Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Calculus on Clouds: Discretized Integration and Depth-wise Evolution for Point Cloud Segmentation

Zhige Ying
Hubei University of Technology
Zhe Wu
Naval University of Engineering
Huizheng Yuan
Hubei University of Technology
Cheng Zhang
Hubei University of Technology
Qiangsheng Zhang
finn0101@qq.com

Nuclear and Radiation Safety Center

Research Article

Keywords: Point cloud, State space model (SSM), Transformer, Discrete integration, 3D scene segmentation

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11288759/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Calculus on Clouds: Discretized Integration and Depth-wise Evolution for Point Cloud Segmentation

Zhige Ying[1†], Zhe Wu[2†], Huizheng Yuan[1], Cheng Zhang[1], Hande Chen[1], QiangSheng Zhang[3*]

[1]School of Innovation and Entrepreneurship, Hubei University of Technology, Wuhan, Hubei, China.

[2]College of Energy and Power Engineering, Naval University of Engineering, Wuhan, Hubei, China.

[3*]Nuclear and Radiation Safety Center, MEE, Beijing, 100082, China.

*Corresponding author(s). E-mail(s): finn0101@qq.com;

†These authors contributed equally to this work.

Abstract

3-D point clouds are unordered, irregularly sampled, and variable in density, which makes it difficult for unified segmentation models to preserve local geometry while incorporating global context. We propose Calculus on Clouds (COC), a geometry-state operator integrated into the OneFormer3D query decoder for joint semantic, instance, and panoptic segmentation. COC contains three coordinated components. Lean Gated Discretized Integration (LGDI) represents regional samples with a positive scalar measure and combines a learned content integral with contracted first-order Jacobian norms and paired second-order cross-product descriptors. State-Aware Gated Cross-Attention (SAGCA) couples a positive low-rank local read with a compact permutation-invariant state bank and query-conditioned gated fusion. Depth-wise Selective Scan Aggregation (DSSA) performs bounded bidirectional state evolution along the naturally ordered decoder-depth axis and refines the final query representation. The formulation avoids imposing an arbitrary scan order on point tokens and uses contracted radial descriptors for the stated geometric properties. Experiments on ScanNetV2 and S3DIS show improvements over the corresponding OneFormer3D references across semantic, instance, and panoptic metrics. Component ablations and operator-level numerical checks further characterize the effects and limits of the proposed design.

Keywords: Point cloud, State space model (SSM), Transformer, Discrete integration, 3D scene segmentation

1 Introduction

Three-dimensional point clouds are a central sensing representation for indoor robots, RGB-D mapping, digital twins, and other automation systems. A perception module in these settings must separate objects, assign semantic labels, and maintain a consistent scene representation for navigation and semantic mapping [3, 6]. Point clouds, however, are unordered, sparse, non-uniformly sampled, and variable in density and scale [2, 4, 5]. These properties make the aggregation of local evidence difficult: a region summary should be permutation invariant, should not be dominated

by sampling density, and should retain geometric variation that is lost by a single mean or maximum.

Existing 3D segmentation pipelines use point-based, voxel-based, or projection-based representations [2, 4, 5]. Point-based models such as PointNet [7], DGCNN [8], KPConv [9], and PointNeXt [10] preserve point-level information, while sparse voxel methods such as MinkowskiNet [14], TorchSparse++ [15], LargeKernel3D [17], and VoxelNeXt [18] trade some geometric resolution for efficient convolution. Recent point-cloud foundation models also investigate language-aligned point representations [1]. Projection-based methods provide efficient 2D processing but can introduce occlusion and geometric ambiguity [19, 20]. Across these representations, local features are still commonly summarized by fixed statistics or learned weights, leaving a gap between permutation-invariant pooling and geometry-aware integration.

Attention-based models provide a flexible way to couple distant regions. Point Transformer [25], Point Transformer V2 [26], Fast Point Transformer [27], and Stratified Transformer [28] demonstrate the value of contextual interaction, but dense query–region attention can be expensive and does not by itself define a compact, addressable scene state. State Space Models (SSMs), including S4 [30], S5 [31], and Mamba [32], offer a complementary view based on stable state evolution. Their point-cloud extensions, such as Point Mamba [33], Point Cloud Mamba [34], PointMamba [35], Pamba [29], and 3D-UMamba [36], show that state propagation is useful for geometric data, but raw point tokens do not possess a canonical scan order.

Query-based segmentation offers a natural interface for unified prediction. DETR [37] and Mask2Former [38, 39] establish query-driven set prediction, while Mask3D [40], SPFormer [41], and OneFormer3D [42] extend this paradigm to 3D scenes. OneFormer3D is particularly relevant because a single decoder supports semantic, instance, and panoptic segmentation. Its query decoder nevertheless leaves three opportunities open: local superpoint aggregation can be more than a fixed statistic, global context can be exposed through an explicit compact state, and the intermediate query trajectory across decoder depth can be used more directly.

We address these opportunities with *Calculus on Clouds* (COC), a geometry-state operator family built on the unified OneFormer3D pipeline [42]. COC applies one operator formulation at three axes with different order structures. For unordered points or voxels within a region, Lean Gated Discretized Integration (LGDI) forms a positive scalar measure and combines a learned content integral with contracted first-order Jacobian norms and paired second-order cross-product descriptors. For query–region interaction, State-Aware Gated Cross-Attention (SAG-CA) combines a positive low-rank kernel read, a mask-preserving softmax path required by iterative mask supervision, and a small permutation-invariant state bank. For the only intrinsically ordered axis, decoder depth, Depth-wise Selective Scan Aggregation (DSSA) performs a stable bidirectional state evolution conditioned on scene descriptors. The three paths use separate operator instances where required by feature width, but share the same measure–kernel–state design rather than introducing unrelated feature extractors.

The design leads to three concrete goals: (i) preserve permutation invariance while making regional aggregation sensitive to local feature geometry, (ii) provide each query with both local evidence and a compact global read, and (iii) use decoder-depth history without imposing an arbitrary order on point tokens. COC is therefore an SSM-inspired geometry-state extension of a query-based segmenter, rather than a direct sequential scan of the point cloud.

The contributions of this work are summarized as follows:

1. **Geometry-aware discretized integration.** LGDI replaces fixed regional statistics with a positive scalar measure, normalized local coordinates, a weak feature Jacobian, and paired cross-product descriptors obtained by complete spatial contraction. The spatial quantities are contracted so that the regional descriptor is invariant to joint point permutation, translation, and proper rotation of the coordinate frame.
2. **Local/global query-state reading.** SAG-CA adds a small permutation-invariant state bank to the query decoder. A query-conditioned positive read and a gated fusion

combine global context with the local kernel integral, while masked rows retain a mask-preserving softmax path for the original decoder semantics.

3. **Stable depth-wise evolution.** DSSA scans decoder depth in both directions, conditions the transition on scene scale, radial spread, and point count, and reads the resulting trajectory as a residual refinement. It is deliberately an SSM-inspired stable transition and does not require an arbitrary ordering of raw points.
4. **Unified evaluation.** We integrate these components into OneFormer3D and evaluate semantic, instance, and panoptic segmentation on ScanNetV2 and S3DIS, with ablations that separate measure integration, state reading, and depth evolution.

2 Related Work

2.1 3D Point Cloud Segmentation and Feature Aggregation

Point cloud segmentation methods differ in how they preserve irregular geometry while constructing a representation that can be processed efficiently. Point-based models operate directly on unordered samples: PointNet [7] establishes permutation-invariant point processing, DGCNN [8] introduces dynamic neighborhood graphs, and KPConv [9], PointNeXt [10], PointMetaBase [11], PointGT [12], and PointASNL [43] improve local geometric modeling through deformable kernels, stronger feature transformations, or adaptive sampling. Point Transformer [25], Point Transformer V2 [26], Fast Point Transformer [27], and Stratified Transformer [28] add long-range contextual interaction with attention.

Voxel-based methods instead regularize the input and exploit sparse computation. VoxNet [13] is an early volumetric approach, while MinkowskiNet [14], TorchSparse++ [15], and ASH [16] improve sparse processing. LargeKernel3D [17] and VoxelNeXt [18] enlarge receptive fields while retaining sparse representations. Projection-based approaches use range or bird's-eye-view images for efficient 2D processing [19, 20], which is attractive in large-scale perception but can lose 3D structure under occlusion or clutter [21, 22]. These families motivate the same unresolved design question for our setting: how should a region be summarized when its samples are unordered and unevenly distributed? COC addresses this question at the superpoint stage with a positive measure, a learned content integral, and contracted Jacobian and paired cross-product descriptors rather than a single fixed statistic.

2.2 Query-based Unified 3D Segmentation

DETR [37] established object queries as carriers for end-to-end set prediction, and Mask2Former [38, 39] extended query decoding with masked attention for universal segmentation. In 3D, Mask3D [40] applies a mask-transformer formulation to instance segmentation, SPFormer [41] uses superpoints for efficient scene reasoning, and OneFormer3D [42] unifies semantic, instance, and panoptic segmentation in one decoder. Top-down and instance-focused methods, including TD3D [46], SoftGroup [47], PBNet [48], ISBNet [49], PointGroup [50], DyCo3D [51], SSTNet [52], HAIS [53], and SegGCN [71], further show the importance of grouping, geometric aggregation, and instance-aware refinement.

Our work retains the unified query interface of OneFormer3D but changes the information flow inside it. Existing query decoders mainly obtain scene context through query–point or query–superpoint attention and commonly use the final decoder state for prediction. COC adds an explicit permutation-invariant state bank that each query can read through a positive kernel and a learned gate, while preserving the mask support and masked-softmax behavior required by iterative mask supervision. It also uses the intermediate decoder states through a depth-wise residual read. Thus, SAG-CA and DSSA complement the query formulation instead of replacing the set-prediction objective.

2.3 State Space Models for Vision and Point Clouds

SSMs provide a state-transition view of sequence modeling rooted in linear systems [54–56]. S4 [30, 57], S5 [31, 58], H3 [59], and Mamba [32] improve long-range modeling through structured or input-dependent transitions. This line of work has been extended to multidimensional visual signals by S4ND [60], ViM [61], and VMamba [62],

and to point clouds by Point Mamba [33], Point Cloud Mamba [34], PointMamba [35], Pamba [29], and 3D-UMamba [36]. These methods establish the relevance of state propagation for geometric perception, but their scans require an ordering or a traversal policy for the processed tokens.

COC uses the state-space idea selectively. LGDI treats the unordered regional samples as a discrete measure and does not scan them. SAG-CA uses a fixed number of state slots to obtain a permutation-invariant global summary and reads it with query-conditioned positive weights. DSSA applies a stable bidirectional transition only along decoder depth, which is an ordered axis, and conditions it on scene-level radial statistics. The implementation is therefore SSM-inspired rather than a claim of a full Mamba block with independent input and output parameter fields.

2.4 Adaptive and Hierarchical Feature Modulation

Adaptive feature modulation appears in both 2D and 3D dense prediction. Swin Transformer [63] and Pyramid Vision Transformer [64] build hierarchical representations, while UP-DETR [65] and Mask2Former [39] demonstrate that structured query refinement can improve downstream prediction. In point clouds, PAConv [44] dynamically assembles kernels, PointVector [45] uses vector-valued geometric features, DyCo3D [51] employs dynamic convolution, and ISBNet [49] combines instance-aware sampling with box-aware convolution.

COC is related to these adaptive designs but operates at three different information axes: the regional measure adapts sample contribution, the state bank adapts global context to each query, and the depth transition adapts the use of intermediate decoder states. This distinction is useful for unified segmentation because the method improves aggregation and refinement without introducing a separate instance-specific branch or imposing a traversal order on the point cloud.

2.5 Relation to Common Aggregation and Recurrence Baselines

The difference between COC and common aggregation baselines is functional rather than a change of terminology. Standard attention pooling assigns a dense query–sample compatibility score and uses a softmax-weighted value sum [24]; its primary object is pairwise sample selection. LGDI instead learns one positive scalar measure per regional sample and uses the same measure for a content integral, a full-channel contracted first-order Jacobian descriptor, and a paired second-order cross-product descriptor. Consequently, two regions with the same mean can still differ through their learned measure and differential descriptors. SE or generic gated pooling usually computes a pooled channel statistic and applies a channel-wise rescaling; it does not expose a sample measure, a weak spatial Jacobian, or an antisymmetric contracted descriptor. SAG-CA uses a feature-wise gate only after constructing separate local and global reads, so the gate selects between two already aggregated information sources rather than replacing sample-level integration.

GRU/BiGRU aggregation treats its inputs as an ordered sequence and updates a recurrent state along that order. This is appropriate for temporal data, but raw point or superpoint tokens have no canonical sequence. COC leaves regional tokens unordered, uses a permutation-invariant state bank for global context, and applies recurrence only to the intrinsically ordered decoder-depth axis in DSSA. A conventional MLP depth aggregator maps a concatenation or a fixed weighted sum of layer states to an output; it has no recurrent state transition and does not use a bidirectional terminal read. DSSA instead propagates bounded states in both depth directions and scores the complete trajectory against the terminal query. These distinctions explain why COC is not equivalent to attention pooling, channel gating, recurrent point scanning, or a static MLP over decoder layers, even though all of them can be implemented with standard neural primitives.

3 Method

COC is implemented inside the unified query decoder of OneFormer3D [42]. The sparse backbone produces point or voxel features, and the existing superpoint or macro-voxel assignments define unordered regions. COC changes three information-transfer operations while retaining the original prediction heads: LGDI integrates each region with a learned positive measure, SAG-CA reads regional tokens together with a compact global state bank, and DSSA evolves the ordered sequence of decoder-layer states. No traversal order is imposed on point, voxel, or region tokens.

Figure 1 summarizes the active path. The backbone features are first aggregated by LGDI; superpoints are used for ScanNet and label-free macro-voxels for S3DIS. The resulting regional states enter every decoder cross layer. Each such layer performs the local/global SAG-CA read and then applies the unchanged self-attention and feed-forward blocks. The collected query states are refined by DSSA before the existing classification, objectness, mask, and semantic heads.

3.1 Measure-State View of Point Cloud Decoding

Let region $\mathcal{R}_k$ contain N_k features $f_i \in \mathbb{R}^D$ and coordinates $p_i \in \mathbb{R}^3$. The implementation first computes a centroid, a local radius, and a normalized relative coordinate:

$$\begin{aligned} c_k &= \frac{1}{N_k} \sum_{i \in \mathcal{R}_k} p_i, \\ s_k &= \left(\frac{1}{N_k} \sum_{i \in \mathcal{R}_k} \frac{\|p_i - c_k\|_2^2}{3} \right)^{1/2}, \\ \widetilde{p}_i &= \frac{p_i - c_k}{\max(s_k, \varepsilon)}. \end{aligned} \tag{1}$$

The radial descriptor is

$$g_i = \left[\|\widetilde{p}_i\|_2,\ \|\widetilde{p}_i\|_2^2,\ \log(1 + \|\widetilde{p}_i\|_2),\ 1 \right] \in \mathbb{R}^4. \tag{2}$$

It uses no fixed coordinate axis. Therefore the radial descriptor is unchanged by joint permutations, translations, and proper rotations of the coordinates. Uniform local scale changes also cancel when $s_k > \varepsilon$; the clamp makes this statement non-strict for degenerate regions. The learned density and the final state additionally depend on f_i and learned maps, so these are geometric-descriptor properties rather than unconditional invariance claims for the complete feature transformation.

The measure network outputs one positive scalar per sample, shared by all feature channels. We use the lower-clamp operator $\mathrm{clamp}_\varepsilon(t) = \max(t, \varepsilon)$ only as a numerical guard:

$$\begin{aligned} d_i &= \mathrm{Softplus}(m([f_i; g_i])), \\ D_k &= \mathrm{clamp}_\varepsilon \left(\sum_{j \in \mathcal{R}_k} d_j \right), \\ \pi_i &= \frac{d_i}{D_k}, \qquad d_i \in \mathbb{R}_{>0}. \end{aligned} \tag{3}$$

Let $m_k = \sum_{j \in \mathcal{R}_k} d_j$. The clamp gives

$$\sum_{i \in \mathcal{R}_k} \pi_i = \frac{m_k}{\max(m_k, \varepsilon)}.$$

Thus the weights sum to one when $m_k \geq \varepsilon$ and are strictly sub-normalized only in the low-mass numerical-guard regime $0 < m_k < \varepsilon$. This is a numerical safeguard rather than a modeling claim of exact normalization in that exceptional case. The normalized weights determine relative sample contribution. The separate mass code used by LGDI is

$$\ell_k = \log \left(1 + \frac{\sum_{i \in \mathcal{R}_k} d_i}{N_k} \right). \tag{4}$$

3.2 Lean Gated Discretized Integration

Let $\phi(f_i) \in \mathbb{R}^D$ be the learned content map. The mean and weighted content integral are

$$\begin{aligned} \mu_k &= \frac{1}{N_k} \sum_{i \in \mathcal{R}_k} f_i, \\ \overline{\phi}_k &= \sum_{i \in \mathcal{R}_k} \pi_i \phi(f_i). \end{aligned} \tag{5}$$

The weak feature differential is a $D \times 3$ matrix,

$$J_k = \sum_{i \in \mathcal{R}_k} \pi_i \big(\phi(f_i) - \overline{\phi}_k \big) \widetilde{p}_i^{\top} \in \mathbb{R}^{D \times 3}. \tag{6}$$

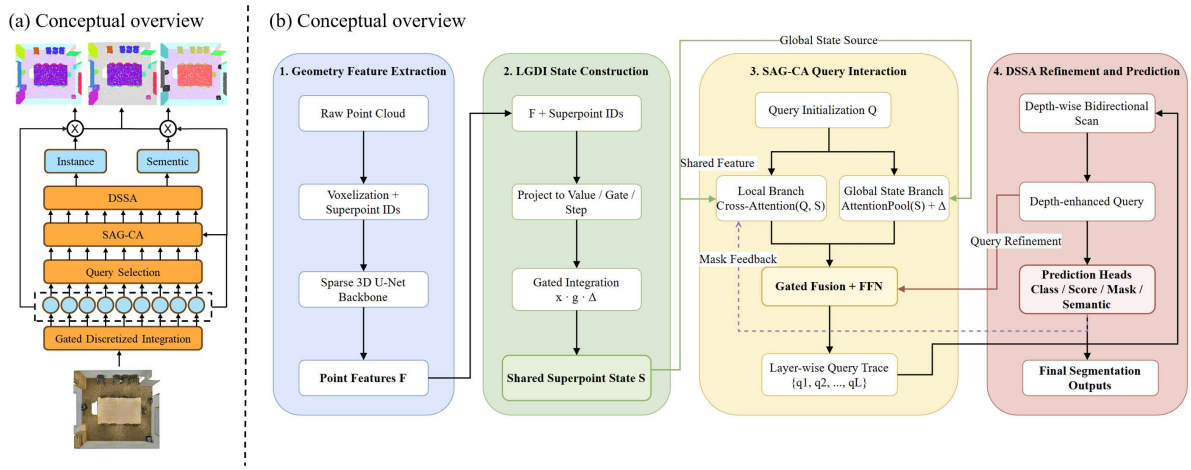


Fig. 1 Overview of COC. A sparse 3D backbone produces point or voxel features. LGDI forms regional states with a positive scalar measure and contracted first-order and paired cross-product descriptors. SAG-CA combines a local kernel read with a query-conditioned read from a permutation-invariant state bank. DSSA refines the ordered decoder-depth trajectory before the unchanged OneFormer3D prediction heads.

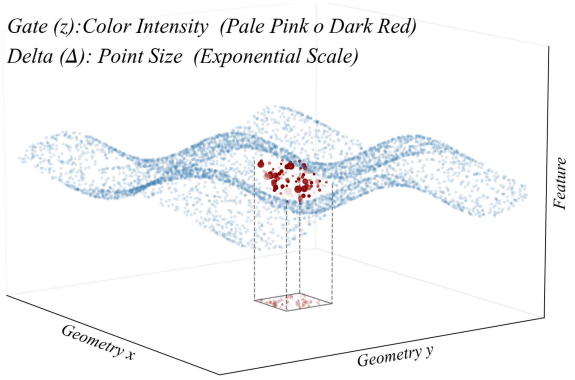


Fig. 2 LGDI computes a scalar positive density from point features and radial local descriptors. It combines a weighted content integral with a contracted first-order Jacobian descriptor and paired second-order cross-product descriptors, followed by a residual regional update.

The code takes the Euclidean norm of each feature row, so the contracted first-order descriptor is a vector, not a scalar:

$$\mathbf{e}_k^{(1)} = [\|J_{k,1:}\|_2, \dots, \|J_{k,D:}\|_2] \in \mathbb{R}^D. \tag{7}$$

For the paired second-order descriptor, the first $2r$ rows are grouped into pairs. Each row of J_k is a three-dimensional coordinate-space vector; the cross product is taken between two such rows and its norm is retained:

$$\begin{aligned} \omega_{k,a} &= J_{k,2a-1,:} \times J_{k,2a,:}, \\ e_{k,a}^{(2)} &= \|\omega_{k,a}\|_2, \qquad a = 1, \dots, r, \\ \mathbf{e}_k^{(2)} &= [e_{k,1}^{(2)}, \dots, e_{k,r}^{(2)}] \in \mathbb{R}^r. \end{aligned} \tag{8}$$

If a narrow ablation has fewer than $2r$ rows, zero rows are appended before pairing, as in the implementation. For $a, b \in \mathbb{R}^3$, $\|a \times b\|_2 = \|a \wedge b\|_2$ is the area norm of the two-vector wedge after the Hodge identification in $\mathbb{R}^3$. Hence $e_{k,a}^{(2)}$ is a contracted cross-product descriptor, rather than a claim that feature channels themselves have a canonical geometric order. The row norms and cross-product norms are invariant to proper rotations after complete spatial contraction; all sums are invariant to a joint permutation of samples.

The terms "calculus" and "exterior-form descriptor" describe the organization of these finite-dimensional statistics. In particular, J_k is a weighted weak feature–coordinate cross-moment, $\|J_{k,d:}\|_2$ is a contracted first-order descriptor, and $\|J_{k,2a-1,:} \times J_{k,2a,:}\|_2$ is the norm of a paired two-vector wedge after contraction. They do not assert a new continuous calculus, a differential-geometric theorem, or an exact discretization limit. The properties used by COC follow from finite sums, complete contraction over the spatial coordinates, positive weights, and the numerical clamp; the

learned maps remain ordinary trainable neural functions.

The second-order term is deliberately a low-cost descriptor. With the active setting $D = 256$ and $r = 8$, it pairs the first $2r = 16$ rows of J_k and leaves the remaining 240 rows to the first-order term. Feature channels have no intrinsic physical ordering, so adjacent pairing is a fixed engineering choice in the learned feature chart, not a claim that these channels form canonical geometric directions or that this pairing is uniquely optimal. The complete first-order vector still uses all D rows. The second-order descriptor introduces no independent trainable matrix: relative to a first-order-only input, it adds r scalars and increases the first layer of F_{int} by rD weights, namely $8 \times 256 = 2048$ weights for the active configuration. Its extra arithmetic is $O(r)$ three-dimensional cross products and norms; the $D \times 3$ Jacobian is already required by the first-order term. Sensitivity to other values of r, random pairings, and learned pairings is outside the reported ablation protocol, so the paper treats the current pairing as a reproducible implementation choice rather than as a theoretical optimum.

The concatenated input has dimension $3D + r + 1$ and the output is a residual update:

$$\begin{aligned} h_k &= [\mu_k; \overline{\phi}_k; \mathbf{e}_k^{(1)}; \mathbf{e}_k^{(2)}; \ell_k] \in \mathbb{R}^{3D+r+1}, \\ S_k &= \mu_k + F_{\text{int}}(h_k), \qquad S_k \in \mathbb{R}^D. \end{aligned} \tag{9}$$

Here $F_{\text{int}} : \mathbb{R}^{3D+r+1} \to \mathbb{R}^D$ is the implemented two-layer residual map: a bias-free linear layer from $3D + r + 1$ to D, followed by LayerNorm and GELU, and a bias-free linear layer from D to D. Thus F_{int} changes the descriptor dimension but introduces no additional spatial tensor. The descriptor is a finite-dimensional feature statistic; the word "second-order" refers to the bilinear cross product of two rows of the weak Jacobian, not to a second spatial derivative. For S3DIS, the same label-free macro-voxel mapping is used to aggregate features and to remap targets.

3.3 State-Aware Gated Cross-Attention

At decoder layer l, let $Q \in \mathbb{R}^{N_q \times D}$ be the queries and $S \in \mathbb{R}^{M \times D}$ the regional states. Let G_q and G_s denote four-dimensional radial descriptors; G_q is

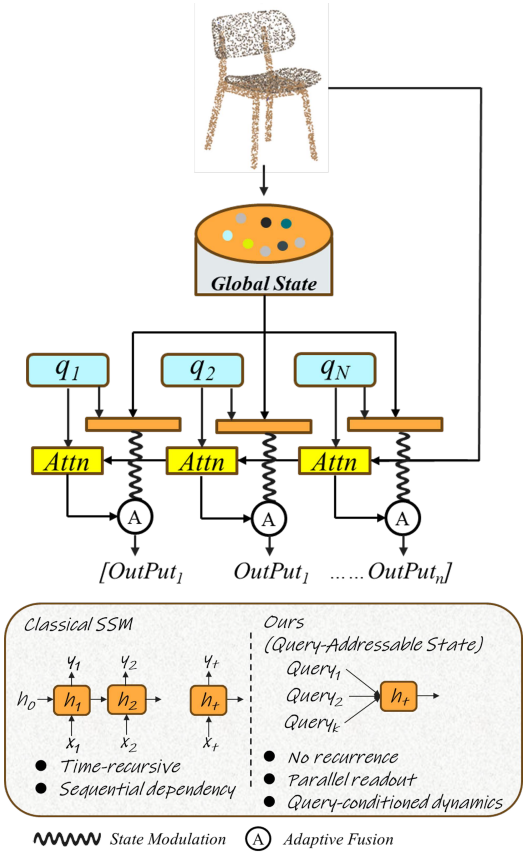


Fig. 3 SAG-CA combines a local positive kernel read (or a mask-preserving softmax path) with a permutation-invariant multi-slot global state bank and query-conditioned gated fusion.

zero when query coordinates are unavailable. With no attention mask, the positive low-rank maps are

$$\begin{aligned} \Phi_q &= \text{Softplus}(QW_q + G_q W_{qg}) + \varepsilon, \\ \Phi_s &= \text{Softplus}(SW_s + G_s W_{sg}) + \varepsilon. \end{aligned} \tag{10}$$

Here $W_q, W_s \in \mathbb{R}^{D \times R_k}$ and $W_{qg}, W_{sg} \in \mathbb{R}^{4 \times R_k}$; the bias-free maps are applied row-wise, giving $\Phi_q \in \mathbb{R}^{N_q \times R_k}$ and $\Phi_s \in \mathbb{R}^{M \times R_k}$. Hence $K_V \in \mathbb{R}^{R_k \times D}$, $d_q \in \mathbb{R}^{N_q \times 1}$, and the row-wise division in Z_{pos} broadcasts over the D channels. Because every feature-map entry is at least ε, the denominator is above the clamp for ordinary nonempty inputs; the clamp remains a numerical guard. The key–value summary and normalized positive read

are computed as

$$
\begin{aligned}
K_V &= \Phi_s^\top S \in \mathbb{R}^{R_k \times D}, \\
d_q &= \mathrm{clamp}_\varepsilon \left[\Phi_q \left(\sum_{j=1}^{M} \Phi_{s,j:} \right)^\top \right], \\
Z_{\mathrm{pos}} &= \frac{\Phi_q K_V}{d_q} \in \mathbb{R}^{N_q \times D}.
\end{aligned}
\tag{11}
$$

The division is row-wise. Thus the unmasked path is linear in the number of regional tokens for fixed R_k and D.

The same path adds a bounded antisymmetric feature-chart correction. Define

$$
\begin{aligned}
\mathcal{C}(x) &= \tanh(xW_{\mathcal{C}}) \in \mathbb{R}^{n \times 2r}, \\
\mathcal{C}(x) &= [\mathcal{C}^{(0)}(x), \mathcal{C}^{(1)}(x)].
\end{aligned}
\tag{12}
$$

For an input with n rows, $W_{\mathcal{C}} \in \mathbb{R}^{D \times 2r}$ and $\mathcal{C}^{(0)}(x), \mathcal{C}^{(1)}(x) \in \mathbb{R}^{n \times r}$. For two chart rows $a, b \in \mathbb{R}^{r \times 2}$, define the contracted antisymmetric inner product

$$
\langle a, b \rangle_\wedge = \sum_{t=1}^{r} \left(a_{t,0} b_{t,1} - a_{t,1} b_{t,0} \right) \in \mathbb{R}. \tag{13}
$$

After reshaping the last dimension into r pairs, the correction uses two key–value summaries:

$$
\begin{aligned}
F_1 &= \mathcal{C}^{(1)}(S)^\top S \in \mathbb{R}^{r \times D}, \\
F_0 &= \mathcal{C}^{(0)}(S)^\top S \in \mathbb{R}^{r \times D}, \\
Z_{\mathrm{corr}} &= \tanh(\eta) \frac{\mathcal{C}^{(0)}(Q) F_1 - \mathcal{C}^{(1)}(Q) F_0}{d_q} \in \mathbb{R}^{N_q \times D}, \\
Z_{\mathrm{loc}} &= Z_{\mathrm{pos}} + Z_{\mathrm{corr}}.
\end{aligned}
\tag{14}
$$

Here the products sum over the r chart pairs and all divisions use the same row denominator d_q. The scalar η is the learned 'form_scale' parameter. The antisymmetric chart term is a bounded feature compatibility correction; it is not an additional spatial derivative.

When an iterative attention mask is supplied, the implementation switches to a mask-preserving softmax path. The word "mask-preserving" refers to the support constraint and masked-softmax operation; it does not mean numerical equivalence to the original OneFormer3D attention. For comparison, let $q, s \in \mathbb{R}^{* \times D}$. The original masked cross-attention uses $W_Q, W_K, W_V \in \mathbb{R}^{D \times D}$ and produces tensors with the same feature width D:

$$
\begin{aligned}
Q^0 &= qW_Q, & K^0 &= sW_K, \\
V^0 &= sW_V, & & \\
A^0_{qj} &= \mathrm{Softmax}_j(\ell^0_{qj}), & & \\
\ell^0_{qj} &= \frac{(Q^0_q)^\top K^0_j}{\sqrt{D}} + \widehat{M}_{qj}, & & \\
Z^0_q &= \sum_{j=1}^{M} A^0_{qj} V^0_j, & &
\end{aligned}
\tag{15}
$$

where $\widehat{M}_{qj} = 0$ for an allowed key and $-\infty$ for a disallowed key. COC keeps this support rule but changes the score and value path as follows. Let $M_{qj} = 1$ denote a disallowed key. The unnormalized score is

$$
\begin{aligned}
L_{qj} &= \frac{q_q^\top s_j}{\sqrt{D}} + \frac{\langle \mathcal{C}(q_q), \mathcal{C}(s_j) \rangle_\wedge}{\sqrt{r}} \\
&\quad + b_{qj}, \\
b_{qj} &= -\frac{\|u_q - v_j\|_2^2}{2\,[\mathrm{Softplus}(\rho) + \varepsilon]^2}, \\
\widetilde{L}_{qj} &= \begin{cases} L_{qj}, & M_{qj} = 0, \\ -\infty, & M_{qj} = 1, \end{cases} \\
w_{qj} &= \mathrm{Softmax}_j(\widetilde{L}_{qj}), \\
Z_{\mathrm{loc}} &= \sum_{j=1}^{M} w_{qj} s_j.
\end{aligned}
\tag{16}
$$

Here $\langle \mathcal{C}(q), \mathcal{C}(s) \rangle_\wedge$ denotes the sum of the two-dimensional antisymmetric products over the r chart pairs; u_q and v_j are the query and source coordinates, and ρ is the learned coordinate-bandwidth parameter. Unlike (15), the active masked path uses the incoming query and source tensors directly, does not apply independent W_Q, W_K, and W_V projections, adds the chart correction, and optionally adds the coordinate-distance bias. Its value is the source tensor s_j in the decoder call. The distance bias b_{qj} is included only when both coordinate tensors are supplied. Thus, "mask-preserving" is the precise comparison to the OneFormer3D baseline: the allowed support, $-\infty$ masking, and softmax normalization are retained, whereas the numerical compatibility function is COC-specific. For a fully masked query

row, the implementation returns zero weights rather than propagating a NaN.

Both local paths then use the same query-conditioned stable interpolation:

$$\begin{aligned} \delta &= \mathrm{Softplus}(QW_\tau + b_\tau), \\ \lambda &= \mathrm{Softplus}(\lambda_0), \\ \tau &= \exp[-\,\mathrm{clamp}\,(\delta \odot \lambda, 0, 20)]\,, \\ Z_{\mathrm{int}} &= \tau \odot Q + (1-\tau) \odot Z_{\mathrm{loc}}. \end{aligned} \tag{17}$$

Here $W_\tau \in \mathbb{R}^{D\times D}$, $b_\tau, \lambda_0 \in \mathbb{R}^D$, and $\delta, \tau, Z_{\mathrm{int}} \in \mathbb{R}^{N_q\times D}$. The element-wise interpolation is therefore dimension preserving.

The global bank has R_s slots, with $R_s \ll M$. The assignment and value tensors are

$$\begin{aligned} &L_a = W_a[S; G_s] + b_a \in \mathbb{R}^{M\times R_s}, \\ &A = \mathrm{Softmax}_M(L_a), \qquad V_s = W_v S \in \mathbb{R}^{M\times D}, \\ &m = A^\top \mathbf{1}_M \in \mathbb{R}^{R_s}. \end{aligned} \tag{18}$$

The softmax is along the M region rows for every slot, so in exact arithmetic $m = \mathbf{1}_{R_s}$. The implementation still computes the guarded mass vector m and applies the following diagonal normalization; it is mathematically redundant for a finite, nonempty region set and only protects against numerical underflow or masked empty inputs. The state bank is therefore

$$\begin{aligned} &H = \mathrm{Diag}(\mathrm{clamp}_\varepsilon(m))^{-1}\, A^\top V_s, \\ &H = A^\top V_s \quad \text{for finite nonempty inputs.} \end{aligned} \tag{19}$$

The assignment matrix uses $W_a \in \mathbb{R}^{(D+4)\times R_s}$ and $b_a \in \mathbb{R}^{R_s}$; the value map is $W_v \in \mathbb{R}^{D\times D}$, and $H \in \mathbb{R}^{R_s\times D}$. Each slot is consequently a normalized weighted mean of transformed regional states, making H invariant to a joint permutation of regions.

For a query, positive maps read the bank as

$$\begin{aligned} U_q &= \mathrm{Softplus}(QW_{rq}) + \varepsilon, \\ K_H &= \mathrm{Softplus}(HW_{rk}) + \varepsilon, \\ D_q^H &= \mathrm{clamp}_\varepsilon\big[(U_q K_H^\top)\mathbf{1}_{R_s}\big]\,, \\ P &= \frac{U_q K_H^\top}{D_q^H}, \\ H_q &= PH \in \mathbb{R}^{N_q\times D}. \end{aligned} \tag{20}$$

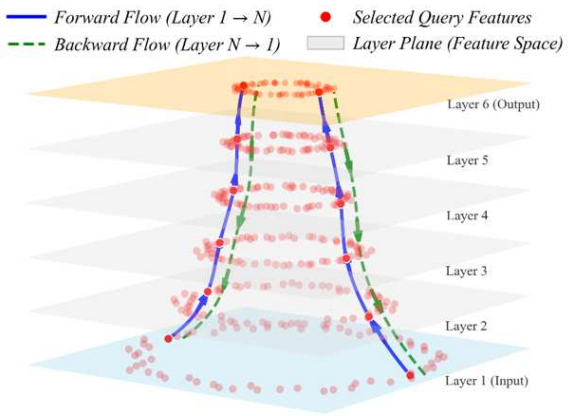


Fig. 4 DSSA treats decoder layers as samples on an ordered depth axis. It performs forward and backward stable state evolution, fuses the two directions, and reads the complete query trajectory before the final residual refinement.

In this read, $W_{rq}, W_{rk} \in \mathbb{R}^{D\times R_k}$, $D_q^H \in \mathbb{R}^{N_q\times 1}$, and $P \in \mathbb{R}^{N_q\times R_s}$. The gate uses $W_g \in \mathbb{R}^{2D\times D}$ and $b_g \in \mathbb{R}^D$, so $\beta, Y \in \mathbb{R}^{N_q\times D}$. The local and global reads are fused by the implemented feature-wise gate:

$$\begin{aligned} \beta &= \sigma(W_g[Q; H_q] + b_g)\,, \\ Y &= \beta \odot Z_{\mathrm{int}} + (1-\beta) \odot H_q. \end{aligned} \tag{21}$$

The cross layer applies the warm-start residual, dropout, and normalization:

$$Q^{l+1/2} = \mathrm{LN}\big(Q + \mathrm{Dropout}\,\big(a(Y-Q)\big)\big)\,. \tag{22}$$

The bank is rebuilt from the current regional states at each cross layer. Decoder cross layers and DSSA use the shared decoder operator instance; the LGDI operator is separate when its input width differs.

3.4 Depth-wise Selective Scan Aggregation

Let $X_l \in \mathbb{R}^{B\times N_q\times D}$ be the query tensor at decoder depth l, $l = 1, \ldots, L$. Unlike point-token order, this depth order is intrinsic to the decoder. For each scene, the code forms a three-scalar descriptor and adds its projection to every layer:

$$\begin{aligned} v_b &= [\mathrm{rms}(\|p-c\|_2), \mathrm{std}(\|p-c\|_2), \log(1+N_b)]\,, \\ \widetilde{X}_l &= X_l + W_{\mathrm{scene}} v_b. \end{aligned} \tag{23}$$

For a batch of B scenes, $v_b \in \mathbb{R}^3$ and $W_{\text{scene}} \in \mathbb{R}^{D\times 3}$, so $\widetilde{X}_l \in \mathbb{R}^{B\times N_q\times D}$. The positive transition step, decay, and candidate state are

$$\begin{aligned}\Delta_l &= \mathrm{Softplus}(W_\Delta \widetilde{X}_l + b_\Delta),\\ \lambda &= \mathrm{Softplus}(\lambda_0),\\ \alpha_l &= \exp[-\,\mathrm{clamp}(\Delta_l \odot \lambda, 0, 20)]\,,\\ U_l &= W_s \widetilde{X}_l.\end{aligned} \tag{24}$$

The active dimensions are $W_\Delta \in \mathbb{R}^{D\times D}$, $b_\Delta \in \mathbb{R}^D$, $W_s \in \mathbb{R}^{D\times D}$, and $\Delta_l, \alpha_l, U_l \in \mathbb{R}^{B\times N_q\times D}$. The learned decay vector $\lambda_0 \in \mathbb{R}^D$ is broadcast over batch, query, and depth indices. The two directional recurrences are

$$\begin{aligned}H_l^{\rightarrow} &= \alpha_l \odot H_{l-1}^{\rightarrow} + (1-\alpha_l)\odot U_l, \qquad H_0^{\rightarrow} = 0,\\ H_l^{\leftarrow} &= \alpha_l \odot H_{l+1}^{\leftarrow} + (1-\alpha_l)\odot U_l, \quad H_{L+1}^{\leftarrow} = 0.\end{aligned} \tag{25}$$

The directional states are fused with a feature-wise gate:

$$\begin{aligned}R_l &= \sigma(W_r[H_l^{\rightarrow}; H_l^{\leftarrow}; X_l] + b_r)\,,\\ T_l &= R_l \odot H_l^{\rightarrow} + (1-R_l)\odot H_l^{\leftarrow}.\end{aligned} \tag{26}$$

Here $W_r \in \mathbb{R}^{3D\times D}$ and $b_r \in \mathbb{R}^D$, giving $R_l, T_l \in \mathbb{R}^{B\times N_q\times D}$. The terminal scores are scalar for each depth and query, w_l is normalized over l, and $T \in \mathbb{R}^{B\times N_q\times D}$. For the terminal read, the final layer X_L scores every trajectory state. With the same chart $\mathcal{C}$ as above,

$$\begin{aligned}a_l &= \frac{\langle T_l, X_L\rangle}{\sqrt{D}} + \langle \mathcal{C}(T_l), \mathcal{C}(X_L)\rangle_\wedge\,,\\ w_l &= \mathrm{Softmax}_l(a_l), \qquad T = \sum_{l=1}^{L} w_l T_l,\\ Y &= X_L + F_{\text{depth}}(T).\end{aligned} \tag{27}$$

The depth map $F_{\text{depth}} : \mathbb{R}^D \to \mathbb{R}^D$ is implemented by a bias-free $D \to D$ linear layer, LayerNorm, GELU, and a second bias-free $D \to D$ linear layer, applied independently to each batch/query row. The last projection in F_{depth} is initialized with a small standard deviation. The DSSA wrapper then blends Y with X_L using its auxiliary coefficient, and the decoder applies its configured aggregation weight. This is an SSM-inspired stable transition along decoder depth, rather than a full Mamba block: the implementation has no independent input/output B and C parameter fields and never scans raw point tokens.

The stability statement is limited to the transition coefficients. Since $\Delta_l = \mathrm{Softplus}(\cdot) \geq 0$ and $\lambda = \mathrm{Softplus}(\lambda_0) > 0$, the implementation gives $e^{-20} \leq \alpha_l \leq 1$. Each directional update is therefore a bounded interpolation between the previous state and the current candidate, and the terminal weights satisfy $w_l \geq 0$ and $\sum_l w_l = 1$. This does not imply a global Lipschitz bound for the trained network or invariance of the final segmentation output; those properties also depend on the learned projections, voxelization, masks, and prediction heads.

3.5 Prediction Heads

The refined query representation is passed to the original unified prediction heads. The classification head predicts instance or semantic classes, the optional score head predicts objectness, and the mask head forms query–feature inner products. ScanNet decoders additionally use the semantic head defined by the OneFormer3D configuration. Auxiliary predictions remain available for iterative training. No task-specific decoder or post-processing module is introduced by COC; the contribution is confined to regional integration, query-state reading, and depth-wise refinement.

Overall distinction from existing methods. COC follows one measure–kernel–state principle at three axes with different order structures. LGDI constructs a geometry-aware regional state from an unordered sample measure; SAG-CA makes that state query-addressable through a local read, a compact global bank, and a feature-wise gate; DSSA reuses the sequence of decoder states only along the naturally ordered depth axis. The modules are complementary rather than mathematically indispensable: removing one path leaves a valid OneFormer3D-style model, but removes the representation supplied at its corresponding axis. The cumulative ablation therefore measures conditional effects in a nested design. The shared optimization principle is to learn positive contributions and bounded state updates while preserving the relevant order contract: permutation-invariant sums for regional and state-bank construction,

masked support for iterative attention, and stable recurrence for decoder depth. Together these operations provide different sufficient statistics to the unchanged unified prediction heads without imposing an arbitrary order on the point cloud.

4 Experiments

4.1 Experimental Setup

All experiments were conducted on a single NVIDIA RTX 4090 GPU. The software environment was Ubuntu 22.04.5 LTS, PyTorch 1.13, and CUDA 11.6. We evaluate the proposed method on ScanNetV2 and S3DIS [66, 67], covering semantic, instance, and panoptic segmentation. Entries annotated with ± are means and standard deviations over three independent runs; entries without ± are single reported results unless explicitly described as an aggregate protocol.

ScanNetV2. ScanNetV2 contains 1,613 reconstructed indoor scenes, including 1,201 training scenes, 312 validation scenes, and 100 hidden test scenes. Following the common evaluation protocol, instance segmentation is evaluated on 18 object categories, while semantic and panoptic segmentation additionally include "wall" and "floor" as stuff classes.

S3DIS. S3DIS contains 272 scenes from six large-scale indoor areas and 13 semantic categories. For Area-5, the implementation trains on Areas 1, 2, 3, 4, and 6 and evaluates on Area 5. We also report the standard 6-fold protocol, which aggregates six held-out-area runs; the released configuration directly specifies the Area-5 split. Following the evaluator configuration, table, chair, sofa, bookcase, and board are thing classes, while the remaining categories are stuff classes.

Evaluation Metrics. For semantic segmentation, we report mean Intersection over Union (mIoU). For instance segmentation, we report mean Average Precision (mAP), mAP_{50}, and mAP_{25} when applicable. For S3DIS, we additionally report mean Precision and mean Recall at IoU threshold 50%, denoted as mPrec_{50} and mRec_{50}. For panoptic segmentation, we report Panoptic Quality (PQ), together with PQ^{th} for thing classes and PQ^{st} for stuff classes.

Implementation Details. COC is implemented in MMDetection3D following the OneFormer3D pipeline [42]. The active configurations use AdamW and a polynomial learning-rate schedule with power 0.9 for 512 epochs. The dataset-specific optimizer settings are 1×10^{-4} learning rate and 0.1 weight decay for ScanNetV2, and 1.2×10^{-4} learning rate and 0.05 weight decay for S3DIS. ScanNetV2 uses a batch size of 4; S3DIS uses a batch size of 2 with two-step gradient accumulation, giving an effective batch size of 4. ScanNetV2 applies random 3-D flips, full-range rotation about the vertical axis, scaling, translation, point jitter, color jitter, and elastic distortion. S3DIS uses flips, scaling, translation, point jitter, color jitter, and elastic distortion; its active configuration sets the rotation range to zero. ScanNetV2 uses the provided graph-based superpoints with a 0.02-m voxel size, whereas S3DIS uses 0.05-m voxelization followed by label-free macro-voxel grouping with group size 2. The loss weights are fixed within each dataset protocol and are not tuned separately for an individual metric. Semantic, instance, and panoptic supervision are therefore evaluated under the same fixed configuration for a given dataset.

Network and optimization settings.

Both dataset configurations use feature width $D = 256$, eight attention heads, a feed-forward hidden width of 1024, zero dropout, kernel rank $R_k = 32$, state-bank slots $R_s = 4$, form rank $r = \max(2, D/32) = 8$, and $\varepsilon = 10^{-6}$. The LGDI measure network has hidden width 128. The parameter denoted by λ_0 in (24) is stored as the decoder decay vector and is initialized to zero; the step projection is initialized with zero weights and bias -4, while the final integral and depth projections use standard deviation 10^{-3}. Both backbones use "min_spatial_shape=128". ScanNetV2 uses six decoder layers and point-derived variable-length queries (the two learned query counts are zero in the configuration). S3DIS uses three decoder layers with 400 instance queries and 13 semantic queries. The ScanNet semantic loss weight is 0.2 and its instance criterion weights are $[0.5, 1.0, 1.0, 0.5]$ with non-object weight 0.1; the corresponding S3DIS values are 5.0, $[0.5, 1.0, 1.0, 2.0]$, and 0.05. S3DIS additionally samples 100,000 points per training block with block size 12, point-sampling $\varepsilon = 10^{-2}$, and label-free macro-voxel group size 2. The complete voxel/superpoint

settings are given by the active configuration files "oneformer3d_1xb4_scannet.py" and "oneformer3d_1xb2_s3dis-area-5.py".

The reported ± values are means and standard deviations from three independent runs. The configuration files do not impose a single random seed; the seed is therefore treated as a run-level factor and should be recorded when reproducing the results. ScanNetV2 results use the 512-epoch checkpoint. For S3DIS, validation is performed every eight epochs and checkpoints are selected using the prescribed AP_{50} and mIoU criteria before final evaluation. All metrics for a dataset are computed with the same fixed preprocessing, model, and loss configuration. The implementation follows the MMDetection3D v1.1.0 framework, and the complete dataset-specific configuration files and source implementation define the reported settings.

4.2 Operator-level Invariance and Stability Checks

To separate mathematical contracts from end-to-end robustness, we evaluate the active geometric operator with fixed random features and region assignments. The check uses "torch.manual_seed(0)", $D = 24$, form rank 6, 32 samples in three regions, and the same $\varepsilon = 10^{-6}$ clamp as the model. For an output z, each entry reports $\|z' - z\|_\infty$ after transforming coordinates while keeping features and region IDs fixed. The permutation test jointly permutes features, coordinates, and region IDs; the scale tests use non-degenerate coordinates; and the degenerate test sets all coordinates in a region to one point and checks a translated copy. These are operator-level checks rather than end-to-end segmentation experiments. The numerical results are listed in Table 1.

The numerical errors are at single-precision round-off level for this synthetic operator test. They verify permutation, translation, reflection, and non-degenerate scale contracts for the radial descriptors and contracted LGDI descriptors; the state-bank row reports a separate permutation check. They do not establish invariance of the complete trained segmentation network, whose features, voxelization, masks, and heads may change under an input transformation. In particular, the S3DIS training configuration uses a zero rotation range; the reported S3DIS scores therefore do not constitute an empirical rotation-robustness evaluation. A dedicated end-to-end perturbation benchmark with fixed checkpoints is required for that claim.

Table 1 Numerical checks of geometric contracts in the active operator.

Perturbation	$\|z' - z\|_\infty$
Joint sample permutation	1.19×10^{-7}
Global translation	2.98×10^{-8}
Proper rotation	0
Mirror reflection	0
Uniform scale $\times 0.5$	0
Uniform scale $\times 2.0$	0
Degenerate region + translation	0 (finite)
State-bank permutation	2.38×10^{-7}

4.3 Experimental Results

Although the numerical gains over the OneFormer3D baseline are moderate for some individual metrics, COC improves the reported semantic, instance, and panoptic scores under the corresponding evaluation protocols. The published baseline values and our local reruns are reported separately where applicable; cross-paper comparisons should therefore be read as reference points rather than strictly controlled same-run comparisons.

4.3.1 Qualitative Analysis

Figures 5 and 6 provide qualitative results on ScanNetV2 and S3DIS. The proposed model produces coherent semantic predictions for large stuff regions such as walls and floors, while also distinguishing adjacent thing instances such as chairs, tables, and boards. This is important for panoptic segmentation, where semantic consistency and instance separability need to be handled at the same time.

In object-dense indoor scenes, semantic prediction can identify object categories but cannot separate adjacent individuals by itself. The query-based instance branch, enhanced by SAG-CA, provides more discriminative object-level responses, which helps separate nearby instances with similar appearance. Meanwhile, the semantic branch provides stable stuff-region predictions, allowing the final panoptic output to combine

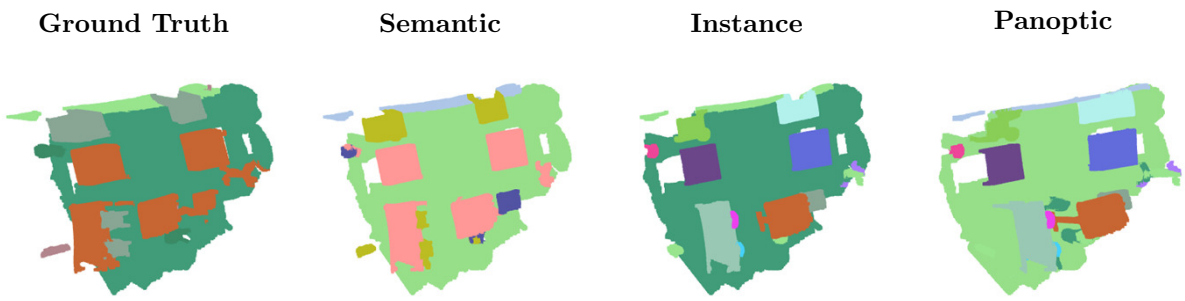


Fig. 5 Qualitative visualization on the ScanNet dataset. The four columns show the ground truth, semantic prediction, instance prediction, and panoptic prediction, respectively.

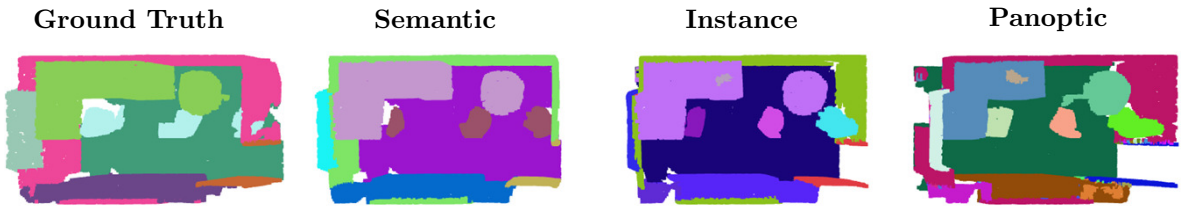


Fig. 6 Qualitative visualization on the S3DIS dataset. The results show that the proposed model can jointly preserve semantic consistency and separate adjacent object instances.

object-level separation with scene-level semantic consistency.

For large planar structures with embedded objects, such as boards on walls or furniture near floors, the visualizations show relatively clean boundaries. This observation is consistent with the design of LGDI, which assigns positive, feature-dependent weights to irregular samples before regional aggregation. The visualization is qualitative evidence of boundary behavior; it does not by itself isolate the contribution of LGDI from the backbone or prediction heads.

4.3.2 Feature Evolution Analysis

To inspect the learned representation, we visualize the evolution of the sparse-backbone feature-response intensity during training. For each point or voxel, we compute the L_2 norm of its backbone feature and render the resulting relative magnitude as a heatmap. The color bar in Fig. 7 labels this display quantity "Relative Attention Intensity"; the label denotes the relative backbone feature-response magnitude and does not imply an attention weight. This quantity is neither the LGDI density d_i nor the DSSA transition coefficient; it only shows an upstream feature signal that is subsequently processed by the learned maps.

The early training stage exhibits scattered responses, including activations on large flat regions. Later responses are more spatially structured, with stronger signals often appearing near object-rich or geometrically complex regions. This trend is consistent with the use of learned feature-dependent measures and depth transitions, but it should not be interpreted as a direct attribution of individual points to either module.

Therefore, the feature evolution visualization supports the motivation of adaptive integration: point samples should not always contribute equally to the regional state. Instead, samples located in informative or geometrically complex regions can be assigned stronger modulated contributions, while redundant samples in smooth regions can be suppressed.

4.3.3 Quantitative Analysis

Table 2 reports the ScanNetV2 comparison. COC improves the listed OneFormer3D reference scores

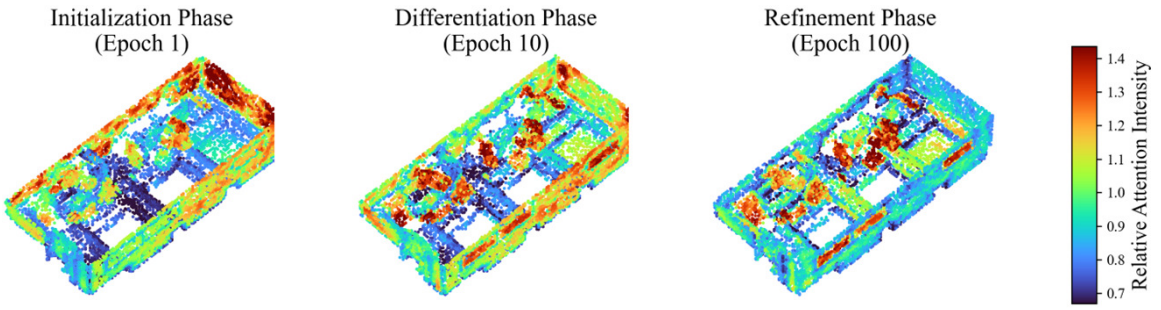


Fig. 7 Evolution of relative backbone feature-response intensity during training. The plotted intensity is the L_2 norm of the sparse-backbone feature at each point or voxel; the color-bar label "Relative Attention Intensity" denotes this display quantity. High-response regions gradually concentrate on geometrically or semantically informative structures, while large homogeneous background regions become relatively suppressed.

across the unified tasks, while the magnitude varies by dataset and metric.

On ScanNetV2, as shown in Table 2, COC improves over OneFormer3D in all reported instance and panoptic metrics. Specifically, mAP_{25} increases from 86.4 to 87.2, mAP_{50} from 78.1 to 78.8, and mAP from 59.3 to 59.8. For semantic segmentation, mIoU improves from 76.6 to 77.4. For panoptic segmentation, PQ increases from 71.2 to 72.0, with PQ^{th} improved from 69.6 to 70.4 and PQ^{st} improved from 86.1 to 87.7. Although Swin3D reports a slightly higher semantic mIoU, it does not provide the same unified instance and panoptic evaluation in this table. Therefore, COC provides a more balanced result under the unified segmentation setting.

For S3DIS Area-5, Table 3 reports the comparison with the published OneFormer3D reference. The Area-5 table uses that published value for the main comparison; the separate local baseline used for ablations is identified below.

On S3DIS Area-5, Table 3 shows improvements over the published OneFormer3D reference on every listed metric. For instance segmentation, mAP_{50} increases from 72.0 to 73.4 (+1.4) and mAP from 58.7 to 59.7 (+1.0). mPrec_{50} increases from 79.7 to 81.0 (+1.3) and mRec_{50} from 73.0 to 73.9 (+0.9). COC reaches 73.8 mIoU for semantic segmentation, which is the highest value among the semantic entries listed in this table; these cross-method values may use different training and evaluation implementations. For panoptic segmentation, PQ improves from 62.2 to 63.0 (+0.8), with PQ^{th} increasing from 58.4 to 59.1 (+0.7) and PQ^{st} from 65.5 to 66.2 (+0.7).

The standard six-fold S3DIS comparison is given in Table 4. It aggregates the held-out-area runs and is interpreted separately from the Area-5 protocol.

The 6-fold S3DIS results in Table 4 report aggregate performance across held-out areas. Compared with OneFormer3D, COC improves every listed metric: mAP_{50} from 75.8 to 76.9 (+1.1), mAP from 63.0 to 63.7 (+0.7), mPrec_{50} from 82.3 to 84.1 (+1.8), mRec_{50} from 74.1 to 75.2 (+1.1), mIoU from 75.0 to 75.8 (+0.8), PQ from 68.5 to 69.2 (+0.7), PQ^{th} from 61.5 to 61.9 (+0.4), and PQ^{st} from 74.5 to 75.6 (+1.1). Because this protocol evaluates all held-out areas rather than only Area 5, the table provides evidence that the improvement is retained across the six standard splits.

The quantitative results show that the proposed framework improves the OneFormer3D reference across the reported unified tasks. LGDI changes regional aggregation, SAG-CA combines local and global query reads, and DSSA adds a depth-wise refinement path. The gains are distributed across metrics rather than concentrated in a single task, although the tables do not establish statistical significance for every comparison.

4.3.4 Ablation Studies

We conduct ablation studies on S3DIS Area-5 to examine the contributions of the proposed modules and selected design variants. Unless otherwise

Table 2 Comparison of Segmentation Results on ScanNetV2

Method	Year	Instance			Semantic	Panoptic		
		mAP_{25}	mAP_{50}	mAP	mIoU	PQ	PQ^{th}	PQ^{st}
DKNet [23]	2022	76.9	66.7	50.8	-	-	-	-
SoftGroup [47]	2022	78.9	67.6	45.8	-	-	-	-
PBNet [48]	2023	78.9	70.5	54.3	-	-	-	-
TD3D [46]	2024	81.9	71.2	47.3	-	-	-	-
ISBNet [49]	2023	82.5	73.1	54.5	-	-	-	-
SPFormer [41]	2023	82.9	73.9	56.3	-	-	-	-
Mask3D [40]	2023	83.5	73.7	55.2	-	-	-	-
PointASNL [43]	2020	-	-	-	63.5	-	-	-
PointTransformer [25]	2021	-	-	-	70.6	-	-	-
PointNeXt-XL [10]	2022	-	-	-	71.5	-	-	-
PointMetaBase-XXL [11]	2023	-	-	-	72.8	-	-	-
Stratified Transformer [28]	2022	-	-	-	74.3	-	-	-
PointTransformerV2 [26]	2022	-	-	-	75.4	-	-	-
SceneGraphFusion [68]	2021	-	-	-	-	31.5	30.2	43.4
TUPPer-Map [69]	2021	-	-	-	-	50.2	47.8	71.5
Swin3D [70]	2025	76.4	63.2	-	**77.5**	-	-	-
OneFormer3D (base) [42]	2024	86.4	78.1	59.3	76.6	71.2	69.6	86.1
COC (Ours)	-	**87.2** (± 0.2)	**78.8** (± 0.2)	**59.8** (± 0.1)	77.4 (± 0.1)	**72.0** (± 0.2)	**70.4** (± 0.2)	**87.7** (± 0.3)

Table 3 Comparison of Segmentation Results on the S3DIS Dataset (Area-5)

Method	Year	Instance				Semantic	Panoptic		
		mAP_{50}	mAP	$mPrec_{50}$	$mRec_{50}$	mIoU	PQ	PQ^{th}	PQ^{st}
PointGroup [50]	2020	57.8	-	61.9	62.1	-	-	-	-
DyCo3D [51]	2021	64.3	64.2	-	-	-	-	-	-
SSTNet [52]	2021	59.3	42.7	65.5	64.2	-	-	-	-
DKNet [23]	2022	70.8	65.3	-	-	-	-	-	-
HAIS [53]	2021	71.1	65.0	-	-	-	-	-	-
TD3D [46]	2023	65.1	48.6	74.4	64.8	-	-	-	-
SoftGroup [47]	2022	66.1	51.6	73.6	66.6	-	-	-	-
PBNet [48]	2023	66.4	53.5	74.9	65.4	-	-	-	-
SPFormer [41]	2023	66.8	-	72.8	67.1	-	-	-	-
Mask3D [40]	2023	71.9	57.8	74.3	63.7	-	-	-	-
SegGCN [71]	2020	-	-	-	-	63.6	-	-	-
PAConv [44]	2021	-	-	-	-	66.6	-	-	-
PointTransformer [25]	2021	-	-	-	-	70.4	-	-	-
PointNeXt-XL [10]	2022	-	-	-	-	70.5	-	-	-
PointTransformerV2 [26]	2022	-	-	-	-	71.6	-	-	-
Stratified Transformer [28]	2022	-	-	-	-	72.0	-	-	-
Swin3D [70]	2025	58.6	-	-	-	72.5	-	-	-
OneFormer3D (Base) [42]	2024	72.0	58.7	79.7	73.0	72.4	62.2	58.4	65.5
COC (Ours)	-	**73.4** (± 0.1)	**59.7** (± 0.1)	**81.0** (± 0.2)	**73.9** (± 0.2)	**73.8** (± 0.2)	**63.0** (± 0.1)	**59.1** (± 0.1)	**66.2** (± 0.1)

Table 4 Comparison of Segmentation Results on the S3DIS Dataset (6-Fold Cross-Validation)

Method	Year	Instance				Semantic	Panoptic		
		mAP_{50}	mAP	$mPrec_{50}$	$mRec_{50}$	mIoU	PQ	PQ^{th}	PQ^{st}
PointGroup [50]	2020	64.0	-	69.6	69.2	-	-	-	-
HAIS [53]	2021	73.2	69.4	-	-	-	-	-	-
SSTNet [52]	2021	67.8	54.1	73.5	73.4	-	-	-	-
DKNet [23]	2022	75.3	71.1	-	-	-	-	-	-
TD3D [46]	2023	68.2	56.2	76.3	74.0	-	-	-	-
SoftGroup [47]	2022	68.9	54.4	75.3	69.8	-	-	-	-
SPFormer [41]	2023	69.2	-	74.0	71.1	-	-	-	-
PBNet [48]	2023	70.6	59.5	80.1	72.9	-	-	-	-
Mask3D [40]	2023	74.3	61.8	76.5	66.2	-	-	-	-
PointTransformer [25]	2021	-	-	-	-	73.5	-	-	-
PointNeXt-XL [10]	2022	-	-	-	-	74.9	-	-	-
OneFormer3D (base) [42]	2024	75.8	63.0	82.3	74.1	75.0	68.5	61.5	74.5
COC (Ours)	-	**76.9**	**63.7**	**84.1**	**75.2**	**75.8**	**69.2**	**61.9**	**75.6**

stated, the metric is semantic mIoU. The main comparison in Table 3 adopts the published OneFormer3D value of 72.4 mIoU. The cumulative

ablation table uses a 72.4 mIoU local baseline and reports 74.0 mIoU for the full COC configuration under the same protocol. SAG-CA and DSSA have decoder-level switches in the active implementation, whereas the LGDI comparison changes the regional pooling path. The subsequent design-choice tables summarize separately trained controlled variants; their labels describe those variants and are not claims that every historical option is exposed as a runtime flag in the active configuration. Scores from these separate tables should therefore not be interpreted as one additive sequence.

The cumulative component configurations are listed in Table 5 before their conditional effects are discussed.

Table 5 reports nested configurations rather than independent one-factor-at-a-time comparisons. The baseline in row 0 obtains 72.4 mIoU. Adding SAG-CA alone (row 1) gives 73.1, a change of +0.7; adding LGDI to that configuration (row 2) gives 73.5, which is +0.4 over row 1 and +1.1 over the baseline; adding DSSA (row 3) gives the final 74.0, a further +0.5 over row 2 and +1.6 over the baseline. These differences are conditional effects of the cumulative sequence and should not be interpreted as isolated causal contributions.

The controlled SAG-CA design variants are reported in Table 6. They are evaluated under the same Area-5 protocol as the cumulative configurations.

Table 6 compares controlled SAG-CA variants. In the active implementation, "Attn Pooling" denotes learned slot assignment in the state bank, and "Adaptive" denotes the feature-wise local/global gate. The standard cross-attention baseline reaches 72.4 mIoU. Mean pooling with fixed summation reaches 73.2, mean pooling with the adaptive gate reaches 73.5, slot assignment without a gate reaches 73.2, and the complete slot-assignment variant with the adaptive gate reaches 74.0. The controlled comparisons support the combined use of learned state construction and gated fusion in this setting, while the table should not be read as a proof that either component is independently necessary in every configuration.

The depth-aggregation variants evaluated for DSSA are listed in Table 7.

Table 7 compares controlled depth-aggregation variants. The weighted-sum baseline obtains 72.4 mIoU, while terminal read with random initialization reaches 72.5. Adding bidirectional depth evolution raises the score to 73.3, and using the small-output initialization of the active implementation further raises it to 74.0. These controlled results are consistent with benefits from bidirectional depth evolution and the tested initialization, but they do not provide a separate theoretical guarantee for either choice.

The regional pooling comparison for LGDI is given in Table 8. In the formulas in this table, $x_i \in \mathbb{R}^D$ denotes the feature vector of the i-th sample in a region, N is the number of samples, and $w_i \geq 0$ denotes the learned scalar weight assigned to that sample. Thus, scatter mean and scatter max operate on the same feature vectors x_i, while linear weighting forms the weighted sum $\sum_i w_i x_i$; the table compares the mechanisms and reports their measured mIoU values.

Table 8 compares different superpoint pooling strategies. Scatter mean achieves 72.4 mIoU, scatter max reaches 72.2, and linear weighting improves the result to 73.5. The proposed LGDI achieves the best result of 74.0 mIoU, outperforming scatter mean by 1.6, scatter max by 1.8, and linear weighting by 0.5 mIoU. These results support the effectiveness of the proposed positive-measure integral for regional aggregation under this protocol.

Taken together, the ablations are consistent with the intended design: learned positive-measure aggregation improves over simple regional statistics, the state-bank variants benefit from feature-wise local/global fusion, and bidirectional depth evolution improves the decoder trajectory under the tested protocol. These controlled comparisons support the module design without implying that every gain is independent or statistically significant.

5 Conclusion

In this paper, we revisited unified 3D indoor scene segmentation from the perspective of discretized integration and state evolution, and proposed Calculus on Clouds (COC), a hybrid SSM–Transformer framework built upon OneFormer3D. COC targets three bottlenecks of existing unified segmentation systems through lightweight

Table 5 Impact of Core Components on the S3DIS Area-5 Benchmark

ID	Base Model	SAG-CA	LGDI	DSSA	mIoU (%)
0	Baseline	-	-	-	72.4
1	Baseline	✓	-	-	73.1
2	Baseline	✓	✓	-	73.5
3	Baseline	✓	✓	✓	**74.0**

Table 6 Design Choices in the SAG-CA Module on S3DIS Area-5

ID	Method	Global Aggregator	Fusion Gate	mIoU (%)
1	Standard Cross-Attention	-	-	72.4
2	SAG-CA (Mean, Fixed)	Mean Pooling	Fixed Sum	73.2
3	SAG-CA (Mean, Gated)	Mean Pooling	Adaptive	73.5
4	SAG-CA (Slot, No Gate)	Attn Pooling	-	73.2
5	**SAG-CA (Ours)**	Attn Pooling	Adaptive	**74.0**

Table 7 Controlled Depth-Aggregation Variants for DSSA on S3DIS Area-5

ID	Aggregation Base	SSM Enhancement	Initialization	mIoU (%)
1	Weighted Sum	-	-	72.4
2	Terminal Read	-	Random	72.5
3	Terminal Read	Bi-directional	Random	73.3
4	**Terminal Read**	**Bi-directional**	**Small-Output-Init**	**74.0**

Table 8 Comparison of Regional Aggregation Strategies for LGDI on S3DIS Area-5

ID	Pooling Method	Mechanism	mIoU (%)
1	Scatter Mean	$\frac{1}{N}\sum x_i$	72.4
2	Scatter Max	$\max(x_i)$	72.2
3	Linear Weighting	$\sum(w_i \cdot x_i)$	73.5
4	**LGDI (Ours)**	**Positive-measure integral**	**74.0**

and complementary modules. Lean Gated Discretized Integration (LGDI) reformulates superpoint pooling as a learnable discretized integration over irregular samples, replacing fixed statistics with adaptive modulation and positive scalar densities. State-Aware Gated Cross-Attention (SAG-CA) augments local cross-attention with a query-addressable global state branch, so that each query can read scene-level context through learned state modulation. Depth-wise Selective Scan Aggregation (DSSA) models the decoder-layer query trajectory as a short bidirectional state evolution, allowing the final prediction to benefit from both shallow geometric cues and deep semantic refinement.

Experiments on ScanNetV2 and S3DIS show improvements over the reported OneFormer3D references across semantic, instance, and panoptic metrics. The ablations are consistent with complementary effects from positive-measure regional aggregation, state-bank fusion, and bidirectional depth evolution, while the controlled comparisons do not establish independent causal effects for every design choice. Because the added paths are lightweight relative to the backbone and decoder, COC remains compatible with unified indoor 3-D perception pipelines.

Several limitations remain and point to future work. First, the current evaluation is restricted to indoor benchmarks; extending COC to large-scale outdoor and dynamic scenes, such as autonomous-driving LiDAR, would better validate the generality of the discretized-integration view. Second, future work may explore deployment-oriented efficiency under different hardware settings, including FLOPs, memory footprint, and on-device latency

on resource-constrained platforms. Third, the proposed modules adopt SSM-inspired discretization in three targeted places rather than a full sequential state model, and a deeper integration of state-space modeling with geometric neighborhood structure is a promising direction. We hope that the integration- and state-evolution-based viewpoint introduced here offers a useful complement to attention-centric designs for irregular 3D point cloud understanding.

References

[1] Xu, R., Wang, X., Wang, T., Chen, Y., Pang, J., & Lin, D. (2024, September). Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision* (pp. 131–147). Springer Nature Switzerland.

[2] Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., & Kwok, N. M. (2022). Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12), 4338–4357.

[3] Seichter, D., Köhler, M., Lewandowski, B., Wengefeld, T., & Gross, H.-M. (2021). Efficient RGB-D semantic segmentation for indoor scene analysis. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 13525–13531).

[4] Xie, Y., Tian, J., & Zhu, X. X. (2020). Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geoscience and remote sensing magazine*, 8(4), 38–59.

[5] Sarker, S., Sarker, P., Stone, G., Gorman, R., Tavakkoli, A., Bebis, G., & Sattarvand, J. (2024). A comprehensive overview of deep learning techniques for 3D point cloud classification and semantic segmentation. *Machine Vision and Applications*, 35(4), 67.

[6] Seichter, D., Langer, P., Wengefeld, T., Lewandowski, B., Höchemer, D., & Gross, H.-M. (2022). Efficient and robust semantic mapping for indoor environments. In *2022 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 9221–9227).

[7] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 652–660).

[8] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, 38(5), 1–13.

[9] Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). KPConv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 6411–6420).

[10] Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H. A. K., Elhoseiny, M., & Ghanem, B. (2022). PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In *NeurIPS 2022* (pp. 1655–1669).

[11] Lin, H., Zheng, X., Li, L., Chao, F., Wang, S., Wang, Y., & Ji, R. (2023). Meta architecture for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 17682–17691).

[12] Zhang, H., Wang, C., Yu, L., Tian, S., Ning, X., & Rodrigues, J. (2024). PointGT: A method for point-cloud classification and segmentation based on local geometric transformation. *IEEE Transactions on Multimedia*, 26, 8052–8062.

[13] Maturana, D., & Scherer, S. (2015). VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *IROS 2015* (pp. 922–928).

[14] Choy, C., Gwak, J., & Savarese, S. (2019). 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR 2019* (pp. 8177–8186).

[15] Tang, H., Yang, S., Liu, Z., Hong, K., Yu, Z., Li, X., & Han, S. (2023, October). Torchsparse++: Efficient training and inference framework for sparse convolution on gpus. In *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture* (pp. 225–239).

[16] Dong, W., Lao, Y., Kaess, M., & Koltun, V. (2022). ASH: A modern framework for parallel spatial hashing in 3D perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5), 5417–5435.

[17] Chen, Y., Liu, J., Zhang, X., Qi, X., & Jia, J. (2023). Largekernel3d: Scaling up kernels in 3d sparse cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13488–13498).

[18] Chen, Y., Liu, J., Zhang, X., Qi, X., & Jia, J. (2023). VoxelNeXt: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 21674–21683).

[19] Li, B., Zhang, T., & Xia, T. (2016). Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*.

[20] Kong, L., Liu, Y., Chen, R., Ma, Y., Zhu, X., Li, Y., & Liu, Z. (2023). Rethinking range view representation for lidar segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 228–240).

[21] Mao, J., Shi, S., Wang, X., & Li, H. (2023). 3D object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, 131(8), 1909–1963.

[22] Song, Z., Liu, L., Jia, F., Luo, Y., Jia, C., Zhang, G., & Wang, L. (2024). Robustness-aware 3d object detection in autonomous driving: A review and outlook. *IEEE Transactions on Intelligent Transportation Systems*.

[23] Wu, Y., Shi, M., Du, S., Lu, H., Cao, Z., & Zhong, W. (2022). 3D instances as 1D kernels. In *European Conference on Computer Vision (ECCV)* (pp. 235–252). Springer.

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

[25] Zhao, H., Jiang, L., Jia, J., Torr, P. H., & Koltun, V. (2021). Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16259–16268).

[26] Wu, X., Lao, Y., Jiang, L., Liu, X., & Zhao, H. (2022). Point transformer v2: Grouped vector attention and partition-based pooling. In *Advances in Neural Information Processing Systems* (pp. 33330–33342).

[27] Park, C., Jeong, Y., Cho, M., & Park, J. (2022). Fast point transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16949–16958).

[28] Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., & Jia, J. (2022). Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 8500–8509).

[29] Li, Z., Ai, Y., Lu, J., Wang, C., Deng, J., Chang, H., & Zhang, T. (2025, April). Pamba: Enhancing Global Interaction in Point Clouds via State Space Model. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 5092–5100).

[30] Gu, A., Goel, K., & Ré, C. (2022). Efficiently Modeling Long Sequences with Structured State Spaces (S4). In *International Conference on Learning Representations (ICLR)*.

[31] Smith, J. T., Al-Ghorabie, K., Gupta, A., Gu, A., & Ré, C. (2023). Hungry Hungry Hippos: Towards Optimal Latency and Throughput for High-Quality Sequence Modeling (S5). In

International Conference on Learning Representations (ICLR).

[32] Gu, A., & Dao, T. (2023). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv preprint arXiv:2312.00752.*

[33] Sun, H., Shi, Y., Han, J., & Shi, H. (2024). Point Mamba: A Simple State Space Model for Point Cloud Analysis. *arXiv preprint arXiv:2403.00785.*

[34] Zhang, T., Yuan, H., Qi, L., Zhang, J., Zhou, Q., Ji, S., & Li, X. (2025, April). Point cloud mamba: Point cloud learning via state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 10121–10130).

[35] Liang, D., Zhou, X., Xu, W., Zhu, X., Zou, Z., Ye, X., & Bai, X. (2024). PointMamba: A simple state space model for point cloud analysis. In *Advances in Neural Information Processing Systems (NeurIPS)* (pp. 32653–32677).

[36] Lu, D., Xu, L., Zhou, J., Gao, K., Gong, Z., & Zhang, D. (2025). 3D-UMamba: 3D U-Net with state space model for semantic segmentation of multi-source LiDAR point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 136, 104401.

[37] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV).*

[38] Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1290–1299).

[39] Chen, B., Li, J., Wang, C., & Xu, K. (2022). Masked-attention Mask Transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 14375–14385).

[40] Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., & Leibe, B. (2023). Mask3D: Mask transformer for 3D semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 8216–8223).

[41] Sun, J., Qing, C., Tan, J., & Xu, X. (2023). Superpoint transformer for 3D scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 2393–2401).

[42] Kolodiazhnyi, M., Vorontsova, A., Konushin, A., & Rukhovich, D. (2024). OneFormer3D: One transformer for unified point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 20943–20953).

[43] Yan, X., Zheng, C., Li, Z., Wang, S., & Cui, S. (2020). PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5589–5598).

[44] Xu, M., Ding, R., Zhao, H., & Qi, X. (2021). PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3173–3182).

[45] Deng, X., Zhang, W., Ding, Q., & Zhang, X. (2023). PointVector: A vector representation in point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 9455–9465).

[46] Kolodiazhnyi, M., Vorontsova, A., Konushin, A., & Rukhovich, D. (2024). Top-down beats bottom-up in 3d instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 3566–3574).

[47] Vu, T., Kim, K., Luu, T. M., Nguyen, T., & Yoo, C. D. (2022). SoftGroup for 3D instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2708–2717).

[48] Zhao, W., Yan, Y., Yang, C., Ye, J., Yang, X., & Huang, K. (2023). Divide and conquer: 3D point cloud instance segmentation with point-wise binarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 562–571).

[49] Ngo, T. D., Hua, B.-S., & Nguyen, K. (2023). ISBNet: a 3D point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 13550–13559).

[50] Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.-W., & Jia, J. (2020). PointGroup: Dual-set point grouping for 3D instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4867–4876).

[51] He, T., Shen, C., & Van Den Hengel, A. (2021). DyCo3D: Robust instance segmentation of 3D point clouds through dynamic convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 354–363).

[52] Liang, Z., Li, Z., Xu, S., Tan, M., & Jia, K. (2021). Instance segmentation in 3D scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 2783–2792).

[53] Chen, S., Fang, J., Zhang, Q., Liu, W., & Wang, X. (2021). Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 15467–15476).

[54] Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., & Ré, C. (2021). Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *Advances in Neural Information Processing Systems (NeurIPS)* (pp. 572–585).

[55] Kailath, T. (1980). *Linear systems* (Vol. 156): Prentice-Hall Englewood Cliffs, NJ.

[56] Brockett, R. W. (2015). *Finite dimensional linear systems*: Society for Industrial and Applied Mathematics.

[57] Gu, A., Goel, K., & Ré, C. (2021). Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

[58] Smith, J. T., Warrington, A., & Linderman, S. W. (2023). Simplified state space layers for sequence modeling. In *International Conference on Learning Representations (ICLR)*.

[59] Dao, T., Fu, D. Y., Saab, K. K., Thomas, A. W., Rudra, A., & Ré, C. (2023). Hungry hungry hippos: Towards language modeling with state space models. In *International Conference on Learning Representations (ICLR)*.

[60] Nguyen, E., Goel, K., Gu, A., Downs, G., Shah, P., Dao, T., & Ré, C. (2022). S4ND: Modeling images and videos as multidimensional signals with state spaces. In *Advances in Neural Information Processing Systems (NeurIPS)* (pp. 2846–2861).

[61] Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., & Wang, X. (2024). Vision Mamba: Efficient visual representation learning with bidirectional state space model. In *International Conference on Machine Learning (ICML)*.

[62] Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., & Liu, Y. (2024). VMamba: Visual State Space Model. In *Advances in Neural Information Processing Systems (NeurIPS)* (pp. 103031–103063).

[63] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., . . . Guo, B. (2021). Swin Transformer: Hierarchical vision transformer

using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 10012–10022).

[64] Wang, W., Xie, E., Li, X., Fan, D. P., Song, K., Liang, D., . . . Shao, L. (2021). Pyramid Vision Transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 568–578).

[65] Wang, Y., Zhang, X., Sun, T., Yuan, X., Jiang, G., & Shen, H. T. (2021). UP-DETR: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1602–1612).

[66] Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., & Nießner, M. (2017). ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5828–5839).

[67] Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016). 3D semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1534–1543).

[68] Wu, S.-C., Wald, J., Tateno, K., Navab, N., & Tombari, F. (2021). SceneGraphFusion: Incremental 3D scene graph prediction from RGB-D sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7515–7525).

[69] Yang, Z., & Liu, C. (2021). Tupper-map: Temporal and unified panoptic perception for 3d metric-semantic mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1094–1101). IEEE.

[70] Yang, Y.-Q., Guo, Y.-X., Xiong, J.-Y., Liu, Y., Pan, H., Wang, P.-S., . . . Guo, B. (2025). Swin3D: A pretrained transformer backbone for 3D indoor scene understanding. *Computational Visual Media*, 11(1), 83–101.

[71] Lei, H., Akhtar, N., & Mian, A. (2020). SegGCN: Efficient 3D point cloud segmentation with fuzzy spherical kernel. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11611–11620).