LDSGSEG: LOCAL DISTRIBUTION ENCODING AND SEMANTIC GUIDANCE FOR FEW-SHOT POINT CLOUD SEMANTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-shot 3D point cloud semantic segmentation aims to segment novel classes in query point clouds given only a few annotated support samples. However, existing prototype-based methods still encounter two limitations: i) they do not explicitly characterize the spatial distribution of neighboring points; ii) query–prototype matching relies mainly on visual representations, without fully exploiting the category-level semantic guidance provided by class names. To address these issues, we propose LDSGSeg, a novel framework that integrates local distribution encoding with category-level semantic guidance. Specifically, to solve the first issue, PCA-Guided Local Distribution Encoding derives local distribution descriptors from local covariance eigenvalues to enrich local point representations. To address the second issue, Point-to-Text Cross-Attention introduces category-level semantic information into query point features by modeling their relationships with the text representations of foreground classes. Furthermore, Text-Guided Prototype Channel Calibration employs the corresponding class text embedding to adaptively reweight the channels of each support-derived visual prototype, thereby providing category-level semantic guidance for prototype matching. Extensive experiments on the S3DIS and ScanNet datasets demonstrate that LDSGSeg consistently improves segmentation performance under various few-shot settings.

1 INTRODUCTION

Point cloud semantic segmentation aims to assign a semantic label to each point in a 3D scene (Wang et al., 2022). It has been widely applied to autonomous driving (Zhao et al., 2024; Chib & Singh, 2023), robotic perception (Soori et al., 2023; Goel & Gupta, 2020), and augmented reality (Devagiri et al., 2022; Sereno et al., 2022). Despite recent advances in fully supervised approaches (Qi et al., 2017a; Wang et al., 2019; Zhao et al., 2021a), these methods rely heavily on large-scale datasets with dense point-level annotations, the acquisition of which requires substantial time and manual effort (Jiang et al., 2023). Consequently, the high annotation cost not only limits the applicability of fully supervised segmentation in data-scarce scenarios but also hinders the adaptation of segmentation models to novel classes. Therefore, learning point cloud segmentation models that can generalize to novel classes from only a few annotated samples remains an critical research challenge.

To address this challenge, few-shot learning has attracted increasing attention in point cloud semantic segmentation (Zhao et al., 2021b; Li et al., 2024; Xiong et al., 2024). Specifically, few-shot 3D point cloud semantic segmentation aims to segment novel classes in query point clouds with only a few annotated support samples. Several recent studies have explored a prototype-based paradigm for few-shot 3D point cloud semantic segmentation, in which query points are classified by comparing their features with class prototypes derived from labeled support features (Zhao et al., 2021b; Mao et al., 2022; He et al., 2023). Building on this paradigm, subsequent studies have strengthened feature representations through interactions between support and query features (Mao et al., 2022; Zheng et al., 2024; An et al., 2024), whereas prototype adaptation, rectification, and progressive refinement have been explored to improve prototype quality (He et al., 2023; Zhu et al., 2023; Wei et al., 2024; 2025; Zhou et al., 2026; Wang et al., 2026b). Recent studies have also introduced multimodal or language knowledge into few-shot and generalized few-shot settings (An et al., 2025b;a; Li et al., 2025; Wang et al., 2026a). Beyond these efforts, local neighborhood modeling has been investigated to

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.

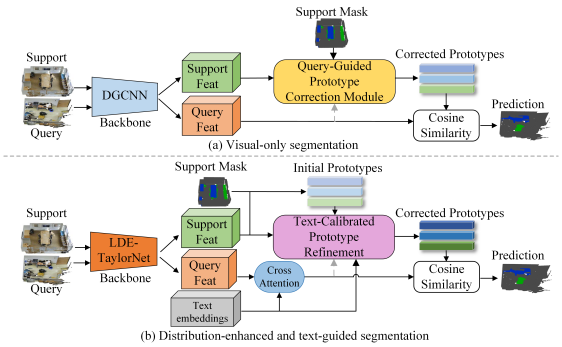

Figure 1: Top: A representative visual-only framework extracts support and query features without explicit local distribution encoding, while query–prototype matching relies only on visual information. Bottom: LDSGSeg introduces local distribution encoding into the point encoder and uses class-name embeddings to guide query features and calibrate support-derived prototypes.

enhance local feature representations in few-shot settings (Wang et al., 2025a;b). Overall, these efforts have improved point representations and prototype quality in few-shot point cloud segmentation.

Despite these advances, existing prototype-based methods still face two limitations. As illustrated in Fig. 1(a), a representative visual-only framework extracts support and query features for prototype construction and matching, without explicitly incorporating local point distribution information or category-level semantic guidance. Specifically, local neighborhood aggregation does not explicitly characterize how neighboring points are distributed in space, while query–prototype matching remains largely dependent on visual representations. In contrast, Fig. 1(b) illustrates that LDSGSeg introduces local distribution encoding into the point encoder and class-name semantics into both query features and support-derived prototypes.

Motivated by these observations, we propose LDSGSeg to explicitly model local point distributions and introduce category-level semantic guidance into query–prototype matching. Within the point encoder, PCA-derived distribution descriptors are incorporated into local feature extraction, allowing point features to explicitly capture the distribution patterns of neighboring points. For semantic guidance, class-name embeddings provide category-level information for the current episode. Point-to-Text Cross-Attention introduces such information into query features by modeling their relationships with the text representations of foreground classes. Furthermore, Text-Guided Prototype Channel Calibration uses the corresponding class text embedding to adaptively reweight support-derived visual prototypes. In this way, LDSGSeg combines explicit local distribution information with category-level semantic guidance for novel-class segmentation.

The main contributions of this work are summarized as follows:

- We propose LDSGSeg, a few-shot 3D point cloud semantic segmentation framework that combines local distribution encoding with category-level semantic guidance to enhance point representations and calibrate support-derived prototypes.

- PCA-Guided Local Distribution Encoding is introduced to explicitly model local point distributions, while incorporate the distribution information into local feature extraction. Point-to-Text Cross-Attention is designed to incorporate category-level semantic information from class text embeddings into query features, providing complementary semantic guidance for query features.

- Text-Guided Prototype Channel Calibration is proposed to adaptively calibrate the channel responses of support-derived visual prototypes with class-level semantic guidance. Extensive experiments on S3DIS and ScanNet demonstrate the effectiveness of LDSGSeg, which achieves strong performance across various few-shot settings.

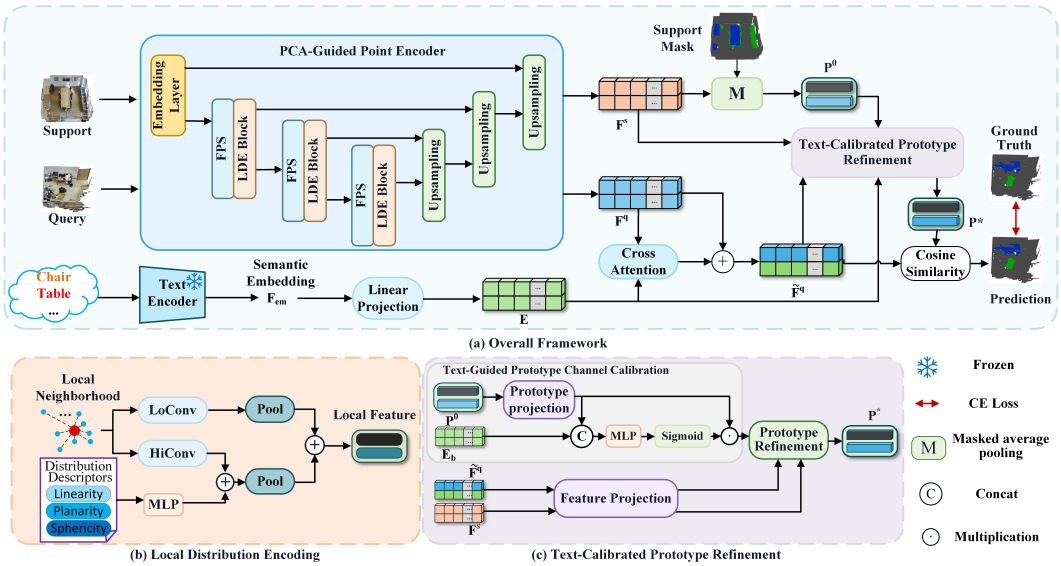

Figure 2: Overview of LDSGSeg. (a) The point Encoder extracts support and query features, while frozen class-name text embeddings provide category-level semantic guidance for query feature enhancement and prototype calibration. (b) Each LDE Block injects local distribution descriptors into the HiConv branch. (c) Text-Calibrated Prototype Refinement calibrates support-derived prototypes before final prediction.

2 Related Work

2.1 3D Point Cloud Semantic Segmentation

3D point cloud semantic segmentation aims to predict a semantic label for each point in a 3D scene. A key challenge is to learn discriminative representations from unordered point sets with irregular spatial distributions. PointNet directly processed raw point clouds using shared MLPs and a symmetric aggregation function (Qi et al., 2017a). To improve local feature modeling, PointNet++, DGCNN, and KPConv capture neighborhood information through hierarchical grouping, dynamic graph convolution, and kernel point convolution, respectively (Qi et al., 2017b; Wang et al., 2019; Thomas et al., 2019). Subsequently, Point Transformer and its variants employ self-attention to model contextual relationships among points (Zhao et al., 2021a; Lai et al., 2022), while PointMamba explores state-space modeling for efficient global information propagation (Liang et al., 2024). Despite these advances, most fully supervised methods still depend on large-scale point-wise annotations, which limits their applicability in data-scarce settings.

2.2 Few-Shot Point Cloud Semantic Segmentation

Recent studies have increasingly explored few-shot point cloud semantic segmentation under limited support annotations (Mao et al., 2022; Zhang et al., 2023; He et al., 2023; An et al., 2025b). As a pioneering work, AttMPTI combines episodic training with multiple prototypes and label propagation for query point cloud segmentation (Zhao et al., 2021b). Existing methods can be broadly divided into three directions: feature enhancement, prototype optimization, and architectural design. For feature enhancement, BFG promotes support–query information exchange through bidirectional feature globalization (Mao et al., 2022). For prototype optimization, DENet reduces prototype ambiguity through bidirectional aggregation and consistency purification (Xiong et al., 2024); subsequent methods further improve prototype quality through heterogeneous information, self-training, or coarse-to-fine refinement (Wei et al., 2025; Zhou et al., 2026; Wang et al., 2026b). PAP3D introduces category semantics through visual–semantic prototype interaction (He et al., 2023), while GPCPR combines LLM-generated category descriptions with pseudo-query context to improve prototype representations (Wei et al., 2024). More recent studies exploit vision–language models or language priors in few-shot, generalized few-shot, and unified few-/zero-shot settings (An et al., 2025b;a; Li

et al., 2025; Wang et al., 2026a;c). For architectural design, TaylorSeg models local neighborhoods using Taylor series-inspired convolutions (Wang et al., 2025a), whereas DyPolySeg combines dynamic polynomial convolution with Mamba for local feature modeling and global context propagation (Wang et al., 2025b). Current methods, however, still provide limited explicit representation of local point distributions, while category-level semantics have rarely been used to guide both query features and support-derived prototypes.

3 METHOD

This section presents LDSGSeg, a few-shot 3D point cloud semantic segmentation framework that integrates local distribution encoding with category-level semantic guidance. We first formulate the episodic segmentation problem, then describe the overall architecture. The main components are then described in detail, including the PCA-Guided Local Distribution Encoding, Point-to-Text Cross-Attention, and Text-Guided Prototype Channel Calibration.

3.1 PROBLEM DEFINITION

Following the standard episodic protocol, the category set $\mathcal{C}$ is divided into disjoint base and novel classes, $\mathcal{C}_{\text{base}}$ and $\mathcal{C}_{\text{novel}}$, where $\mathcal{C}_{\text{base}} \cap \mathcal{C}_{\text{novel}} = \emptyset$. Training episodes are sampled from $\mathcal{C}_{\text{base}}$, while evaluation episodes are constructed from $\mathcal{C}_{\text{novel}}$. For an N-way K-shot episode, the support set is $\mathcal{S} = \{(\mathbf{X}_s^{n,k}, \mathbf{M}_s^{n,k})\}_{n=1,k=1}^{N,K}$, where $\mathbf{X}_s^{n,k} \in \mathbb{R}^{N_s \times D}$ and $\mathbf{M}_s^{n,k} \in \{0,1\}^{N_s}$ denote a support point cloud and its binary foreground mask, respectively.

The query set is $\mathcal{Q} = \{(\mathbf{X}_q^j, \mathbf{Y}_q^j)\}_{j=1}^{N_{\text{qry}}}$, where $\mathbf{X}_q^j \in \mathbb{R}^{N_q \times D}$ is a query point cloud and $\mathbf{Y}_q^j \in \{0, \ldots, N\}^{N_q}$ contains point-wise labels available only for supervision or evaluation. Support and query point clouds share the same N foreground classes, and each point contains 3D coordinates, RGB values, and normalized spatial coordinates. Given the labeled support set, the model predicts each query-point label by measuring the cosine similarity between query features and class prototypes, where label 0 denotes background and labels $1, \ldots, N$ denote the foreground classes.

3.2 OVERALL ARCHITECTURE

The overall architecture of LDSGSeg is illustrated in Fig. 2. Given the support and query point clouds, the shared LDE-Enhanced Point Encoder extracts support features F^s and query features F^q. The support masks are used to generate the initial prototypes P^0 from F^s through masked average pooling. Meanwhile, a frozen text encoder extracts semantic embeddings F_{em} from the class names of the current episode, which are projected into episode text embeddings E by a learnable linear layer. For query enhancement, point-to-text cross-attention uses E to introduce category-level semantic information into F^q to obtain semantic-enhanced query features $\widetilde{F}^q$. For prototype calibration, E is used to adjust the channel responses of P^0, while the calibrated prototypes are further refined with support-query correlations to produce P^*. The resulting semantic-enhanced query features $\widetilde{F}^q$ are matched with the refined prototypes P^* by inner product to predict the point-wise labels of the query point cloud.

3.3 PCA-GUIDED LOCAL DISTRIBUTION ENCODING

The PCA-Guided Local Distribution Encoding (LDE) module is designed to enhance point representations by incorporating explicit local distribution information during neighborhood feature aggregation. As shown in Fig. 2(b), each LDE Block follows the backbone architecture with two neighborhood-processing branches, denoted as LoConv and HiConv. After being transformed by an MLP, the local distribution descriptors are fused with the features produced by HiConv before pooling. The outputs of the two branches are then aggregated to obtain the local representation.

At encoding stage l, the local neighborhood of center point i is denoted by $\mathcal{N}_i^{(l)}$, while $\mathbf{r}_{i,k}^{(l)} \in \mathbb{R}^3$ represents the normalized relative coordinate of its k-th neighbor. To represent the spatial distribution of the neighborhood, we construct a covariance matrix from these normalized relative coordinates.

The local covariance matrix is given as:

$$\mathbf{\Sigma}_i^{(l)} = \frac{1}{K_{\mathrm{nn}} - 1} \sum_{k=1}^{K_{\mathrm{nn}}} \left(\mathbf{r}_{i,k}^{(l)} - \bar{\mathbf{r}}_i^{(l)}\right) \left(\mathbf{r}_{i,k}^{(l)} - \bar{\mathbf{r}}_i^{(l)}\right)^\top. \tag{1}$$

Here, $\bar{\mathbf{r}}_i^{(l)}$ is the centroid of the local neighborhood, and K_{nn} is the number of neighboring points. We then perform eigendecomposition on the covariance matrix to obtain its eigenvalues:

$$\mathbf{\Sigma}_i^{(l)} = \mathbf{U}_i^{(l)} \operatorname{diag}\left(\lambda_{i,1}^{(l)}, \lambda_{i,2}^{(l)}, \lambda_{i,3}^{(l)}\right) \left(\mathbf{U}_i^{(l)}\right)^\top, \tag{2}$$

Here, $U_i^{(l)}$ contains the corresponding eigenvectors, while the eigenvalues are ordered as $\lambda_{i,1}^{(l)} \leq \lambda_{i,2}^{(l)} \leq \lambda_{i,3}^{(l)}$. The eigenvalues are normalized by their sum as $\widetilde{\lambda}_{i,r}^{(l)} = \lambda_{i,r}^{(l)} / (\sum_{u=1}^{3} \lambda_{i,u}^{(l)} + \varepsilon)$, where $r \in \{1, 2, 3\}$. Based on the normalized eigenvalues, we compute three local distribution descriptors: linearity, planarity, and sphericity:

$$\ell_i^{(l)} = \frac{\widetilde{\lambda}_{i,3}^{(l)} - \widetilde{\lambda}_{i,2}^{(l)}}{\widetilde{\lambda}_{i,3}^{(l)} + \varepsilon}, \qquad \pi_i^{(l)} = \frac{\widetilde{\lambda}_{i,2}^{(l)} - \widetilde{\lambda}_{i,1}^{(l)}}{\widetilde{\lambda}_{i,3}^{(l)} + \varepsilon}, \qquad s_i^{(l)} = \frac{\widetilde{\lambda}_{i,1}^{(l)}}{\widetilde{\lambda}_{i,3}^{(l)} + \varepsilon}. \tag{3}$$

These descriptors capture whether the local neighborhood exhibits line-like, planar, or volumetric point distributions.

The descriptor vector $\mathbf{d}_i^{(l)} = [\ell_i^{(l)}, \pi_i^{(l)}, s_i^{(l)}]$ is mapped by a lightweight point-wise encoder Φ_l to obtain the distribution feature $\mathbf{g}_i^{(l)}$:

$$\mathbf{g}_i^{(l)} = \Phi_l(\mathbf{d}_i^{(l)}). \tag{4}$$

Here, Φ_l consists of two point-wise convolution layers, each followed by batch normalization and ReLU activation. Its output dimension matches the feature dimension of HiConv at stage l. The distribution feature is broadcast to all neighboring points and fused with the HiConv response using a learnable scaling parameter:

$$\widehat{\mathbf{H}}_{i,k}^{(l)} = \mathbf{H}_{i,k}^{(l)} + \alpha_l \mathbf{g}_i^{(l)}. \tag{5}$$

Here, $\mathbf{H}_{i,k}^{(l)}$ denotes the HiConv response of the k-th neighbor. The scaling parameter α_l is initialized to zero, allowing the network to gradually learn the contribution of the distribution feature. After fusion, the outputs of LoConv and the distribution-enhanced HiConv are pooled and aggregated as:

$$\mathbf{F}_i^{(l)} = \mathrm{Pool}\left(\mathbf{L}_{i,k}^{(l)}\right) + \mathrm{Pool}\left(\widehat{\mathbf{H}}_{i,k}^{(l)}\right). \tag{6}$$

Here, $\mathbf{L}_{i,k}^{(l)}$ denotes the LoConv response, while Pool denotes neighborhood max pooling followed by nonlinear activation. In this way, LDE produces distribution-enhanced local representations for subsequent prototype construction and query–prototype matching.

3.4 Point-to-Text Cross-Attention

Although LDE enhances local point representations, the query features are still mainly derived from visual information and may lack category-level semantic guidance from the current episode. To address this limitation, Point-to-Text Cross-Attention introduces foreground class semantics into the query features by modeling their relationships with the corresponding text embeddings.

For each dataset class, we use the prompt "a point cloud scan of a {class}." to extract a 512-dimensional semantic embedding with a frozen CLIP text encoder. For an N-way episode, the embeddings of the N foreground classes are retrieved as $F_{\mathrm{em}} \in \mathbb{R}^{N \times 512}$. A learnable linear layer projects F_{em} into the point-feature space to obtain the episode text embeddings E:

$$E = F_{\mathrm{em}} W_e^\top + b_e, \qquad E \in \mathbb{R}^{N \times d}. \tag{7}$$

Here, d denotes the point-feature dimension. The text encoder remains frozen, while the projection layer is optimized during episodic training.

Given query features $F^q \in \mathbb{R}^{N_q \times d}$, F^q serves as the queries, while E serves as both keys and values. We employ multi-head cross-attention with $H = 4$ heads. For the h-th attention head, the semantic context is computed as

$$C_h^q = \mathrm{Softmax}\left(\frac{(F^q W_Q^h)(EW_K^h)^\top}{\sqrt{d_h}}\right)(EW_V^h), \qquad h = 1, \ldots, H. \tag{8}$$

Here, $d_h = d/H$ denotes the feature dimension of each attention head. The attention matrix implicitly measures the correspondence between each query point and the foreground classes in the current episode. The outputs of all heads are concatenated and projected to obtain the text-induced semantic context, $C^q = \mathrm{Concat}(C_1^q, \ldots, C_H^q)W_O$. In our implementation, $d = 128$ and $H = 4$, giving $d_h = 32$. The resulting semantic context is then fused with the original query features through a scaled residual connection:

$$\widetilde{F}^q = F^q + \beta C^q. \tag{9}$$

Here, β is a learnable scaling parameter initialized to zero, and $\widetilde{F}^q$ denotes the semantic-enhanced query features. By allowing query features to interact with the foreground class embeddings of the current episode, this module provides category-level guidance before query-prototype matching.

3.5 Text-Guided Prototype Channel Calibration

Class prototypes are constructed from annotated support points, serving as category representations for query–prototype matching. However, prototypes estimated from only a few support samples may be sensitive to sparse annotations or episode-specific appearance variations. To improve prototype reliability, we introduce Text-Guided Prototype Channel Calibration, which uses category-level semantic information to adaptively adjust the channel responses of support-derived prototypes.

For class c, the initial prototype is obtained by masked average pooling over the support features:

$$\mathbf{p}_c^0 = \frac{1}{|\Omega_c|}\sum_{i \in \Omega_c} \mathbf{f}_{s,i}, \qquad c \in \{0, 1, \ldots, N\}. \tag{10}$$

Here, Ω_c denotes the set of support points belonging to class c, where $c = 0$ represents the background class, while $c \in \{1, \ldots, N\}$ corresponds to the foreground classes of the current episode. After normalization, the prototypes are stacked into the initial prototype matrix $P^0 = [\mathbf{p}_0^0, \mathbf{p}_1^0, \ldots, \mathbf{p}_N^0]^\top \in \mathbb{R}^{(N+1)\times d}$.

Since E contains only foreground class embeddings, a zero vector is used as the background semantic placeholder, forming $E_b = [\mathbf{0}_{\mathrm{bg}}; E] \in \mathbb{R}^{(N+1)\times d}$. The initial prototype matrix P^0 is first projected into the prototype feature space:

$$\bar{P} = P^0 W_p^\top + b_p. \tag{11}$$

The projected visual prototypes are concatenated with their corresponding semantic embeddings to generate channel-wise calibration weights:

$$G = \sigma\left(\mathrm{MLP}([\bar{P} \| E_b])\right). \tag{12}$$

Here, $\|$ denotes channel concatenation, σ is the sigmoid function. The resulting $G \in \mathbb{R}^{(N+1)\times d}$ provides an adaptive weight for each prototype channel. The calibrated prototypes are then obtained by residual multiplicative modulation:

$$P^c = \bar{P} \odot (1 + G). \tag{13}$$

This operation preserves the visual prototype information while adaptively adjusting its channel responses according to the corresponding class semantics.

After semantic calibration, the prototypes are further refined by modeling correlations among P^c, F^s, and $\widetilde{F}^q$. This refinement process is formulated as:

$$P^* = \mathcal{R}(P^c, F^s, \widetilde{F}^q), \tag{14}$$

where $\mathcal{R}$ denotes the support-query correlation-based prototype refinement operation. The resulting P^* is used for subsequent query–prototype matching.

The refined prototypes P^* are matched with the semantic-enhanced query features $\tilde{F}^q$ by cosine similarity for point-wise prediction, while the model is optimized using the standard cross-entropy loss.

Table 1: Few-shot segmentation results (%) on S3DIS. Avg denotes the average mIoU over S0 and S1.

Method	Param.	2-way 1-shot			2-way 5-shot			3-way 1-shot			3-way 5-shot		
		S0	S1	Avg	S0	S1	Avg	S0	S1	Avg	S0	S1	Avg
DGCNN (Wang et al., 2019)	0.62M	36.34	38.79	37.57	56.49	56.99	56.74	30.05	32.19	31.12	46.88	47.57	47.23
ProtoNet (Snell et al., 2017)	0.27M	48.39	49.98	49.19	57.34	63.22	60.28	40.81	45.07	42.94	49.05	53.42	51.24
MPTI (Zhao et al., 2021b)	0.29M	52.27	51.48	51.88	58.93	60.56	59.75	44.27	46.92	45.60	51.74	48.57	50.16
AttMPTI (Zhao et al., 2021b)	0.37M	53.77	55.94	54.86	61.67	67.02	64.35	45.18	49.27	47.23	54.92	56.79	55.86
BFG (Mao et al., 2022)	–	55.60	55.98	55.79	63.71	66.62	65.17	46.18	48.36	47.27	55.05	57.80	56.43
2CBR (Zhu et al., 2023)	0.35M	55.89	61.99	58.94	63.55	67.51	65.53	46.51	53.91	50.21	55.51	58.07	56.79
PAP3D (He et al., 2023)	2.45M	59.45	66.08	62.76	65.40	70.30	67.85	48.99	56.57	52.78	61.27	60.81	61.04
Seg-PN (Zhu et al., 2024)	0.24M	64.84	67.98	66.41	67.63	71.48	69.36	60.12	63.22	61.67	62.58	64.53	63.56
TaylorSeg-PN (Wang et al., 2025a)	0.27M	67.12	71.11	69.12	70.44	72.23	71.34	60.28	65.70	63.00	62.78	67.06	64.92
EPSegFZ (Wang et al., 2026c)	2.02M	**73.08**	**73.75**	**73.42**	**75.90**	**76.11**	**76.01**	65.58	**66.27**	**65.93**	**68.30**	68.25	68.28
LDSGSeg	0.47M	72.61	71.66	72.14	71.91	73.34	72.63	**67.17**	64.46	65.82	67.46	**69.49**	**68.48**

Table 2: Few-shot segmentation results (%) on ScanNet. Avg denotes the average mIoU over S0 and S1.

Method	Param.	2-way 1-shot			2-way 5-shot			3-way 1-shot			3-way 5-shot		
		S0	S1	Avg	S0	S1	Avg	S0	S1	Avg	S0	S1	Avg
DGCNN (Wang et al., 2019)	1.43M	31.55	28.94	30.25	42.71	37.24	39.98	23.99	19.10	21.55	34.93	28.10	31.52
ProtoNet (Snell et al., 2017)	0.27M	33.92	30.95	32.44	45.34	42.01	43.68	28.47	26.13	27.30	37.36	34.98	36.17
MPTI (Zhao et al., 2021b)	0.29M	39.27	36.14	37.71	46.90	43.59	45.25	29.96	27.26	28.61	38.14	34.36	36.25
AttMPTI (Zhao et al., 2021b)	0.37M	42.55	40.83	41.69	54.00	50.32	52.16	35.23	30.72	32.98	46.74	40.80	43.77
BFG (Mao et al., 2022)	–	42.15	40.52	41.34	51.23	49.39	50.31	34.12	31.98	33.05	46.25	41.38	43.82
2CBR (Zhu et al., 2023)	0.35M	50.73	47.66	49.20	52.35	47.14	49.75	47.00	46.36	46.68	45.06	39.47	42.27
PAP3D (He et al., 2023)	2.45M	57.08	55.94	56.51	64.55	59.64	62.10	55.27	55.60	55.44	59.02	53.16	56.09
Seg-PN (Zhu et al., 2024)	0.24M	63.15	64.32	63.74	67.08	69.05	68.07	61.80	65.34	63.57	62.94	68.26	65.60
TaylorSeg-PN (Wang et al., 2025a)	0.27M	67.52	70.75	69.14	68.39	71.55	69.97	63.60	67.55	65.58	66.98	69.78	68.38
EPSegFZ (Wang et al., 2026c)	2.02M	**69.43**	68.25	68.84	**71.31**	69.97	70.64	**67.19**	66.86	**67.03**	**69.62**	69.83	69.73
LDSGSeg	0.47M	68.62	**71.28**	**69.95**	70.48	**72.04**	**71.26**	64.35	**68.02**	66.19	69.40	**70.81**	**70.11**

4 EXPERIMENTS

Detailed dataset protocols, evaluation settings, as well as implementation details are provided in Appendix A.

4.1 COMPARISON WITH EXISTING METHODS

We compare LDSGSeg with representative few-shot point cloud semantic segmentation methods, including DGCNN (Wang et al., 2019), ProtoNet (Snell et al., 2017), MPTI and AttMPTI (Zhao et al., 2021b), BFG (Mao et al., 2022), 2CBR (Zhu et al., 2023), PAP3D (He et al., 2023), Seg-PN (Zhu et al., 2024), TaylorSeg-PN (Wang et al., 2025a), and the recent EPSegFZ (Wang et al., 2026c). For comparability, we follow the standard evaluation protocol and dataset splits adopted in previous studies.

Results on S3DIS. As shown in Table 1, LDSGSeg consistently outperforms TaylorSeg-PN across all evaluated settings on S3DIS. Compared with the recent EPSegFZ, LDSGSeg achieves competitive performance with substantially fewer learnable parameters. In particular, LDSGSeg reaches comparable results in the 3-way settings despite using only 0.47M trainable parameters. These results demonstrate a favorable trade-off between segmentation performance and model complexity.

Results on ScanNet. Table 2 reports the results on ScanNet, which contains more categories and more diverse indoor scenes. LDSGSeg consistently improves over TaylorSeg-PN across all evaluated settings. Compared with EPSegFZ, LDSGSeg remains competitive across the evaluated ScanNet settings, while achieving better results in most configurations. These results indicate that LDSGSeg maintains strong performance under more diverse indoor scenes.

Table 3: Component analysis under 1-shot settings (%). LDE, PTCA, and PCC denote Local Distribution Encoding, Point-to-Text Cross-Attention, and Text-Guided Prototype Channel Calibration, respectively. Best results are shown in bold.

Components			S3DIS						ScanNet					
LDE	PTCA	PCC	2-way 1-shot			3-way 1-shot			2-way 1-shot			3-way 1-shot		
			S_0	S_1	Avg	S_0	S_1	Avg	S_0	S_1	Avg	S_0	S_1	Avg
×	×	×	67.12	71.11	69.12	60.28	**65.70**	63.00	67.52	70.75	69.14	63.60	67.55	65.58
✓	×	×	68.90	70.11	69.51	62.31	64.02	63.17	68.23	70.40	69.32	63.41	68.82	66.12
×	✓	×	68.56	70.24	69.40	64.27	63.43	63.85	68.10	69.51	68.81	63.26	68.76	66.01
×	×	✓	69.76	69.39	69.58	65.38	63.53	64.46	67.04	70.68	68.86	**65.01**	66.39	65.70
✓	×	✓	68.66	69.77	69.22	63.96	63.66	63.81	66.83	70.37	68.60	62.60	**69.53**	66.07
✓	✓	×	68.10	70.36	69.23	63.17	64.07	63.62	68.88	69.11	69.00	63.62	68.41	66.02
×	✓	✓	70.44	70.18	70.31	65.77	64.04	64.91	**69.36**	68.37	68.87	64.02	68.10	66.06
✓	✓	✓	**72.61**	**71.66**	**72.14**	**67.17**	64.46	**65.82**	68.62	**71.28**	**69.95**	64.35	68.02	**66.19**

Table 4: Effect of the LDE placement under 1-shot settings (%). PTCA and PCC are enabled in all variants. The dual-branch variant shares one stage-specific scale between LoConv and HiConv. Best results are shown in bold.

LDE Placement		S3DIS						ScanNet					
LoConv	HiConv	2-way 1-shot			3-way 1-shot			2-way 1-shot			3-way 1-shot		
		S_0	S_1	Avg	S_0	S_1	Avg	S_0	S_1	Avg	S_0	S_1	Avg
×	×	70.44	70.18	70.31	65.77	64.04	64.91	**69.36**	68.37	68.87	64.02	**68.10**	66.06
✓	×	70.06	69.14	69.60	66.19	60.76	63.48	67.56	70.03	68.80	63.27	67.59	65.43
✓	✓	71.19	69.67	70.43	67.01	63.25	65.13	68.21	70.73	69.47	64.09	67.71	65.90
×	✓	**72.61**	**71.66**	**72.14**	**67.17**	**64.46**	**65.82**	68.62	**71.28**	**69.95**	**64.35**	68.02	**66.19**

4.2 Ablation Experiments

We conduct ablation experiments to evaluate the proposed components, the placement of local distribution encoding, and the effect of category-level text semantics. All variants are trained independently under the same optimization and evaluation protocol. Since the two folds contain different evaluation categories, we use Avg as the primary measure while also reporting the fold-wise results.

4.2.1 Contribution of Individual Components

We first analyze the individual and combined effects of Local Distribution Encoding, Point-to-Text Cross-Attention, and Text-Guided Prototype Channel Calibration. As shown in Table 3, the proposed components exhibit complementary effects, with their joint configuration achieving the best overall performance across all evaluated settings. LDE introduces explicit local distribution information into neighborhood feature aggregation. Point-to-Text Cross-Attention incorporates foreground class semantics into query features, while Text-Guided Prototype Channel Calibration uses the corresponding class embeddings to recalibrate support-derived prototypes. These results demonstrate that local distribution information and category-level semantic guidance work effectively together within the proposed framework.

4.2.2 Placement of Local Distribution Encoding

We further examine the placement of LDE within the local feature extraction block. As shown in Table 4, incorporating LDE into HiConv achieves the best performance among the evaluated placements. This result suggests that the distribution descriptors are more effective when combined with the local responses produced by HiConv. Applying LDE only to LoConv or to both branches leads to smaller improvements, further supporting the choice of HiConv for distribution information integration.

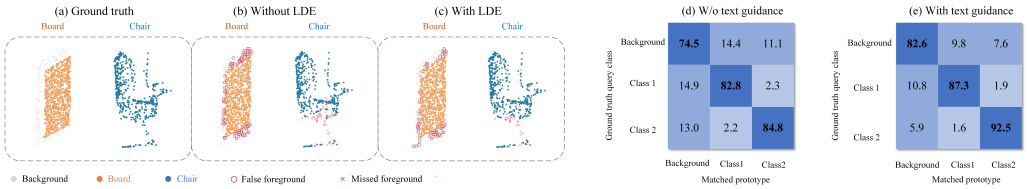

Figure 3: Qualitative analyses of LDE and text guidance. (a)–(c) compare the ground truth with foreground predictions without and with LDE, showing that LDE reduces false and missed foreground predictions. (d)–(e) compare query–prototype matching distributions without and with text guidance, where semantic guidance increases the correspondence between query points and their correct class prototypes.

Table 5: Effect of category-level text semantics under 1-shot settings (%). Each variant is trained independently while retaining the same semantic architecture. Best results are shown in bold.

Semantic Input	S3DIS						ScanNet					
	2-way 1-shot			3-way 1-shot			2-way 1-shot			3-way 1-shot		
	S_0	S_1	Avg	S_0	S_1	Avg	S_0	S_1	Avg	S_0	S_1	Avg
Zero Token	65.90	70.11	68.01	60.03	61.40	60.72	67.21	67.58	67.40	62.49	65.82	64.16
Random Token	68.94	68.35	68.65	60.50	62.07	61.29	67.85	68.26	68.06	62.57	66.49	64.53
Mismatched CLIP	66.23	66.49	66.36	58.23	60.97	59.60	65.31	66.48	65.90	62.81	64.92	63.87
Correct CLIP	**72.61**	**71.66**	**72.14**	**67.17**	**64.46**	**65.82**	**68.62**	**71.28**	**69.95**	**64.35**	**68.02**	**66.19**

4.2.3 Effect of Category-Level Text Semantics

Finally, we examine whether the performance gains are mainly associated with correctly aligned category-level semantics rather than the introduction of additional text tokens. To this end, we compare correct CLIP embeddings with fixed mismatched CLIP embeddings, zero tokens, and random tokens. As shown in Table 5, the correctly aligned CLIP embeddings consistently achieve the best performance, whereas the alternative semantic inputs lead to clear performance degradation. These results provide evidence that the semantic guidance benefits from the correspondence between class labels and their text representations, rather than from additional token inputs alone.

A computational complexity analysis is provided in Appendix A.

5 Conclusion

In this paper, we proposed LDSGSeg, a few-shot 3D point cloud semantic segmentation framework that integrates local distribution encoding with category-level semantic guidance. PCA-derived distribution information is incorporated into local feature extraction, while class-name embeddings guide query features and calibrate support-derived prototypes. Experiments on S3DIS and ScanNet demonstrate consistent improvements over TaylorSeg-PN, with competitive performance against EPSegFZ using substantially fewer learnable parameters. Ablation studies further verify the complementary effects of LDE, PTCA, and PCC. Future work will explore more efficient semantic guidance and extend LDSGSeg to generalized few-shot and open-vocabulary segmentation.

REFERENCES

Changshuo Wang, Zhijian Hu, Xiang Fang, Zai Yang Yu, Yibin Wu, Mingkun Xu, Yusong Wang, Xingyu Gao, and Prayag Tiwari. Biologically-Inspired Evolutionary Domain Symbiosis for Few-shot and Zero-shot Point Cloud Semantic Segmentation. In *Proc. AAAI Conf. Artif. Intell.*, volume 40, pp. 9666–9674, 2026a. doi: 10.1609/aaai.v40i12.37929.

Jie Zhou, Yong Zhao, and Fan Zhong. Prototype Optimization and Self-Training for Few-Shot 3D Point Cloud Semantic Segmentation. *Comput.-Aided Des.*, 190:103976, 2026. doi: 10.1016/j.cad.2025.103976.

Lili Wei, Congyan Lang, Zheming Xu, Liqian Liang, and Jun Liu. Few-Shot 3D Point Cloud Segmentation via Relation Consistency-Guided Heterogeneous Prototypes. *IEEE Trans. Multimedia*, 27:3158–3170, 2025. doi: 10.1109/TMM.2025.3557699.

Zhaochong An, Guolei Sun, Yun Liu, Runjia Li, Junlin Han, Ender Konukoglu, and Serge Belongie. Generalized Few-Shot 3D Point Cloud Segmentation with Vision-Language Model. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 16997–17007, 2025a.

Zhaoyang Li, Yuan Wang, Guoxin Xiong, Wangkai Li, Yuwen Pan, and Tianzhu Zhang. Generalized Few-Shot Point Cloud Segmentation via LLM-Assisted Hyper-Relation Matching. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 23063–23073, 2025.

Jingyuan Zhao, Wenyi Zhao, Bo Deng, Zhenghong Wang, Feng Zhang, Wenxiang Zheng, Wanke Cao, Jinrui Nan, Yubo Lian, and Andrew F. Burke. Autonomous driving system: A comprehensive survey. *Expert Systems with Applications*, 242:122836, 2024. doi: 10.1016/j.eswa.2023.122836.

Pranav Singh Chib and Pravendra Singh. Recent Advancements in End-to-End Autonomous Driving using Deep Learning: A Survey. *arXiv preprint arXiv:2307.04370*, 2023. URL `https://arxiv.org/abs/2307.04370`.

Mohsen Soori, Behrooz Arezoo, and Roza Dastres. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, 3:54–70, 2023. doi: 10.1016/j.cogr.2023.04.001.

Ruchi Goel and Pooja Gupta. Robotics and Industry 4.0. In *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development*, pp. 157–169. Springer, 2020. doi: 10.1007/978-3-030-14544-6_9.

Jeevan S. Devagiri, Sidike Paheding, Quamar Niyaz, Xiaoli Yang, and Samantha Smith. Augmented Reality and Artificial Intelligence in industry: Trends, tools, and future challenges. *Expert Systems with Applications*, 207:118002, 2022. doi: 10.1016/j.eswa.2022.118002.

Mickael Sereno, Xiyao Wang, Lonni Besançon, Michael J. McGuffin, and Tobias Isenberg. Collaborative Work in Augmented Reality: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 28(6):2530–2549, 2022. doi: 10.1109/TVCG.2020.3032761.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 652–660, 2017a.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5):1–12, 2019.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point Transformer. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 16259–16268, 2021a.

Limin Jiang, Changshuo Wang, Xin Ning, and Zaiyang Yu. LTTPoint: A MLP-Based Point Cloud Classification Method with Local Topology Transformation Module. In *2023 7th Asian Conference on Artificial Intelligence Technology (ACAIT)*, pp. 783–789, 2023. doi: 10.1109/ACAIT60137.2023.10528609.

Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3D point cloud semantic segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 8873–8882, 2021b.

Zhaoyang Li, Yuan Wang, Wangkai Li, Rui Sun, and Tianzhu Zhang. Localization and Expansion: A Decoupled Framework for Point Cloud Few-shot Semantic Segmentation. *arXiv preprint arXiv:2408.13752*, 2024. URL https://arxiv.org/abs/2408.13752.

Guoxin Xiong, Yuan Wang, Zhaoyang Li, Wenfei Yang, Tianzhu Zhang, Xu Zhou, Shifeng Zhang, and Yongdong Zhang. Aggregation and Purification: Dual Enhancement Network for Point Cloud Few-shot Segmentation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 1480–1488, 2024. doi: 10.24963/ijcai.2024/164.

Yongqiang Mao, Zonghao Guo, Xiaonan Lu, Zhiqiang Yuan, and Haowen Guo. Bidirectional feature globalization for few-shot semantic segmentation of 3D point cloud scenes. In *Proc. Int. Conf. 3D Vis.*, pp. 505–514, 2022.

Shuting He, Xudong Jiang, Wei Jiang, and Henghui Ding. Prototype adaption and projection for few- and zero-shot 3D point cloud semantic segmentation. *IEEE Trans. Image Process.*, 32:3199–3211, 2023.

Chao Zheng, Li Liu, Yu Meng, Xiaorui Peng, and Meijun Wang. Few-shot point cloud semantic segmentation via support-query feature interaction. *IEEE Trans. Circuits Syst. Video Technol.*, 34 (11):10753–10763, 2024. doi: 10.1109/TCSVT.2024.3417233.

Zhaochong An, Guolei Sun, Yun Liu, Fayao Liu, Zongwei Wu, Dan Wang, Luc Van Gool, and Serge Belongie. Rethinking few-shot 3D point cloud semantic segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024.

Guanyu Zhu, Yong Zhou, Rui Yao, and Hancheng Zhu. Cross-class bias rectification for point cloud few-shot segmentation. *IEEE Trans. Multimedia*, 25:9175–9188, 2023.

Lili Wei, Congyan Lang, Ziyi Chen, Tao Wang, Yidong Li, and Jun Liu. Generated and Pseudo Content guided Prototype Refinement for Few-shot Point Cloud Segmentation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 31103–31123, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/377d0752059d3d4686aa021b664a25dd-Abstract-Conference.html.

Changshuo Wang, Weijun Li, Shuting He, Xiang Fang, Xingyu Gao, Zhonghang Liu, Prayag Tiwari, and Dimitrios Kanoulas. From Coarse to Fine: Deep Prototype Refinement Network for Few-Shot Point Cloud Semantic Segmentation. OpenReview, 2026b. URL https://openreview.net/forum?id=EdzZDEjpQe.

Zhaochong An, Guolei Sun, Yun Liu, Runjia Li, Min Wu, Ming-Ming Cheng, Ender Konukoglu, and Serge Belongie. Multimodality helps few-shot 3D point cloud semantic segmentation. In *Proc. Int. Conf. Learn. Represent.*, 2025b.

Changshuo Wang, Shuting He, Xiang Fang, Meiqing Wu, Siew-Kei Lam, and Prayag Tiwari. Taylor series-inspired local structure fitting network for few-shot point cloud semantic segmentation. In *Proc. AAAI Conf. Artif. Intell.*, volume 39, pp. 7527–7535, 2025a.

Changshuo Wang, Xiang Fang, and Prayag Tiwari. DyPolySeg: Taylor Series-Inspired Dynamic Polynomial Fitting Network for Few-shot Point Cloud Semantic Segmentation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 62845–62856, 2025b. URL https://proceedings.mlr.press/v267/wang25aa.html.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proc. Adv. Neural Inf. Process. Syst.*, pp. 5099–5108, 2017b.

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and deformable convolution for point clouds. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 6411–6420, 2019.

Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified Transformer for 3D point cloud segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 8500–8509, 2022.

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 *Proc. Adv. Neural Inf. Process. Syst.*, 2024.

Canyu Zhang, Zhenyao Wu, Xinyi Wu, Ziyu Zhao, and Song Wang. Few-shot 3D point cloud semantic segmentation via stratified class-specific attention based Transformer network. In *Proc. AAAI Conf. Artif. Intell.*, pp. 3410–3417, 2023.

Jiahui Wang, Haiyue Zhu, Haoren Guo, Abdullah Al Mamun, Cheng Xiang, and Tong Heng Lee. EPSegFZ: Efficient Point Cloud Semantic Segmentation for Few- and Zero-Shot Scenarios with Language Guidance. In *Proc. AAAI Conf. Artif. Intell.*, 2026c.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proc. Adv. Neural Inf. Process. Syst.*, pp. 4077–4087, 2017.

Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Jiaming Liu, Han Xiao, Chaoyou Fu, Hao Dong, and Peng Gao. No time to train: Empowering non-parametric networks for few-shot 3D scene segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 3838–3847, 2024.

Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1534–1543, 2016.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5828–5839, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proc. Adv. Neural Inf. Process. Syst.*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. Int. Conf. Learn. Represent.*, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. Int. Conf. Mach. Learn.*, pp. 8748–8763, 2021.

A Experimental Details and Computational Complexity

A.1 Datasets and Evaluation Metrics

We evaluate LDSGSeg on two widely used indoor point cloud benchmarks, S3DIS (Armeni et al., 2016) and ScanNet (Dai et al., 2017). S3DIS contains 272 rooms from six areas, with 12 non-clutter categories used for evaluation, while ScanNet contains 1,513 reconstructed scenes with 20 annotated categories. Following the standard protocol (Zhao et al., 2021b; Wang et al., 2025a), we use preprocessed $1\,\mathrm{m} \times 1\,\mathrm{m}$ blocks, from which 2,048 points are sampled. The target categories of each dataset are partitioned into two non-overlapping folds, denoted by S_0 and S_1. When one fold is used for evaluation, the other provides the categories for episodic training.

Experiments are conducted under 2-way/3-way and 1-shot/5-shot settings. For each N-way class combination, 100 episodes are constructed. Each episode contains K annotated support blocks and one query block for every foreground class. Support and query blocks are sampled from different scenes. Following previous few-shot point cloud segmentation studies (Zhao et al., 2021b; Mao et al., 2022), we compute the IoU of each evaluation class over all relevant episodes and report foreground mIoU, excluding the background class. Results are reported separately on S_0 and S_1, and Avg denotes their arithmetic mean.

A.2 Implementation Details

All experiments are implemented in PyTorch (Paszke et al., 2019) on a single NVIDIA RTX 3090 GPU. Each training batch contains one few-shot episode. Each point is represented by 3D coordinates, RGB values, and normalized coordinates within the block. During training, we apply random rotation around the vertical axis, Gaussian coordinate jittering with a standard deviation of 0.01 clipped to 0.05, as well as random translation within $[-0.1, 0.1]$. No data augmentation is applied during evaluation. Models are trained for 24,000 iterations using AdamW (Loshchilov & Hutter, 2019), with an initial learning rate of 0.001 and a weight decay of 0.1. The learning rate is reduced by a factor of 0.5 every 7,000 iterations using StepLR. Gradients are clipped to a maximum norm of 1.0. Validation is performed every 2,000 iterations, with the checkpoint achieving the highest validation mIoU retained for evaluation.

The point encoder contains three hierarchical encoding stages, each equipped with an LDE Block for local feature extraction. The initial embedding dimension is 60, with 16 neighbors used for local aggregation and 10 neighbors used for feature propagation. The decoded point features are projected into a 128-dimensional feature space. For semantic guidance, we use a frozen CLIP ViT-B/32 text encoder (Radford et al., 2021) to extract 512-dimensional embeddings from class-name prompts in advance. These embeddings are projected into 128-dimensional point-feature space by a learnable linear layer. The Point-to-text Cross-Attention module employs four attention heads. All trainable components are jointly optimized during episodic training, while the CLIP text encoder remains frozen.

Table 6: Performance and computational complexity on S3DIS under the 2-way 1-shot setting on S_0. 3D pre-train denotes task-specific point-cloud backbone pre-training. LDSGSeg uses text embeddings extracted once by a frozen pre-trained CLIP encoder, and this offline extraction cost is excluded.

Method	mIoU	Param.	3D Pre-train Time	Episodic Train Time	Total Time
DGCNN	36.34	0.62M	4.0 h	0.8 h	4.8 h
AttMPTI	53.77	0.37M	4.0 h	5.5 h	9.5 h
2CBR	55.89	0.35M	6.0 h	0.2 h	6.2 h
PAP3D	59.45	2.45M	3.6 h	1.1 h	4.7 h
Seg-NN	49.45	0.00M	0.0 h	0.0 h	0.0 h
Seg-PN	64.84	0.24M	0.0 h	0.5 h	0.5 h
TaylorSeg-PN	67.12	0.27M	0.0 h	0.6 h	0.6 h
LDSGSeg	**72.61**	0.47M	0.0 h	0.8 h	0.8 h

A.3 COMPUTATIONAL COMPLEXITY

We further compare the computational complexity of LDSGSeg with representative few-shot point cloud segmentation methods. As shown in Table 6, LDSGSeg contains 0.47M learnable parameters, representing a noticeable increase over TaylorSeg-PN. Its episodic training time increases from 0.6 h to 0.8 h, while no task-specific 3D backbone pre-training is required. The CLIP text embeddings are extracted only once before episodic training and remain fixed thereafter, resulting in limited additional training overhead. Despite the increased model size, LDSGSeg achieves higher mIoU than TaylorSeg-PN, showing a favorable balance between segmentation performance and computational cost.