Generalized Few-Shot Point Cloud Segmentation via LLM-Assisted Hyper-Relation Matching

Supplementary Material

Supplementary Material

This supplementary material provides a detailed explanation of three key components of our proposed LARM framework, including (A) multimodal distillation loss, (B) text-guided meta-feature selection, and (C) the process of generating textual descriptions using GPT. (D) provide extension experiments. Each section elaborates on the implementation details and the underlying reasoning.

A. Multimodal Information Distillation Training Process

To enhance the representational power of the visual backbone with textual information, we employ a multimodal distillation loss mechanism based on contrastive learning. The multimodal distillation process aligns the visual prototypes $\mathbf{P}_i$ with their corresponding textual features $\mathbf{t}_{i,j}$, ensuring that semantically similar visual and textual features are closer in the embedding space.

1. Text Feature Extraction

For a dataset with N categories, we generate M textual descriptions per category using GPT (details in Section C). These descriptions are encoded using CLIP's textual encoder [2] and a text adapter to produce text features:

$$\mathbb{T} = \{T_i\}_{i=1}^{N}, \quad \text{where } T_i = \{\mathbf{t}_{i,j}\}_{j=1}^{M}.$$

Here, $\mathbf{t}_{i,j} \in \mathbb{R}^d$ represents the textual feature of the j-th description for the i-th category.

2. Visual Prototypes

The visual prototypes $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^{N}$ are extracted from the visual backbone. Each prototype $\mathbf{p}_i \in \mathbb{R}^d$ represents the aggregated feature of the i-th category in the feature space.

3. Contrastive Loss and Segmentation Loss

The training process on base categories incorporates two key loss functions: the contrastive loss $\mathcal{L}_{\text{con}}$ and the segmentation loss $\mathcal{L}_{\text{seg}}$. These losses are combined to form the total training objective.

Contrastive Loss. The contrastive loss $\mathcal{L}_{\text{con}}$ aligns visual prototypes $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^{N}$ with their corresponding textual features $\mathbb{T} = \{T_i\}_{i=1}^{N}$ (details in Section A.1 and A.2). It ensures semantic consistency between visual and textual modalities while maintaining discriminative power across categories:

$$\mathcal{L}_{\text{con}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{M}\sum_{j=1}^{M}\log\frac{\exp\left(\text{sim}(\mathbf{p}_i, \mathbf{t}_{i,j})/\tau\right)}{\sum_{k=1}^{N}\sum_{l=1}^{M}\exp\left(\text{sim}(\mathbf{p}_i, \mathbf{t}_{k,l})/\tau\right)}\right], \tag{S1}$$

where $\text{sim}(\mathbf{p}, \mathbf{t}) = \frac{\mathbf{p}\cdot\mathbf{t}}{\|\mathbf{p}\|\|\mathbf{t}\|}$ is the cosine similarity, and $\tau > 0$ is a temperature parameter.

Segmentation Loss. The segmentation loss $\mathcal{L}_{\text{seg}}$ is a standard cross-entropy loss applied to the point-wise predictions of the model. It uses the ground-truth labels of base categories as supervision to train the feature extractor and base classifier. Formally, it is defined as:

$$\mathcal{L}_{\text{seg}} = -\frac{1}{|\mathcal{D}_{\text{base}}|}\sum_{k=1}^{|\mathcal{D}_{\text{base}}|}\sum_{i=1}^{l}\mathbf{M}_{k,i}^{b}\log\hat{\mathbf{M}}_{k,i}^{b}, \tag{S2}$$

where:

- $\mathcal{D}_{\text{base}} = \{(\mathbf{P}_k^b, \mathbf{M}_k^b)\}_{k=1}^{|\mathcal{D}_{\text{base}}|}$ is the base category training set.
- $\mathbf{P}_k^b \in \mathbb{R}^{l\times d}$ is the point cloud with l points and d features.
- $\mathbf{M}_k^b$ represents the ground-truth supervision for Base categories.
- $\hat{\mathbf{M}}_k^b$ is the model's predicted probability for each point belonging to the corresponding Base category.

Total Loss. The total loss function for training on the base categories is a weighted combination of the contrastive loss and segmentation loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg}} + \lambda\mathcal{L}_{\text{con}},$$

where $\lambda = 0.1$ is a weighting parameter balancing the two losses. This combination allows the model to learn both multimodal alignment (via $\mathcal{L}_{\text{con}}$) and accurate point-level segmentation (via $\mathcal{L}_{\text{seg}}$).

Supervised Training on Base Categories. The total loss $\mathcal{L}_{\text{total}}$ is used exclusively during the supervised training phase on base categories $\mathcal{C}^b$. This ensures that the backbone learns discriminative features for Base classes while also aligning the visual features with textual descriptions in a shared multimodal space. Specifically, the contrastive loss $\mathcal{L}_{\text{con}}$ enforces the semantic consistency between visual prototypes $\mathbf{P}$ and textual features $\mathbb{T}$ for Base categories, making their embeddings closer in the joint visual-textual feature space.

Impact on Novel Categories. Although the multimodal alignment is explicitly performed only on the base categories during training, this alignment benefits the novel categories as well. By aligning the visual and textual embeddings in a unified space, the model implicitly maps novel categories into the same multimodal space during fine-tuning. This is because the textual features for both base and novel categories are drawn from the same textual embedding space (e.g., CLIP's textual encoder), which is pretrained to capture general semantic relationships across a wide range of categories. When novel categories are fine-tuned using limited support samples, their visual features naturally align with their corresponding textual features in the shared multimodal space. This occurs because the visual backbone, already trained to align base category visual features with textual features, generalizes this alignment mechanism to novel categories due to the shared text embedding space and the transferable nature of the learned visual-textual alignment.

Conclusion. The multimodal alignment achieved during Base category training not only ensures discriminative and semantically aligned features for base classes but also provides a robust foundation for aligning visual and textual features of novel categories in the same multimodal space. This shared alignment mechanism effectively compensates for the scarcity of labeled Novel category samples and improves the model's generalization performance.

B. Text-Guided Meta-Feature Selection

The text-guided meta-feature selector leverages textual features to enhance novel category representation by extracting relevant information from base category prototypes.

Why Text-Guided Channel-Wise Attention Selects Relevant Meta-Features

The text-guided channel-wise attention mechanism is based on the premise that textual features, derived from a pre-trained language model (e.g., GPT), encode rich semantic information about categories. This semantic information can be leveraged to identify and extract meta-features from Base category prototypes that are most relevant to a given novel category. Further, texts are more controllable and accessible compared to visual features [4, 6], making them particularly suitable for guiding the attention mechanism to focus on relevant features.

1. Textual Features as Semantic Queries. Textual features T_{n_i} for a novel category n_i are vectorized representations that capture high-level semantic meaning. These features inherently encode relationships between categories in a shared textual embedding space. For instance, textual descriptions of "table" may share semantic similarities with "bed" or "chair" because they belong to similar semantic groups. By using T_{n_i} as a query in the attention mechanism, we guide the model to identify the most relevant information in the base category prototypes $\mathcal{P}_b$ that can contribute to constructing a meaningful representation for the novel category.

Importantly, texts can be conveniently processed by large pre-trained LLMs (e.g., GPT) to generate diverse and semantically rich representations [4, 6]. For example, a textual description like "a table with four legs and a flat surface" can be encoded into a high-dimensional vector that captures multiple semantic dimensions, such as shape, functionality, and structure. This multi-dimensional semantic representation enables the attention mechanism to explore base category prototypes $\mathcal{P}_b$ from various perspectives, identifying meta-features that are relevant and transferable to the novel category. Unlike visual data, which is often constrained by factors such as image quality, viewpoint, or occlusion, textual representations are inherently abstract and can encode a broader range of semantic relationships. By leveraging these diverse textual representations, the attention mechanism can better guide the selection of useful information from Base categories, ensuring that the extracted meta-features align with the semantic requirements of the novel category.

2. Why Channel-Wise Attention Works for Meta-Feature Selection. Channel-wise attention operates on the feature channels of base prototypes $\mathcal{P}_b$, where each channel corresponds to a dimension in the feature space learned by the visual backbone. In deep neural networks, feature channels often represent specific patterns or attributes [7] (e.g., texture, shape, color, or object parts). For example, one channel may encode "flat surfaces", while another may encode "four-legged structures". This phenomenon has been supported by prior work in the interpretability of convolutional neural networks [7], where feature channels were shown to correspond to disentangled semantic concepts, such as object shapes, textures, or specific parts of objects (e.g., table legs or tabletops).

By applying attention on the channel dimension, the model effectively selects the feature channels in $\mathcal{P}_b$ that are most relevant to the novel category's textual semantics. The mechanism can be understood as a soft feature selection process, where attention weights assign higher importance to channels that contribute to the novel category's representation, while suppressing irrelevant channels. Because text features are semantically rich and accessible, they provide a powerful signal to guide this selection process, ensuring that the selected meta-features are both meaningful and transferable.

C. Generating Textual Descriptions Using GPT

1. Prompt Design

To generate high-quality textual descriptions for each category, we adopt 3D-specific heuristic prompt templates inspired by [4, 8]. The templates include:

- *Caption Generation:* "Describe a point cloud of a [CLASS] in one sentence."
- *Question Answering:* "How to describe a point cloud of a [CLASS]?"
- *Paraphrase Generation:* "Generate a synonym: A point cloud of a [CLASS]."
- *Words to Sentence:* "Make a sentence with words: point cloud, [CLASS], obscure."

2. Text Generation Process

For a point cloud dataset with N categories, we replace the placeholder "`[CLASS]`" in the templates with each category name. Each template is sent as a prompt to GPT-3 [1], configured with a temperature of 0.7 to encourage diversity. GPT generates M unique descriptions per category, which are stored for further processing.

3. Text Encoding

The generated descriptions are encoded using CLIP's textual encoder [2] to produce textual features. A text adapter is further applied to align the textual features with the visual feature space, resulting in the final text features:

$$\mathbb{T} = \{T_i\}_{i=1}^{N}, \quad T_i = \{\mathbf{t}_{i,j}\}_{j=1}^{M}.$$

4. Integration with Multimodal Training

The textual features $\mathbb{T}$ are integrated into the multimodal distillation loss (Section A) and the text-guided meta-feature selector (Section B) to enhance the multimodal representation capabilities of the model.

5. GPT-Generated Examples

Figure S1 presents examples of diverse text descriptions generated by GPT, demonstrating its ability to effectively complement the original visual features. These generated descriptions are particularly beneficial for novel categories.

D. Extended Experiments

1. Effects of different LLMs.

We conducted additional experiments using Llama-2 to generate category descriptions under the same experimental settings on the S3DIS dataset (1-shot setting), as shown in Tab. S1. While minor performance differences are observed across LLMs, all variants of our method consistently outperform existing SOTA methods.

Table S1

Methods	mIoU-B	mIoU-N	mIoU-A	HM
GW [5]	74.10	29.66	53.58	41.92
PE [3]	74.54	39.78	58.50	51.34
Ours (LLaMA-2)	75.03	42.79	60.15	54.88
Ours(gpt-3.5)	**75.68**	**44.52**	**61.29**	**56.04**

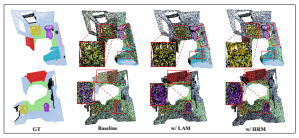
Table S3

Descriptions	mIoU-B	mIoU-N	mIoU-A	HM
Caption	75.06	43.28	60.39	54.90
QA	74.83	42.79	60.04	54.45
Paraphase	74.52	42.15	59.58	53.84
Word-to-Sectence	74.71	42.33	59.77	54.29
Combination	**75.68**	**44.52**	**61.29**	**56.04**

Table S2

Number of Agents (K)	mIoU-B	mIoU-N	mIoU-A	HM
2	75.22	43.93	60.78	55.44
4	**75.68**	**44.52**	**61.29**	**56.04**
6	75.53	44.46	61.19	55.96
8	75.16	44.30	60.92	55.73

Figure S2



2. Discussion of prompt engineering.

To further assess prompt sensitivity, we conducted a controlled experiment (see Tab. S3) using only one prompt type at a time, with the number of descriptions fixed at 6. This isolates the impact of each individual prompt design. Results show that although all prompt types contribute positively, the combination of diverse prompts consistently outperforms any single type, confirming the advantage of prompt diversity.

3. More ablation about the HRM.

Our additional ablation on HRM (Tab. S2) shows optimal performance at K=4, which indicates that a moderate number of reference agents balances diversity and avoids noise from over-partitioning.

3. More visualization results.

As GFS-3DSeg is inherently a challenging task, the overall performance is still behind fully supervised methods, which may result in some visual noise. Nonetheless, we provide additional qualitative results in Fig. S2 to better illustrate the effectiveness of each module.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[2] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1, 3

[3] Chih-Jung Tsai, Hwann-Tzong Chen, and Tyng-Luh Liu. Pseudo-embedding for generalized few-shot 3d segmentation. In *European Conference on Computer Vision*, pages 383–400. Springer, 2024. 3

[4] 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. *Advances in Neural Information Processing Systems*, 37:31103–31123, 2025. 2, 3

[5] Yating Xu, Conghui Hu, Na Zhao, and Gim Hee Lee. Generalized few-shot point cloud segmentation via geometric words.

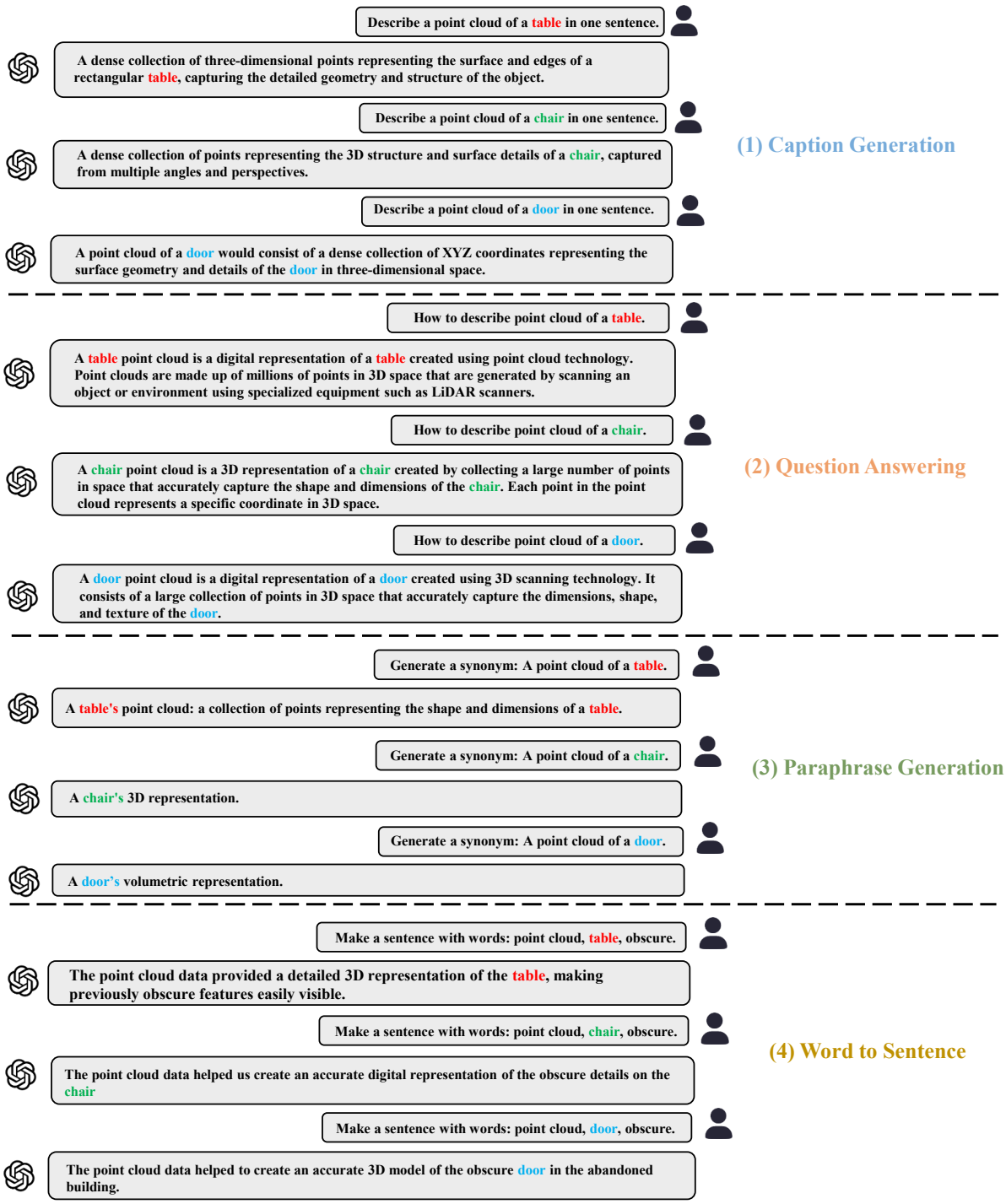


Figure S1. The descriptions examples generated by GPT.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21506–21515, 2023. 3

[6] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023. 2

[7] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014. 2

[8] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao

Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Point-clip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2639–2650, 2023. 3