow U^{l-1} + \mathrm{DropPath}\Big(\mathrm{SiLU}(\phi_o(\widehat{G}^l))\Big)$
$U^l \leftarrow \bar{U}^l + \mathrm{DropPath}\big(\mathrm{Mamba}_l(\mathrm{LN}(\bar{U}^l))\big)$ ▷ State-space residual update
return U^l

Higher-saliency groups therefore receive smaller priority indices and are placed earlier in the serialized sequence. Fixing this map after warm-up prevents the scan order from changing repeatedly during later optimization, allowing the state-space encoder to learn under a consistent sequence organization. For a token assigned to semantic group ℓ_i, its locked priority is

$$R_i = \mathcal{M}_{\mathrm{lock}}(\ell_i) \tag{18}$$

Tokens are stably sorted according to R_i. Thus, the priority map controls the order between semantic groups, while stable sorting preserves the original frame-major order among tokens assigned to the same group. After locking, the EMA statistics and $\mathcal{M}_{\mathrm{lock}}$ are no longer updated, whereas the routing and scoring heads remain optimized through the task loss and continue to determine token assignments and feature modulation.

Input Adaptivity after Priority Locking. The locked EMA priority should be distinguished from the input-dependent routing process. After the warm-up stage, $\mathcal{M}_{\mathrm{lock}}$ only fixes the relative ordering among semantic groups and does not determine which tokens belong to each group. For an input sequence $\mathcal{X}$, the routing head still predicts token-specific assignments $\ell_i(\mathcal{X})$ from the current MSTS-enhanced representations, and the corresponding token priority is

$$R_i(\mathcal{X}) = \mathcal{M}_{\mathrm{lock}}(\ell_i(\mathcal{X}))\,. \tag{19}$$

Therefore, different input sequences may produce different group assignments, group occupancies, and serialized token compositions even though they share the same inter-group priority map. The EMA-based locking is introduced to stabilize the relative ordering between groups and avoid frequent permutation changes during later optimization, while token routing and saliency-conditioned feature modulation remain input-dependent. Accordingly, MSR-Scanning can be viewed as instance-conditioned semantic routing under a globally stabilized group-priority map.

Table 5: Task-specific optimization and evaluation configurations. Here, T and N denote the numbers of frames and points per frame, respectively; CE-LS denotes cross-entropy with label smoothing. The optimizer entries report momentum and weight decay in parentheses.

Setting	MSR-Action3D	Synthia 4D	HOI4D
Evaluation scope	clip-level recognition	point-wise segmentation	frame-wise segmentation
Input (T, N)	$(16/24/36,\ 2{,}048)$	$(3,\ 16{,}384)$	$(150,\ 2{,}048)$
Ball-query radius / neighbors	$0.3/32$	$0.9/32$	$0.9/32$
Epochs / batch size	$50/16$	$50/16$	$50/16$
Optimizer	SGD $(0.9,\ 10^{-4})$	SGD $(0.9,\ 10^{-4})$	SGD $(0.9,\ 10^{-4})$
Initial LR / schedule	0.01; $\times$0.1 at 20, 30	0.01; $\times$0.1 at 20, 30	0.02; $\times$0.5 at 35, 45
LR warm-up / priority locking	$10/20$ epochs	$10/20$ epochs	$10/20$ epochs
Loss	CE-LS (0.1)	weighted masked CE	CE-LS (0.05)
Gradient clipping	5.0	none	3.0
Inference aggregation	video-probability sum	global class statistics	overlap-logit average
Evaluation	video-level Acc	12-class mIoU	Acc, Edit, F1@{10,25,50}

B Additional Experimental Details

B.1 Benchmark Selection and Evaluation Scope

Dataset Selection. We select MSR-Action3D, Synthia 4D, and HOI4D to evaluate MAG-SSM from three complementary perspectives. MSR-Action3D evaluates clip-level action recognition across different temporal lengths, Synthia 4D evaluates dense point-wise semantic segmentation in dynamic outdoor scenes, and HOI4D evaluates frame-wise action segmentation over long sequences. Together, these benchmarks cover global video classification, dense spatial understanding, and long-sequence temporal segmentation. All experiments use PyTorch 2.0.1 with CUDA 11.8 on an NVIDIA RTX 4090 GPU with 24 GB memory.

Model and Routing Configurations. Unless otherwise specified, MSTS uses $K_{\mathrm{A}} = 8$ anchor neighbors for offset prediction and $K_{\mathrm{M}} = 32$ points per frame for cross-frame aggregation. The offset constraint factor is $\eta = 0.25$. MSR-Scanning uses $K_s = 64$ semantic groups, residual factor $\lambda = 0.2$, EMA momentum $\beta = 0.9$, and a fixed Gumbel-Softmax temperature $\tau_g = 1.0$ without annealing. For the MSR-Action3D recognition model, MSTS produces $M = 64$ anchors per output frame and uses temporal radius $m = 1$, corresponding to a three-frame window $\Omega_t = \{t-1, t, t+1\}$. The three sequence branches use subsampling strides $r_s = \{1, 2, 4\}$ with corresponding OGLF neighborhood sizes $K = \{12, 6, 3\}$. For action recognition, each branch is aggregated by global max pooling, after which the branch features are concatenated and passed to the MLP classification head.

Operator Implementations. For the geometry-aware aggregation in Eq. equation 2, let $r_{i,j} = w_{i,j}\phi_{\mathrm{aff}}(v_{i,j})$. The channel-wise exponential soft pooling is implemented as

$$[\mathcal{A}_{\mathrm{geo}}]_c = \frac{\sum_{j\in\mathcal{N}_i} r_{i,j,c}\exp(r_{i,j,c} - m_{i,c})}{\sum_{j\in\mathcal{N}_i}\exp(r_{i,j,c} - m_{i,c})}, \qquad m_{i,c} = \max_{j\in\mathcal{N}_i} r_{i,j,c}, \tag{20}$$

where subtracting $m_{i,c}$ improves numerical stability. For Eq. equation 6, both spatial and temporal aggregation are channel-wise max pooling:

$$\mathcal{A}_{\mathrm{sp}}\left(\{g_{\tau,j}\}_{j=1}^{K_{\mathrm{M}}}\right) = \max_j g_{\tau,j}, \qquad \mathcal{A}_{\mathrm{temp}}\left(\{h_\tau\}_{\tau\in\Omega_t}\right) = \max_{\tau\in\Omega_t} h_\tau. \tag{21}$$

For OGLF, K-Norm normalizes neighbor-center feature residuals using a shared local standard deviation, concatenates the normalized residual with the center feature, and applies learnable affine scaling. Let $\widetilde{u}_i$ and $\widetilde{u}_j$ denote the normalized center and neighbor features. The normalized local response is

$$h_{i,j} = \alpha \odot \left[\frac{\widetilde{u}_j - \widetilde{u}_i}{\sigma_u + \epsilon}\middle\|\widetilde{u}_i\right] + \delta, \tag{22}$$

where σ_u is the shared standard deviation of local neighbor-center residuals, $\epsilon = 10^{-5}$, and α and δ are learnable affine parameters. K-Pool then performs channel-wise exponential weighted aggregation:

$$g_{i,c} = \frac{\sum_{j \in \mathcal{N}_K(i)} h_{i,j,c} \exp(h_{i,j,c})}{\sum_{j \in \mathcal{N}_K(i)} \exp(h_{i,j,c})}. \tag{23}$$

Task-Specific Protocols. For MSR-Action3D, the clip probabilities belonging to the same video are summed to obtain the video-level prediction. For Synthia 4D, the two clips returned by each training sample are optimized sequentially, and class-weighted cross-entropy is computed only over valid points. For HOI4D, complete sequences are divided into overlapping 24-frame windows with stride 12. During inference, logits at shared frames are averaged, and temporal interpolation is applied when the output length differs from the target sequence length.

B.2 Baseline Result Provenance and Controlled HOI4D Reproduction

For MSR-Action3D and Synthia 4D, the baseline results are taken from the corresponding publications under their established evaluation protocols, including P4Transformer (Fan et al., 2021a), PSTNet (Fan et al., 2021b), and 3DInAction (Ben-Shabat et al., 2024). The reported MAG-SSM results are obtained using our implementation under the settings summarized in Table 5.

For HOI4D, the comparison includes both results retained from the original publications and results reproduced under a controlled action-segmentation framework. To isolate the contribution of the 4D representation backbone, we adopt the P4Transformer-based action-segmentation framework as a common evaluation platform. The official data split, point-cloud input, sliding-window strategy, temporal segmentation head, loss function, training pipeline, and evaluation code remain unchanged, while only the feature backbone is replaced by Mamba4D, UST-SSM or MAG-SSM. Therefore, the differences among these reproduced variants primarily reflect their backbone representation quality rather than changes in the downstream segmentation architecture or evaluation protocol.

The reproduced Mamba4D, UST-SSM, and MAG-SSM models use identical point-cloud inputs and the same 24-frame sliding windows with stride 12. No RGB images, auxiliary 2D encoders, or cross-modal features are used. For MAG-SSM, the routing and scoring heads use one tenth of the backbone learning rate, while the remaining task-level settings follow the shared action-segmentation framework.

B.3 Task-Specific Network Details

HOI4D Action Segmentation. For HOI4D, MAG-SSM is integrated into a frame-wise action segmentation framework. A P4DConv stem first extracts initial point-cloud video features, after which MSTS adaptively relocates the sampling anchors and constructs cross-frame neighborhoods around the relocated anchors to obtain motion-adaptive representations and the corresponding sampling offsets. MSR-Scanning then performs semantic routing and serialization on these representations while consistently reordering the tokens, anchors, and associated offset information. Within each Mamba state-space block, OGLF reconstructs local neighborhoods according to the reordered geometric coordinates and injects the token-aligned MSTS offsets into the local geometric features as adaptive-sampling priors, followed by the Mamba state-space update. After the stacked Mamba blocks, the tokens are restored to their corresponding frames and aggregated within each frame using max pooling to obtain frame-wise features, which are finally passed to an MLP prediction head to produce frame-wise action logits for temporal action segmentation. The reproduced Mamba4D, UST-SSM, and MAG-SSM variants use the same point-cloud input, prediction head, training pipeline, and evaluation procedure, with only the representation backbone replaced. MSR-Scanning then organizes these motion-adaptive representations into priority-aware semantic sequences while consistently reordering the tokens, anchors, and associated offsets. OGLF reconstructs local neighborhoods from the reordered geometry and incorporates the aligned MSTS offsets as geometric priors before each Mamba update.

Synthia 4D Semantic Segmentation. For Synthia 4D, MAG-SSM adopts a hierarchical encoder–decoder architecture, in which four MSTS-based encoding stages progressively extract and downsample spatio-temporal features to form multi-scale motion-adaptive representations together with

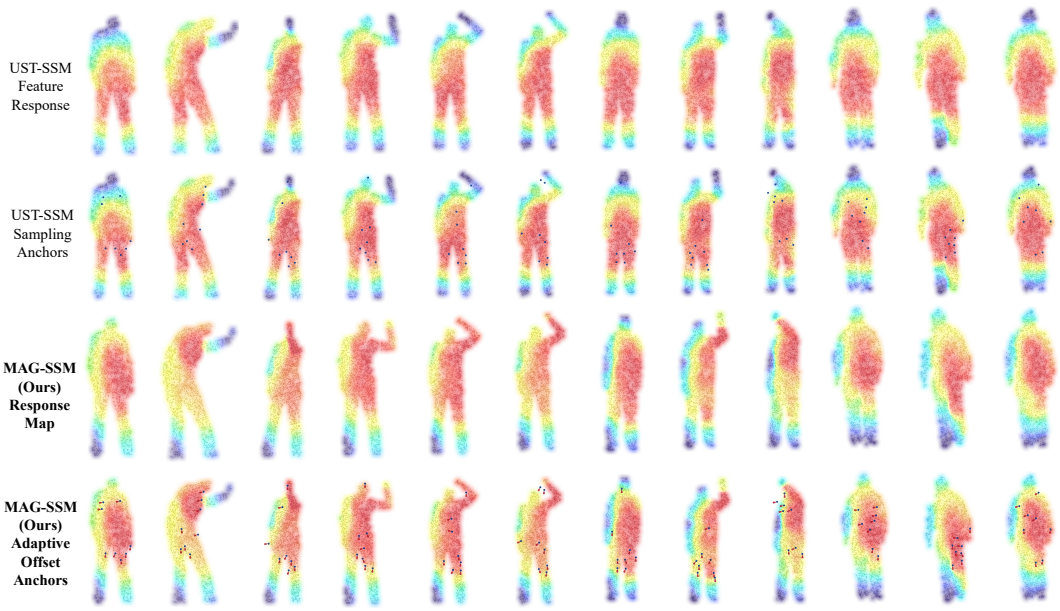


Figure 6: Visualization of feature responses and sampling anchors on a representative MSR-Action3D sequence. From top to bottom, the rows show the feature responses of UST-SSM, the sampling anchors of UST-SSM, the response maps of MAG-SSM, and the adaptive offset anchors generated by MSTS. Warmer colors denote stronger responses.

the corresponding sampling offsets. The deepest encoded features are then flattened into a token sequence and semantically routed and serialized by MSR-Scanning, while preserving the correspondence among tokens, spatial coordinates, and MSTS offsets. Within each subsequent Mamba state-space block, OGLF reconstructs local neighborhood relations according to the reordered geometric coordinates and injects the aligned MSTS offsets into the local representations as geometric priors, followed by Mamba-based state-space modeling. The processed tokens are then inverse-permuted to restore the original point-wise correspondence and passed to a P4DTransConv decoder with skip connections to progressively recover the spatial resolution. A final convolution layer produces point-wise semantic predictions. The MAG-SSM result follows the protocol summarized above, while the baseline results in Table 3 are taken from their corresponding publications under the established Synthia 4D evaluation protocols.

C Additional Qualitative Results

C.1 Motion-Adaptive Sampling Visualization

Figure 6 visualizes the effect of applying motion-adaptive relocation before cross-frame neighborhood retrieval. The feature responses of UST-SSM broadly cover the human body, but their spatial distribution remains comparatively diffuse as the pose changes. Its sampling anchors are tied to the original spatial sampling pattern and therefore do not explicitly adjust when the arm, torso, or lower limbs move to different locations.

MAG-SSM exhibits a different response pattern. Across the sequence, strong responses shift with the changing pose and become more concentrated around motion-responsive structures, including the raised arm, the evolving torso contour, and the lower limbs. The corresponding adaptive offset anchors are redistributed toward these regions rather than remaining fixed at their original locations. This indicates that MSTS converts encoded spatio-temporal context into a bounded spatial adjustment of the receptive field.

The relocation is performed before ball-query retrieval so that the same adapted anchor serves as the query reference throughout the temporal window. Under fixed sampling, a query center may

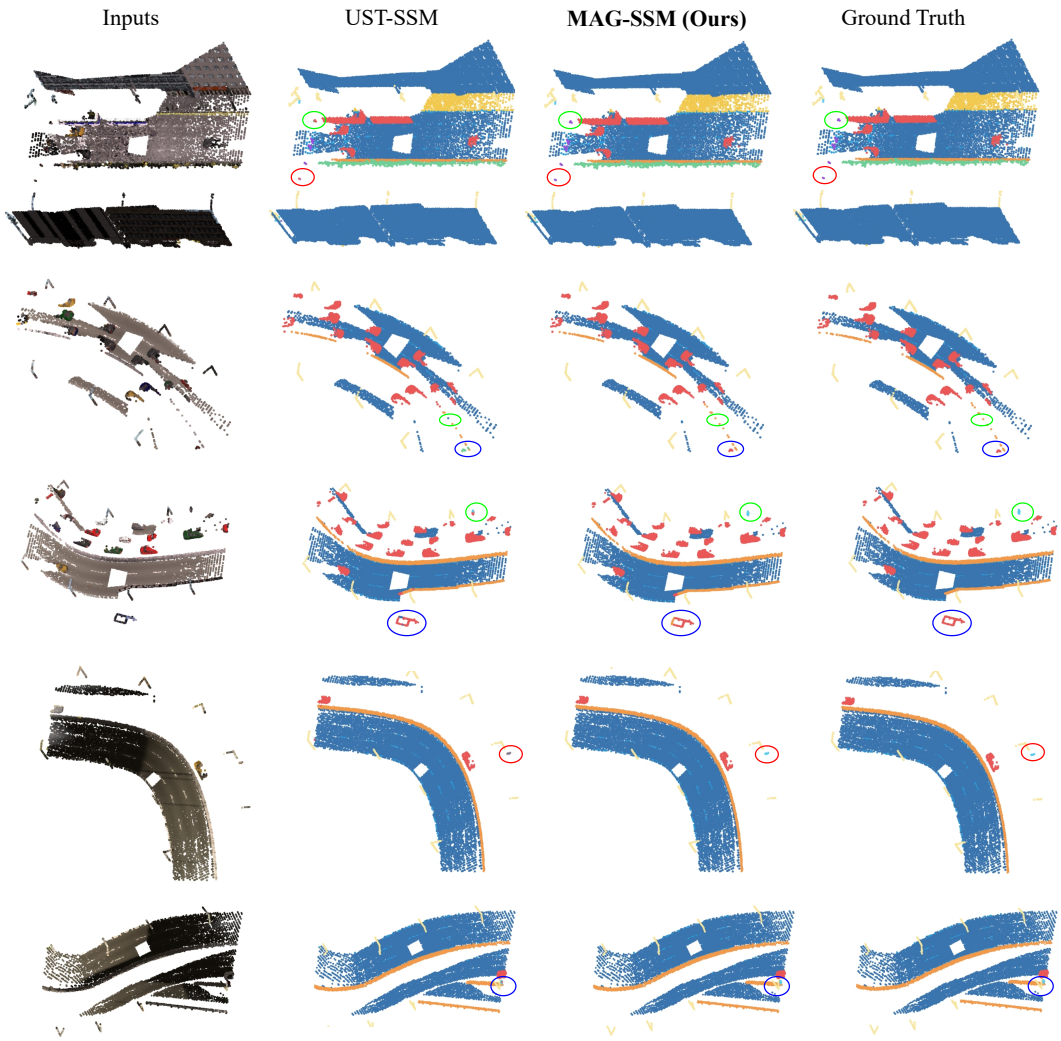


Figure 7: Additional qualitative comparisons on Synthia 4D. From left to right, we show the input point clouds, predictions of UST-SSM, predictions of MAG-SSM, and the ground-truth annotations. Colored circles highlight representative regions containing isolated misclassifications, incomplete structures, or boundary inconsistencies.

collect different physical structures as the body moves, weakening the consistency of the aggregated evidence. Relocating the query center allows the cross-frame neighborhoods to better follow local geometric variation, while the offset bound prevents excessive displacement outside the intended support.

The improvement shown in the response maps should therefore be interpreted as a more structured redistribution of sampling support, rather than merely an increase in activation magnitude. The resulting motion-adaptive tokens provide more relevant inputs to MSR-Scanning, and their aligned offsets are subsequently reused by OGLF as adaptive geometric priors. This visualization supports the coordinated role of MSTS, MSR-Scanning, and OGLF in preserving informative dynamic context during state propagation.

C.2 Synthia 4D Semantic Segmentation

Figure 7 presents additional semantic-segmentation results on Synthia 4D. Across the five scenes, MAG-SSM generally produces cleaner semantic regions and fewer isolated label errors than UST-

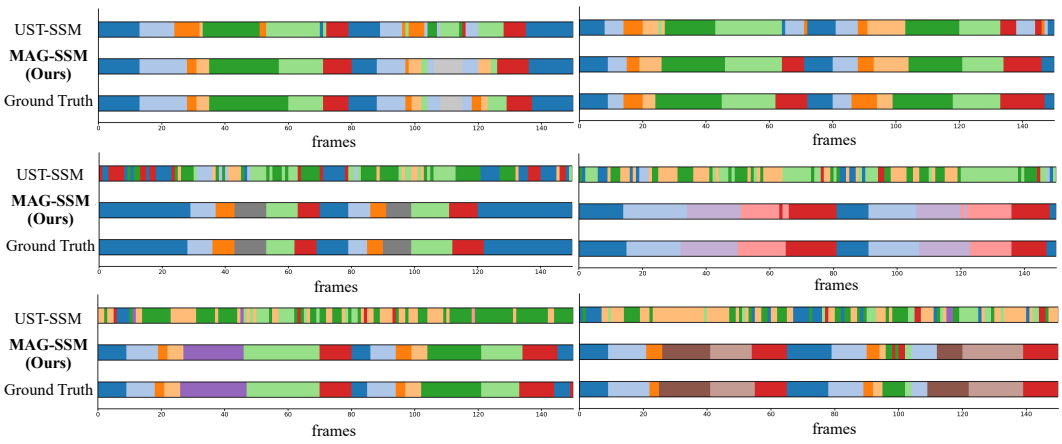


Figure 8: Additional temporal action-segmentation results on HOI4D. Each panel shows one complete sequence, and each sequence contains the predictions of UST-SSM and MAG-SSM together with the ground-truth action timeline. Different colors denote different action classes.

SSM. The differences are particularly visible in the circled regions, including sparsely observed objects, thin structures, and areas close to semantic boundaries. UST-SSM occasionally produces small disconnected predictions or assigns locally inconsistent labels, whereas the corresponding MAG-SSM predictions are more spatially complete and better aligned with the ground-truth structures.

These improvements are consistent with the complementary roles of MSTS, MSR-Scanning, and OGLF. MSTS relocates local sampling centers according to the encoded spatio-temporal context, allowing cross-frame neighborhoods to better adapt to local geometric variations instead of relying on fixed spatial supports that may become misaligned as objects and viewpoints evolve. This enables features near semantic boundaries, thin structures, and sparsely observed regions to retain more relevant local evidence. MSR-Scanning then organizes the resulting motion-adaptive representations according to task-relevant group saliency, providing a more coherent sequence for long-range state propagation and reducing interference among heterogeneous dynamic contexts. OGLF further reconstructs local neighborhoods after one-dimensional serialization and incorporates the aligned MSTS offsets as adaptive geometric priors, so that the local fusion stage remains aware of the preceding sampling adjustment. The more complete and spatially coherent predictions in Figure 7 are therefore consistent with the joint effects of motion-adaptive neighborhood alignment, saliency-aware sequence organization, and offset-guided local geometric recovery, which together reduce isolated prediction noise and incomplete semantic regions.

C.3 HOI4D Action Segmentation

Figure 8 compares temporal predictions on six additional HOI4D sequences. The most apparent difference is the reduction of short and fragmented action segments. In several examples, UST-SSM frequently switches between classes within intervals that correspond to a single ground-truth action. MAG-SSM instead recovers longer continuous segments and preserves the dominant action order more consistently.

The improvement is especially clear in the second and third rows. UST-SSM produces dense label oscillations and multiple spurious transitions, whereas MAG-SSM reconstructs the principal action phases and yields temporal structures that are substantially closer to the ground truth. In the first row, where both models already capture much of the overall action order, MAG-SSM still reduces several internal fragmentations and provides more consistent segment durations.

The observed temporal improvements are consistent with the coordinated roles of MSR-Scanning, MSTS, and OGLF. A conventional frame-major serialization may separate tokens that describe similar semantic or motion patterns, even when these tokens are closely related to the same action phase.

Table 6: Micro-ablation of the offset-prediction cues and constraint factor in MSTS.

Setting	(I)	(II)	(III)	(IV)	(V)	(VI)	(VII)	(VIII)	(IX)
Pos.	×	✓	✓	×	✓	×	✓	✓	✓
Sem.	×	✓	×	✓	×	✓	✓	✓	✓
η	–	–	–	–	0.25	0.25	0.125	0.25	0.5
Acc. (%)	94.77	91.64	86.41	83.97	95.47	94.43	95.12	**96.17**	95.82

Table 7: Ablation of saliency-aware ranking and residual enhancement factor λ in MSR-Scanning.

Setting	(I)	(II)	(III)	(IV)	(V)
Ranking	×	×	✓	✓	✓
λ	0.1	0.2	0.1	0.2	0.3
Acc. (%)	87.11	89.89	94.08	**95.47**	95.12

Table 8: Ablation of local aggregation and offset prior in OGLF.

Setting	Baseline	+ Local KNN	OGLF
KNN	×	✓	✓
ΔP_s	×	×	✓
Acc. (%)	94.77	95.81	**96.86**

MSR-Scanning alleviates this problem by assigning motion-adaptive tokens to semantic groups and organizing them according to task-relevant saliency. After the group priority is stabilized, informative contexts are arranged more coherently during state propagation, which reduces interference from frequently alternating heterogeneous tokens and helps the state-space encoder maintain consistent action representations over longer temporal intervals. MSTS improves the quality of these tokens before serialization by relocating local sampling centers according to encoded spatio-temporal context. This allows the sampled neighborhoods to better adapt to cross-frame geometric changes, which is particularly important for HOI4D actions involving moving hands, manipulated objects, and continuously changing hand-object relations. OGLF then reconstructs local geometric relations that may be weakened by 1D serialization and incorporates the aligned MSTS offsets as adaptive geometric priors during local aggregation. This helps explain why MAG-SSM reduces rapid label oscillations, fragmented short segments, and spurious action transitions while producing temporal predictions that more closely follow the major action phases in the ground truth.

D Additional Ablation Studies

D.1 Effects of MSTS Design

We conduct detailed ablations of MSTS on the MSR-Action3D dataset to analyze the effects of offset-prediction cues and the constraint factor η. As shown in Table 6, using unconstrained offsets leads to a clear performance drop, suggesting that free-form sampling shifts may introduce unstable or noisy neighborhoods. When positional and semantic cues are jointly used with a moderate constraint factor, MSTS achieves the best accuracy of 96.17%. This indicates that adaptive receptive fields benefit from jointly modeling geometric and semantic context, while the offset range should be properly constrained to avoid excessive displacement. Among the tested settings, $\eta = 0.25$ provides the best balance between geometric flexibility and neighborhood stability.

D.2 Effects of MSR-Scanning Design

We evaluate the design of MSR-Scanning on the MSR-Action3D dataset by varying the saliency-aware ranking strategy and residual enhancement factor λ. As shown in Table 7, residual modulation alone yields limited gains, whereas introducing saliency-aware ranking substantially improves performance by reorganizing tokens according to learned group importance. Combining ranking with moderate residual enhancement achieves the best accuracy of 95.47% at $\lambda = 0.2$. These results indicate that sequence reordering and feature modulation provide complementary benefits, while overly strong residual enhancement may disturb the learned sequential representation.

D.3 Effects of OGLF Design

We perform ablations of OGLF on the MSR-Action3D dataset to determine whether its improvement arises from the additional local feature aggregation or from the further use of MSTS offsets as geometric priors. As shown in Table 8, introducing the additional KNN-based local aggregation used by OGLF improves the accuracy from 94.77% to 95.81%. Note that the baseline Mamba block already contains a KNN operation; OGLF introduces an additional neighborhood construction specifically for local feature aggregation and geometric recovery after serialization. Further incorporating the aligned MSTS offsets ΔP_s increases the accuracy to 96.86%. These results show that OGLF benefits from both additional local geometric aggregation and the geometric prior provided by aligned MSTS offsets.

E Limitations and Societal Impacts

E.1 Limitations

MAG-SSM is designed to provide an effective and efficient framework for motion-adaptive 4D point cloud representation learning. Its current implementation uses a bounded offset for each sampling anchor, a fixed number of semantic groups, and a predefined priority-locking epoch. These design choices provide a simple and consistent implementation, while future work may explore more flexible offset parameterization, adaptive semantic-group capacity, and data-dependent priority assignment to better accommodate diverse scene dynamics.

The current evaluation focuses on three established benchmarks covering action recognition, 4D semantic segmentation, and action segmentation. Although these experiments demonstrate the effectiveness of MAG-SSM across different prediction granularities, broader evaluation on additional datasets, sensing conditions, and denser point clouds would provide a more complete understanding of its generalization. In addition, OGLF currently reconstructs local geometry using KNN-based neighborhoods and aligned sampling offsets. Extending this design with multi-resolution neighborhoods or richer geometric priors may further benefit scenes containing highly complex local structures.

E.2 Societal Impacts

Positive Societal Impacts. Motion-adaptive 4D point cloud understanding can support applications that require reliable perception of dynamic geometry, including robotics, autonomous navigation, augmented reality, industrial inspection, assistive systems, and human–robot interaction. By improving the organization of dynamic context and the preservation of local geometric information, MAG-SSM may contribute to more accurate recognition of human actions, moving objects, and temporal scene changes.

Potential Considerations. As with other perception methods for dynamic scenes, applications involving human activity data should consider privacy, consent, and appropriate data governance. Performance may also vary when the sensing conditions differ substantially from the training data. These considerations are general to learning-based 4D perception systems and do not arise from a specific component of MAG-SSM.

Responsible Use. Practical deployment should follow standard responsible-AI practices, including secure data handling, evaluation under representative sensing conditions, uncertainty assessment, and human oversight in safety-relevant applications. Future work may further investigate model calibration, robustness under sensor variation, and privacy-preserving 4D representation learning.