Reasoning Beyond Points: A Visual Introspective Approach for Few-Shot 3D Segmentation

Changshuo Wang
Department of Computer Science
University College London
London, United Kingdom
wangchangshuo1@gmail.com

Shuting He
School of Computing and Artificial Intelligence
Shanghai University of Finance and Economics
Shanghai, China
heshuting555@gmail.com

Xiang Fang*
Interdisciplinary Graduate Programme
Nanyang Technological University, Singapore
xfang9508@gmail.com

Zhijian Hu
LAAS
CNRS
Toulouse, France
huzhijian1991@gmail.com

Jia-Hong Huang
Information Institute
University of Amsterdam
Amsterdam, Netherlands
j.huang@uva.nl

Yixian Shen
Information Institute
University of Amsterdam
Amsterdam, Netherlands
y.shen@uva.nl

Prayag Tiwari
School of Information Technology
Halmstad University
Halmstad, Sweden
prayag.tiwari@ieee.org

Abstract

Point Cloud Few-Shot Semantic Segmentation (PC-FSS) aims to segment unknown categories in query samples using only a small number of annotated support samples. However, scene complexity and insufficient representation of local geometric structures pose significant challenges to PC-FSS. To address these issues, we propose a novel pre-training-free **V**isual **I**ntrospective **P**rototype **Seg**mentation network (**VIP-Seg**). Specifically, we design a Visual Introspective Prototype (VIP) module that employs a multi-step reasoning approach to tackle intra-class diversity and domain gaps between support and query sets. The VIP module consists of a Prototype Enhancement Module (PEM) and a Prototype Difference Module (PDM), which work alternately to progressively refine prototypes. The PEM enhances prototype discriminability and reduces intra-class diversity, while the PDM learns common representations from the differences between query and support features, effectively eliminating semantic inconsistencies caused by domain gaps. To further reduce intra-class diversity and enhance point discriminative ability, we propose a Dynamic Power Convolution (DyPowerConv) that leverages learnable power functions to effectively capture local geometric structures and detailed features of point clouds. Extensive experiments on S3DIS and ScanNet demonstrate that our proposed VIP-Seg significantly outperforms current state-of-the-art methods, proving its effectiveness in PC-FSS tasks. Our code will be available at https://github.com/changshuowang/VIP-Seg .

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

In recent years, point cloud data has become increasingly important in numerous applications such as autonomous driving [36, 4], robotics [21, 7], and augmented reality [6, 19]. As a fundamental task in 3D scene understanding, point cloud semantic segmentation [11] plays a crucial role in these domains. However, acquiring large-scale, high-quality annotated point cloud data demands substantial time and human resources, severely limiting the practical application of traditional deep learning methods.

To address the data scarcity challenge, researchers have turned to few-shot learning for point cloud segmentation tasks [13, 30]. Point Cloud Few-Shot Semantic Segmentation (PC-FSS) aims to segment novel categories in query samples using only a handful of annotated support samples, significantly reducing annotation costs. Zhao et al. [37] pioneered this approach by introducing AttMPTI, based on a pre-trained DGCNN [27]. Subsequent works [15, 10, 38] further enhanced feature extraction and prototype generation strategies, improving performance to some extent. However, PC-FSS faces two major challenges that limit its effectiveness: representation inconsistency within semantic categories and cross-domain feature misalignment. The first occurs when identical semantic categories in support and query samples show significant differences in physical characteristics, such as size, orientation, or visual appearance. Prototypes from support samples aid segmentation of similar objects in query samples but can introduce biases due to these variations. The second challenge involves feature distribution mismatches, where query data contains semantic content not present in the support set, and vice versa.

Moreover, most existing methods (as shown in Fig. 1) rely on pre-training paradigms, which not only increase computational costs but also potentially introduce domain shifts when facing unseen categories, particularly in cross-domain applications. Additionally, the irregular and sparse nature of point clouds makes it challenging to effectively capture local geometric structures, a problem that becomes more pronounced in few-shot scenarios where limited data is available for learning robust representations.

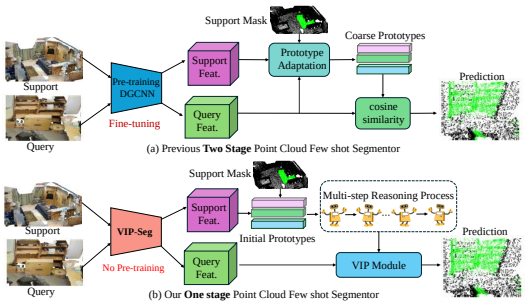

Figure 1: Comparison between previous methods and our approach for PC-FSS. (a) Previous methods typically follow a two-stage pipeline that requires pre-training a DGCNN followed by fine-tuning with prototype adaptation. (b) Our proposed VIP-Seg eliminates the pre-training stage with a single-stage approach that integrates DyPowerConv and employs a multi-step reasoning process to progressively refine prototypes.

To overcome these limitations, we propose VIP-Seg, a novel pre-training-free network for PC-FSS. As shown in Fig. 1, our approach features two key innovations: First, we introduce Dynamic Power Convolution (DyPowerConv), which adaptively models local geometric features by learning region-specific power functions, capturing fine-grained details and structural variations. This enhances the model's ability to distinguish similar structures, further reducing intra-class diversity. Second, we develop a Visual Introspective Prototype (VIP) module to address intra-semantic diversity and domain gaps through a multi-step reasoning approach. The VIP module combines a Prototype Enhancement Module (PEM) and a Prototype Difference Module (PDM) that work alternately to progressively refine prototypes. The PEM improves prototype discriminability through attention mechanisms, while the PDM learns common representations from feature differences, effectively mitigating domain gaps. This iterative process gradually aligns feature distributions and boosts segmentation accuracy.

Our main contributions can be summarized as follows:

- We propose VIP-Seg, a novel pre-training-free framework for point cloud few-shot semantic segmentation that achieves superior performance without time-consuming pre-training.
- We introduce Dynamic Power Convolution, which leverages learnable power functions to adaptively model complex local geometric features, significantly enhancing the network's ability to capture fine-grained structural details.

- We design a Visual Introspective Prototype module that employs a multi-step reasoning approach to effectively address intra-semantic diversity and domain gaps between support and query sets.
- Extensive experiments demonstrate that our approach significantly outperforms state-of-the-art methods across various few-shot settings, proving the effectiveness of our approach.

2 Related Works

2.1 Point Cloud Semantic Segmentation

Point cloud semantic segmentation [24, 29, 25] is a crucial task in 3D scene understanding that has witnessed significant advancements in recent years. Pioneering works such as PointNet [17] and PointNet++ [18] established the foundation by directly processing point cloud data through multi-layer perceptrons (MLPs). Subsequent research introduced innovative methods leveraging graph convolution, attention mechanisms, and multi-modal approaches. For instance, DGCNN [27] proposed the EdgeConv operation to capture inter-point relationships via dynamically constructed local graphs. Point Transformer [35] and its variants [11, 16] incorporated self-attention mechanisms to effectively model long-range dependencies. Recently, several methods based on State Space Models have achieved significant advancements in 3D tasks. Despite these advancements, these methods typically demand substantial annotated data for training, limiting their practical applications.

2.2 Point Cloud Few-shot Semantic Segmentation

To tackle the data scarcity challenge in point cloud semantic segmentation, few-shot learning approaches [20] have emerged as a promising solution. Early work by [37] introduced prototype networks to this domain through the AttMPTI method, sparking subsequent research focused on feature enhancement, prototype optimization, and domain adaptation [15, 34, 8, 32, 26, 23]. Recent advancements have developed more sophisticated techniques: [13] incorporated structural information for precise target localization while minimizing background interference, [30] addressed intra-class diversity and semantic inconsistency through bilateral aggregation and consistency purification, and [28] leveraged LLM-generated content to optimize prototypes and mitigate categorical bias. Meanwhile, [2] proposed a novel setting to avoid foreground leakage, while [1] enhanced performance through multi-modal data. Despite these advancements, challenges remain in effectively capturing complex local structures and addressing domain differences.

2.3 Dynamic Convolution

Dynamic convolution enhances a model's adaptability and expressive power by generating convolution kernels dynamically based on input data. In 2D image processing, dynamic convolution [33, 12] has been widely adopted for its effectiveness. Inspired by these successes, researchers have extended it to the 3D point cloud domain. [9] proposed DyCo3D, which incorporates dynamic context learning to better capture local point cloud features. [22] developed KPConv, which generates dynamic convolution kernels by learning local geometric structures, while [31] introduced PAConv, leveraging a weight bank and ScoreNet to dynamically assemble convolution kernels, thereby adapting to the irregular structure of point clouds. However, these methods primarily focus on combining multiple convolution kernels through attention coefficients, leaving significant room for improvement in fine-grained semantic understanding of local geometric structures.

3 Method

In this section, we first introduce the problem definition of Point Cloud Few-Shot Semantic Segmentation (PC-FSS). Then, we describe the overall architecture of VIP-Seg, as illustrated in Fig. 2. Next, we present the proposed Dynamic Power Convolution (DyPowerConv). Finally, we introduce our Visual Introspective Prototype (VIP) module that employs a multi-step reasoning approach.

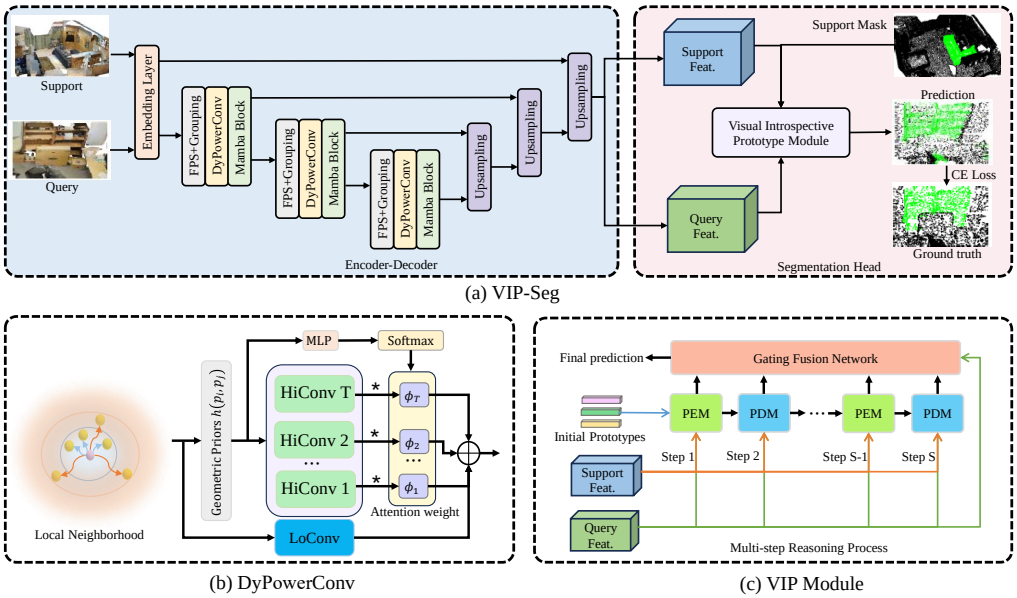

Figure 2: Overview of the proposed VIP-Seg. (a) The overall framework consists of an encoder-decoder backbone for feature extraction and a segmentation head for prototype-based classification. (b) Illustration of our Dynamic Power Convolution (DyPowerConv) module, which combines multiple high-order convolutions (HiConv) with dynamic attention weights to adaptively model local geometric structures. Each HiConv learns a different power function to capture fine-grained details. (c) The Visual Introspective Prototype (VIP) module implements a multi-step reasoning process where the Prototype Enhancement Module (PEM) and Prototype Difference Module (PDM) work alternately to progressively refine prototypes. A gating fusion network integrates predictions from multiple reasoning steps to generate the final segmentation result.

3.1 Problem Definition

Following the standard episodic learning paradigm [37], we partition the dataset into two non-overlapping class sets: base classes $\mathcal{C}_{base}$ used for training and novel classes $\mathcal{C}_{novel}$ used for testing, where $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$. The framework operates through N-way K-shot tasks, utilizing paired support and query sets. In each episode, the support set $\mathcal{S} = \{(P_s^{n,k}, M_s^{n,k})\}$ contains K labeled samples for each of the N categories, where $P_s^{n,k} \in \mathbb{R}^{L\times(3+d)}$ represents a point cloud with L points (each having 3D coordinates and d-dimensional features, such as color or surface normals), and $M_s^{n,k} \in \{0,1\}^L$ denotes the corresponding binary segmentation mask indicating foreground (target class) and background points. The query set $\mathcal{Q} = \{(P_q, M_q)\}$ contains point clouds $P_q \in \mathbb{R}^{L\times(3+d)}$ to be segmented, with ground truth masks $M_q \in \{0,1,2,...,N\}^L$ assigning each point to one of the N target classes or the background class.

3.2 Overall Architecture of VIP-Seg

As shown in Fig. 2, we propose VIP-Seg, formulating PC-FSS as a dual optimization problem that combines local structure modeling with progressive prototype refinement. VIP-Seg adopts an encoder-decoder architecture coupled with the proposed VIP module for effective few-shot segmentation. The encoder consists of three stacked DyPowerConv-Mamba blocks that integrate our proposed DyPowerConv for local geometric feature extraction and Mamba blocks [14] for capturing long-range dependencies. DyPowerConv adaptively models local point relationships through learnable power functions, while Mamba blocks efficiently process sequential data with state space models, enabling effective long-range interactions among points. The decoder progressively recovers point cloud resolution through inverse interpolation, propagating features from coarser to finer levels.

3.3 Dynamic Power Convolution

To effectively capture local geometric structures and detailed features of point clouds, thereby further reducing intra-class diversity, we propose Dynamic Power Convolution (DyPowerConv). This convolution adaptively models complex local geometric features through learnable power functions, enabling flexible representation of local structures.

3.3.1 Power Function Design

The design of our DyPowerConv is motivated by the need for flexible modeling of local geometric structures in point clouds. We employ learnable power functions to capture features at different scales and levels of detail. DyPowerConv comprises two key components: Low-order Convolution (LoConv) and Dynamic High-order Convolution (DyHiConv), expressed as:

$$g_i = g_i^L + g_i^{DH}, \tag{1}$$

where g_i^L and g_i^{DH} represent the outputs of LoConv and DyHiConv, respectively. LoConv captures basic geometric structures, while DyHiConv adaptively models fine-grained details through dynamic power functions.

3.3.2 Low-order Convolution

LoConv primarily extracts basic geometric information from local structures. We adopt Nonparameter Trigonometric Functions (NTF) to encode point cloud coordinates p_i and color information $c_i \in \mathbb{R}^3$, mapping them to the same dimension as their features, then adding their information and applying a non-linear transformation to obtain a high-dimensional representation of basic structural information. The LoConv can be formulated as:

$$g_i^L = \mathcal{A}(\{\mathbf{W}_l f_j' | p_j \in \mathcal{N}(p_i)\}), \tag{2}$$

$$f_j' = (f_j^p + f_j^c + f_j)/3, \tag{3}$$

$$f_j^p = [\sin(\alpha p_j / \beta^{\frac{6i}{d}}), \cos(\alpha p_j / \beta^{\frac{6i}{d}})]_{i=1}^d \in \mathbb{R}^d, \tag{4}$$

where f_j^c is obtained similarly to f_j^p. α and β represent the wavelength and amplitude hyperparameters of the trigonometric functions, respectively. $\mathbf{W}_l \in \mathbb{R}^{C_{in} \times C_{out}}$ denotes the non-linear transformation matrix.

3.3.3 Dynamic High-order Convolution

To capture the details of complex local geometric structures, DyHiConv draws inspiration from dynamic convolution [33] to generate multiple convolution weights using input information. Unlike traditional dynamic convolution, we improve the power function design by using a smooth power function $(|f_j - f_i| + \epsilon)^{p_t}$ instead of traditional sign functions, where p_t is a learnable parameter for each expert and ϵ is a small constant (typically 10^{-6}). This design effectively captures local geometric details while maintaining gradient continuity. DyHiConv (see Fig. 2(b)) can be expressed as:

$$g_i^{DH} = \sum_{t=1}^{T} \phi_t g_i^t, \tag{5}$$

where T is the number of experts (set to 8 in our implementation), g_i^t represents the high-order convolution of the t-th expert, and ϕ_t represents the attention assembly coefficient of the expert. Specifically, the high-order features g_i^t generated by each expert are calculated through:

$$g_i^t = \mathcal{A}(\{w_t(p_j) \odot (|f_j - f_i| + \epsilon)^{p_t} | p_j \in \mathcal{N}(p_i)\}), \tag{6}$$

where $w_t(p_j)$ is the dynamically generated weight, p_t is the learnable power exponent parameter of the t-th expert, $\odot$ represents element-wise multiplication, and $\mathcal{A}$ is an aggregation function (typically max pooling).

The attention assembly coefficients ϕ_t are constructed from explicit geometric information h_j:

$$\phi_t = \frac{\exp(\mathbf{W}_t h_j)}{\sum_{t=1}^{T} \exp(\mathbf{W}_t h_j)}, \tag{7}$$

where $\mathbf{W}_t$ is a learnable transformation matrix.

3.3.4 Explicit Structure Introduction

To better utilize the geometric information of point clouds, we use the coordinates of neighboring points p_j and center point p_i as basic geometric elements to construct the weight w_j for HiConv:

$$w_j = \mathbf{W}_h h_j, \tag{8}$$

where $h_j = [p_i, p_j, p_j - p_i, \|p_j - p_i\|] \in \mathbb{R}^{10}$, and $\mathbf{W}_h \in \mathbb{R}^{10 \times C_{out}}$ denotes the transformation matrix. The introduction of explicit geometric information facilitates the learning of relative spatial layout relationships between points and the capture of local geometric features and details.

3.4 Visual Introspective Prototype Module

To address feature discrepancies between support and query sets (e.g., intra-class diversity and domain gaps), we propose the Visual Introspective Prototype Module (VIP). It employs a multi-step reasoning process to iteratively refine prototypes, simulating human "think-reflect-revise" reasoning. VIP consists of two components: the Prototype Enhancement Module (PEM) and Prototype Difference Module (PDM), which alternately form a reasoning chain, progressively aligning feature distributions.

3.4.1 Prototype Enhancement Module

The PEM aims to enhance the discriminability of prototype features and reduce intra-class diversity through self-attention and cross-attention mechanisms. Given a point cloud with M points, let $\mathbf{F}_s \in \mathbb{R}^{M \times C}$ and $\mathbf{F}_q \in \mathbb{R}^{M \times C}$ denote the support and query features, respectively, where C is the feature dimension. The PEM first applies local max pooling and projection mapping to extract statistical characteristics of each channel:

$$\mathbf{F}'_s = \text{MaxPool}(\mathbf{F}_s) \cdot \mathbf{W}_1, \qquad \mathbf{F}'_q = \text{MaxPool}(\mathbf{F}_q) \cdot \mathbf{W}_1, \tag{9}$$

where $\mathbf{W}_1 \in \mathbb{R}^{C \times C}$ is a learnable transformation matrix. Next, the PEM enhances the prototype features from two aspects:

1) **Self-correlation Enhancement**: The PEM learns internal structural information by computing self-correlation matrices of the support and query features:

$$\mathbf{A}_s = \mathbf{W}_3(\mathbf{F}'^T_s \mathbf{F}'_s), \qquad \mathbf{A}_q = \mathbf{W}_3(\mathbf{F}'^T_q \mathbf{F}'_q). \tag{10}$$

$$\mathbf{F}^{self}_p = \text{Softmax}(\mathbf{A}_s)\mathbf{F}_p + \text{Softmax}(\mathbf{A}_q)\mathbf{F}_p, \tag{11}$$

2) **Cross-correlation Enhancement**: The PEM learns shared information through interaction between the support and query features:

$$\mathbf{A}_{cross} = \mathbf{F}'^T_q \mathbf{F}'_s, \tag{12}$$

$$\mathbf{F}^{cross}_p = \text{Softmax}(\mathbf{A}_{cross}) \odot \mathbf{F}_p, \tag{13}$$

Finally, the enhanced prototype features output by the PEM are:

$$\mathbf{F}^e_p = \mathbf{F}^{self}_p + \mathbf{F}^{cross}_p + \mathbf{F}_p, \tag{14}$$

where $\mathbf{F}_p \in \mathbb{R}^{(K+1) \times C}$ represents the initial prototype features, and $\mathbf{F}^e_p$ denotes the enhanced prototype features.

3.4.2 Prototype Difference Module

The PDM focuses on learning the differences between the support and query feature distributions to further eliminate domain gaps. After sharing similar pooling and mapping operations with the PEM, the PDM calculates the difference information between the support and query features:

$$\triangle_G = \mathbf{F}'^T_q \mathbf{F}'_q - \mathbf{F}'^T_s \mathbf{F}'_s, \tag{15}$$

and uses this difference information to adjust the prototype features:

$$\mathbf{F}^{delta}_p = \text{sigmoid}(\triangle_G) \odot \mathbf{F}^e_p. \tag{16}$$

Additionally, the PDM further optimizes the prototype features through cross-attention:

$$\mathbf{F}_p^{e\_cross} = \text{Softmax}(\mathbf{A}_{cross}) \odot \mathbf{F}_p^e, \tag{17}$$

where $\mathbf{A}_{cross}$ is the cross-correlation matrix between the support and query features. The final prototype features output by the PDM are:

$$\mathbf{F}_p^r = \mathbf{F}_p^{delta} + \mathbf{F}_p^{e\_cross} + \mathbf{F}_p^e, \tag{18}$$

where $\mathbf{F}_p^e$ is the output features from PEM, $\mathbf{F}_p^r$ is the output features after PDM processing.

3.4.3 Multi-step Reasoning Process

We design a multi-step reasoning process to progressively optimize the prototype features. Specifically, during S reasoning steps, the PEM and PDM work alternately:

$$\mathbf{F}_p^t = \begin{cases} \text{PEM}(\mathbf{F}_q, \mathbf{F}_s, \mathbf{F}_p^{t-1}), & \text{if } t\%2 = 0 \\ \text{PDM}(\mathbf{F}_q, \mathbf{F}_s, \mathbf{F}_p^{t-1}) + \mathbf{F}_p^{t-1}, & \text{if } t\%2 = 1 \end{cases} \tag{19}$$

where $\mathbf{F}_p^t$ represents the prototype features at step t, and $\mathbf{F}_p^0$ represents the initial prototype features. This multi-step reasoning process enables iterative refinement: the PEM step enhances discriminability by extracting key information, while the PDM step eliminates domain gaps through difference learning, progressively aligning feature distributions.

3.4.4 Gating Fusion Network

To effectively fuse the multi-step reasoning results, we design a gating fusion network. At each reasoning step t, we compute the cosine similarity between the query features $\mathbf{F}_q$ and the prototype features $\mathbf{F}_p^t$. The intermediate prediction results $\mathbf{L}_t$ are then computed as:

$$\mathbf{L}_t = \text{sim}(\mathbf{F}_q, \mathbf{F}_p^t) \cdot \mathbf{L}_p = \frac{\mathbf{F}_q \cdot \mathbf{F}_p^t}{\|\mathbf{F}_q\| \|\mathbf{F}_p^t\|} \cdot \mathbf{L}_p, \tag{20}$$

where $\text{sim}(\cdot, \cdot)$ denotes the cosine similarity function that measures the semantic alignment between query and prototype features, $\cdot$ denotes the dot product, and $\|\cdot\|$ represents the L2 norm. $\mathbf{L}_p$ represents the prototype labels.

Finally, the gating network learns importance weights $\mathbf{w}$ for predictions from each step and fuses all predictions to obtain the final result. The final prediction $\mathbf{L}_{final}$ is obtained by a weighted sum:

$$\mathbf{L}_{final} = \sum_{t=1}^{S} \mathbf{w}_t \cdot \mathbf{L}_t = \sum_{t=1}^{S} \mathbf{GFN}(\mathbf{F_q}) \cdot \mathbf{L}_t, \tag{21}$$

where $\mathbf{w} \in \mathbb{R}^S$ is the weight from the gating network $\mathbf{GFN}(\cdot)$, and $\mathbf{L}_{final}$ is the final prediction.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate VIP-Seg on two widely adopted 3D segmentation benchmarks.

S3DIS dataset [3] consists of RGB point clouds collected from 272 rooms across 6 indoor areas. Each point is annotated with one of 13 semantic labels (12 object categories and clutter). Following the common practice [37], we split each scene into 1m $\times$ 1m blocks and sample 2,048 points per block, resulting in a total of 7,547 blocks.

ScanNet dataset [5] contains 1,513 scanned indoor scenes with dense point-wise annotations over 20 semantic categories (excluding unannotated areas). Using the same preprocessing pipeline, we generate 36,350 blocks, each containing 2,048 points.

To evaluate the model in a few-shot setting, we split the categories of each dataset into two disjoint subsets, denoted as S_0 and S_1. When one subset is used as the test set, the other serves as the training set.

Evaluation Metric: We employ the mean Intersection-over-Union (mIoU), a standard metric for point cloud segmentation, to assess model performance.

Table 1: **Few-shot Results (%) on S3DIS.** S_i denotes the split i is used for testing. *Avg* is the average mIoU across splits. The best results are shown in **bold**.

Method	2-Way						3-Way					
	1-shot			5-shot			1-shot			5-shot		
	S_0	S_1	*Avg*	S_0	S_1	*Avg*	S_0	S_1	*Avg*	S_0	S_1	*Avg*
DGCNN [27]	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 [20]	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 [37]	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 [37]	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 [15]	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 [38]	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 [8]	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 [39]	64.84	67.98	66.41	67.63	71.48	69.56	59.11	60.42	59.77	59.48	64.72	62.10
TaylorSeg-PN [26]	67.12	71.11	69.12	70.44	72.23	71.34	60.28	65.70	63.00	62.78	67.06	64.33
DAFNet [25]	68.13	70.27	69.20	70.51	73.15	71.83	61.33	65.55	63.44	65.25	68.67	66.96
VIP-Seg	**73.50**	**74.92**	**74.21**	**73.84**	**76.88**	**75.36**	**65.54**	**69.92**	**67.73**	**72.93**	**71.44**	**72.19**
Improvement	+5.37	+4.65	+5.01	+3.33	+3.73	+3.53	+4.21	+4.37	+4.29	+7.68	+2.77	+5.23

Table 2: **Few-shot Results (%) on ScanNet.** S_i denotes the split i is used for testing. *Avg* is the average mIoU across splits. The best results are shown in **bold**.

Method	2-Way						3-Way					
	1-shot			5-shot			1-shot			5-shot		
	S_0	S_1	*Avg*	S_0	S_1	*Avg*	S_0	S_1	*Avg*	S_0	S_1	*Avg*
DGCNN [27]	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 [20]	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 [37]	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 [37]	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 [15]	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 [38]	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 [8]	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 [39]	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 [26]	67.52	70.75	69.14	68.39	71.55	69.97	63.60	67.55	65.58	66.98	69.78	68.38
DAFNet [25]	68.79	69.95	69.37	70.91	70.60	70.76	66.14	66.70	66.42	68.97	71.95	70.46
VIP-Seg	**71.94**	**72.67**	**72.31**	**70.95**	**73.48**	**72.22**	**68.91**	**69.19**	**69.05**	**73.22**	**72.74**	**72.98**
Improvement	+3.15	+2.72	+2.94	+0.04	+2.88	+1.46	+2.77	+2.49	+2.63	+4.25	+0.79	+2.52

4.2 Comparison with Existing Methods

Results analysis on the S3DIS dataset. As shown in Table 1, our VIP-Seg demonstrates superior performance on the S3DIS dataset. In the 2-way 1-shot setting, VIP-Seg achieves an average mIoU of 74.21%, surpassing the previous best method DAFNet [25] by 5.01 percentage points. In the 3-way 1-shot setting, VIP-Seg reaches an average mIoU of 67.73%, exceeding DAFNet by 4.29 percentage points. The most significant improvement is observed in the 3-way 5-shot setting, where VIP-Seg achieves 72.19% mIoU, outperforming DAFNet by 5.23 percentage points. These consistent gains across different settings validate the effectiveness and robustness of our approach. The improvements stem from our DyPowerConv's ability to capture local geometric features and the VIP module's effective prototype refinement through multi-step reasoning.

Results analysis on the ScanNet dataset. Our VIP-Seg also exhibits impressive performance on the more challenging ScanNet dataset, as illustrated in Table 2. In the 2-way 1-shot setting, VIP-Seg achieves an average mIoU of 72.31%, outperforming DAFNet [25] by 2.94 percentage points. In the 3-way 1-shot setting, VIP-Seg attains an average mIoU of 69.05%, surpassing DAFNet by 2.63 percentage points. Particularly noteworthy is the 3-way 5-shot setting, where VIP-Seg achieves an average mIoU of 72.98%, exceeding DAFNet by 2.52 percentage points. These significant improvements demonstrate VIP-Seg's ability to effectively handle complex indoor scenes with greater category diversity and varying levels of point cloud density. The multi-step reasoning mechanism proves particularly beneficial in distinguishing semantically similar objects in cluttered environments. The consistent performance gains across different settings further validate the generalization capability of our approach for PC-FSS.

Table 3: Ablation study on the key components of VIP-Seg.

LoConv	DyHiConv	Mamba	VIP	S_0	S_1	Avg
✓	✗	✗	✗	48.92	51.15	50.04
✗	✓	✗	✗	49.87	52.34	51.11
✓	✓	✗	✗	51.98	54.02	53.00
✓	✓	✓	✗	53.64	56.21	54.93
✓	✓	✗	✓	70.55	72.42	71.49
✓	✓	✓	✓	**73.50**	**74.92**	**74.21**

Table 4: Ablation study on different components of the VIP module.

PEM	PDM	2-way-1-shot			3-way-1-shot		
		S_0	S_1	Avg	S_0	S_1	Avg
✗	✗	53.64	56.21	54.93	48.37	51.86	50.12
✓	✗	70.16	72.73	71.45	69.43	72.45	70.94
✗	✓	71.54	73.26	72.40	70.66	73.01	71.84
✓	✓	**73.50**	**74.92**	**74.21**	**73.84**	**76.88**	**75.36**

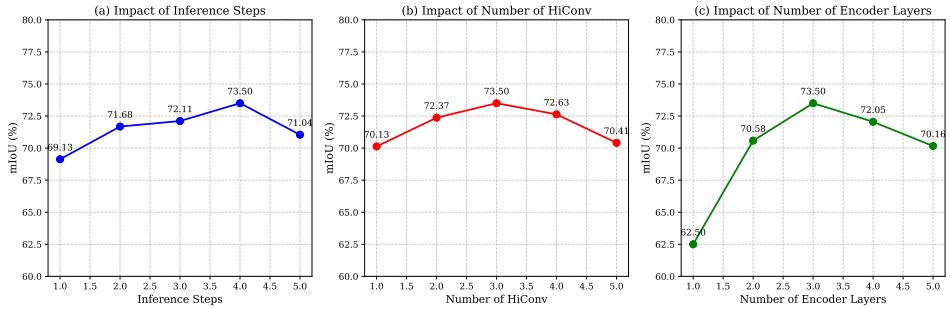

Figure 3: Parameter sensitivity analysis of VIP-Seg framework. (a) Impact of the number of reasoning steps in the VIP module on segmentation performance. (b) Impact of the number of HiConv experts in the DyPowerConv module. (c) Impact of encoder depth on model performance.

4.3 Ablation Experiments

All results are reported under 2-way-1-shot settings on the S_0 split of the S3DIS dataset.

4.3.1 Effectiveness of Different Components

Table 3 presents the ablation study on different components of our VIP-Seg. Using only LoConv or DyHiConv yields similar performance (50.04% and 51.11% mIoU), while their combination in the DyPowerConv module improves results to 53.00%. Adding Mamba blocks provides a modest gain of 1.93 percentage points. Most significantly, incorporating the VIP module leads to a substantial improvement, increasing mIoU from 53.00% to 71.49% without Mamba blocks. The complete architecture achieves the best performance of 74.21% mIoU, confirming that each component contributes to the overall framework, with the VIP module providing the most significant impact.

4.3.2 Analysis of VIP Module Components

Table 4 shows the ablation study on different components of our VIP module. Without either PEM or PDM (baseline), the model achieves 54.93% and 50.12% mIoU on 2-way and 3-way settings, respectively. Using only PEM significantly improves performance to 71.45% (+16.52%) for 2-way and 70.94% (+20.82%) for 3-way settings, demonstrating its effectiveness in enhancing prototype discriminability. Similarly, using only PDM yields substantial improvements to 72.40% (+17.47%) for 2-way and 71.84% (+21.72%) for 3-way settings, indicating its ability to eliminate domain gaps. The complete VIP module with both PEM and PDM achieves the best performance across all settings.

4.3.3 Hyperparameter Analysis

In Fig. 3(a), we analyze the effect of different reasoning steps in the VIP module. The performance gradually improves as the number of steps increases from 1 (69.13% mIoU) to 4 (73.50% mIoU), but decreases with 5 steps (71.04% mIoU). This pattern suggests that our multi-step reasoning approach effectively refines prototype features, but an optimal number of steps exists to balance refinement and avoid over-processing. Fig. 3(b) shows the impact of HiConv layers. Performance peaks at 3 layers (73.50% mIoU), suggesting an optimal balance between capacity and complexity for capturing local geometric features. Fig. 3(c) depicts the effect of encoder layers. The model achieves peak

Table 5: Impact of different geometric information in the explicit structure h_j on VIP-Seg performance.

Setting	2-way-1-shot			3-way-1-shot		
	S_0	S_1	Avg	S_0	S_1	Avg
$[p_j]$	70.66	72.03	71.35	70.22	72.12	71.17
$[p_i, p_j]$	71.67	73.15	72.41	71.36	74.07	72.72
$[p_i, p_j, p_j - p_i]$	72.97	74.12	73.55	72.13	75.15	73.64
$[p_i, p_j, p_j - p_i, \|p_i, p_j\|]$	**73.50**	**74.92**	**74.21**	**73.84**	**76.88**	**75.36**

Table 6: Performance and computational efficiency comparison.

Method	mIoU	Param.	Pre-train Time	Episodic Train	Total Time
DGCNN [27]	36.34	0.62 M	4.0 h	0.8 h	4.8 h
AttMPTI [37]	53.77	0.37 M	4.0 h	5.5 h	9.5 h
2CBR [38]	55.89	0.35 M	6.0 h	0.2 h	6.2 h
PAP3D [8]	59.45	2.45 M	3.6 h	1.1 h	4.7 h
Seg-PN [39]	64.84	0.24 M	0.0 h	0.5 h	0.5 h
VIP-Seg	**73.50**	2.77 M	0.0 h	1.25 h	1.25 h

performance with 3 layers (73.50% mIoU), significantly outperforming 1 layer (62.50% mIoU). Performance declines with 4 and 5 layers, likely due to overfitting or optimization challenges.

4.3.4 Impact of Explicit Geometric Structure

Table 5 examines the influence of different geometric information in the explicit structure h_j used in our DyPowerConv. Using only neighboring points $[p_j]$ achieves 71.35% and 71.17% mIoU on 2-way and 3-way settings, respectively. Adding center points $[p_i, p_j]$ improves performance by 1.06% and 1.55%, while further incorporating relative displacement $[p_j - p_i]$ brings additional gains of 1.14% and 0.92%. The complete representation that includes Euclidean distance $\|p_i - p_j\|$ achieves the best results across all settings.

4.4 Computational Complexity

Table 6 compares the computational efficiency of VIP-Seg with existing methods. Despite having 2.77M parameters, our approach eliminates pre-training, significantly reducing overall training time. Compared to pre-training methods like PAP3D[8] (4.7h) and 2CBR[38] (6.2h), VIP-Seg requires only 1.25h of training—a 73-80% reduction. When compared to Seg-PN[39], another pre-training-free method, VIP-Seg achieves an 8.66% higher mIoU with just 0.75h additional training time, demonstrating an effective balance between computational efficiency and performance.

5 Conclusion

In this paper, we propose VIP-Seg, a novel pre-training-free framework for point cloud few-shot semantic segmentation that effectively addresses the challenges of intra-class diversity and domain gaps. Our approach introduces two key innovations: the VIP module, which employs a multi-step reasoning process to progressively refine prototype features, and DyPowerConv, which adaptively models local geometric structures through learnable power functions. Extensive experiments on S3DIS and ScanNet datasets demonstrate that VIP-Seg significantly outperforms current state-of-the-art methods across various few-shot settings. **Limitations:** Despite its effectiveness, our approach has several limitations that warrant discussion. First, while VIP-Seg eliminates the need for pre-training, the episode-based training paradigm still requires substantial computational resources during the training phase. Second, the model's performance may degrade when dealing with extremely sparse point clouds or scenes with significant occlusions, as the local geometric structures become harder to capture. Third, the optimal number of reasoning steps in the VIP module is dataset-dependent and may require tuning for different application scenarios. **Future Work:** Several promising directions emerge from this work. First, incorporating multi-modal information (e.g., RGB images, depth maps) could enhance the model's robustness to sparse or occluded point clouds. Second, exploring the integration of large-scale pre-trained models in a parameter-efficient manner could potentially combine the benefits of pre-training with our efficient few-shot learning approach.

Acknowledgments

This work was supported in part by the European Union's Horizon 2024 Research and Innovation Programme for the Marie Skłodowska-Curie Actions under Grant No. 101211118. This work was also supported by the UKRI Future Leaders Fellowship [MR/V025333/1] (RoboHike). Shuting He was sponsored by Shanghai Pujiang Programme 24PJD030 and Natural Science Foundation of Shanghai 25ZR1402138.

References

[1] 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. *arXiv preprint arXiv:2410.22489*, 2024.

[2] 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 *CVPR*, pages 3996–4006, 2024.

[3] 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 *CVPR*, pages 1534–1543, 2016.

[4] Pranav Singh Chib and Pravendra Singh. Recent advancements in end-to-end autonomous driving using deep learning: A survey. *IEEE Transactions on Intelligent Vehicles*, 2023.

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[6] 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.

[7] Ruchi Goel and Pooja Gupta. Robotics and industry 4.0. *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development*, pages 157–169, 2020.

[8] Shuting He, Xudong Jiang, Wei Jiang, and Henghui Ding. Prototype adaption and projection for few-and zero-shot 3d point cloud semantic segmentation. *TIP*, 32:3199–3211, 2023.

[9] Tong He, Chunhua Shen, and Anton Van Den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *CVPR*, pages 354–363, 2021.

[10] Lvlong Lai, Jian Chen, Chi Zhang, Zehong Zhang, Guosheng Lin, and Qingyao Wu. Tackling background ambiguities in multi-class few-shot point cloud semantic segmentation. *Knowledge-Based Systems*, 253:109508, 2022.

[11] 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 *CVPR*, pages 8500–8509, 2022.

[12] Chao Li, Aojun Zhou, and Anbang Yao. Omni-dimensional dynamic convolution. *arXiv preprint arXiv:2209.07947*, 2022.

[13] 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.

[14] 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. *NeurIPS*, 37: 32653–32677, 2024.

[15] Yongqiang Mao, Zonghao Guo, LU Xiaonan, Zhiqiang Yuan, and Haowen Guo. Bidirectional feature globalization for few-shot semantic segmentation of 3d point cloud scenes. In *2022 International Conference on 3D Vision (3DV)*, pages 505–514. IEEE, 2022.

[16] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *CVPR*, pages 16949–16958, 2022.

[17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[18] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeuIPS*, 30, 2017.

[19] Mickael Sereno, Xiyao Wang, Lonni Besançon, Michael J Mcguffin, and Tobias Isenberg. Collaborative work in augmented reality: A survey. *TVCG*, 28(6):2530–2549, 2020.

[20] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 30, 2017.

[21] 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.

[22] 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 *ICCV*, pages 6411–6420, 2019.

[23] Changshuo Wang, Xiang Fang, and Prayag Tiwari. Dypolyseg: Taylor series-inspired dynamic polynomial fitting network for few-shot point cloud semantic segmentation. In *Forty-second International Conference on Machine Learning*, 2018.

[24] Changshuo Wang, Xin Ning, Linjun Sun, Liping Zhang, Weijun Li, and Xiao Bai. Learning discriminative features by covering local geometric space for point cloud analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022.

[25] Changshuo Wang, Shuting He, Xiang Fang, Jiawei Han, Zhonghang Liu, Xin Ning, Weijun Li, and Prayag Tiwari. Point clouds meets physics: Dynamic acoustic field fitting network for point cloud understanding. In *CVPR*, pages 22182–22192, 2025.

[26] 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 *AAAI*, volume 39, pages 7527–7535, 2025.

[27] 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 Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[28] 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. *NeurIPS*, 37: 31103–31123, 2025.

[29] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *NeuIPS*, 35:33330–33342, 2022.

[30] 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 *IJCAI*, 2024.

[31] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *CVPR*, pages 3173–3182, 2021.

[32] Yating Xu, Na Zhao, and Gim Hee Lee. Towards robust few-shot point cloud semantic segmentation. *arXiv preprint arXiv:2309.11228*, 2023.

[33] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *NeuIPS*, 32, 2019.

[34] 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 *AAAI*, volume 37, pages 3410–3417, 2023.

[35] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, pages 16259–16268, 2021.

[36] 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*, page 122836, 2023.

[37] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3d point cloud semantic segmentation. In *CVPR*, pages 8873–8882, 2021.

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

[39] 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 *CVPR*, pages 3838–3847, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions and scope are accurately reflected in the main claims made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of the work in the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We verified the effectiveness of our method through experiments.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: At this stage, I have not open access to the code, but the method provided in this article is easy to implement. If the paper is accepted by the conference, we will make the code publicly available immediately.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provided these informations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided sufficient information on the computer resources needed to reproduce the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discuss the potential impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We don't use the type of data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets used in this article are properly credited, and their licenses are clearly stated.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: At this stage the paper does not release new assets, if the paper is accepted, we will release our code, along with license and other documents.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.