Learning Hierarchical Orthogonal Prototypes for Generalized Few-Shot 3D Point Cloud Segmentation

Anonymous ICME submission

Abstract—**Generalized few-shot 3D point cloud segmentation aims to adapt to novel classes from only a few annotations while maintaining strong performance on base classes, but this remains challenging due to the inherent stability–plasticity trade-off: adapting to novel classes can interfere with shared representations and cause base-class forgetting. We present HOP3D, a unified framework that learns hierarchical orthogonal prototypes with an entropy-based few-shot regularizer to enable robust novel-class adaptation without degrading base-class performance. HOP3D introduces hierarchical orthogonalization that decouples base and novel learning at both the gradient and representation levels, effectively mitigating base–novel interference. To further enhance adaptation under sparse supervision, we incorporate an entropy-based regularizer that leverages predictive uncertainty to refine prototype learning and promote balanced predictions. Extensive experiments on ScanNet200 and ScanNet++ demonstrate that HOP3D consistently outperforms state-of-the-art baselines under both 1-shot and 5-shot settings. The code will be publicly released upon acceptance.**

Index Terms—**Few-shot learning, orthogonal regularization, prototype refinement**

I. Introduction

3D point cloud semantic segmentation assigns a label to each point in a 3D scene and underpins autonomous driving, robotics, and AR/VR applications [1]–[3]. With large-scale dense annotations, fully supervised models have achieved remarkable progress [4]–[7], yet high-quality 3D annotation is costly and difficult to scale [8], [9]. This motivates few-shot 3D segmentation, which adapts to novel categories from only a few labeled examples [10]–[13]. A more realistic setting is generalized few-shot 3D point cloud segmentation (GFS-3DS), where the model must recognize base classes with abundant supervision and novel classes with sparse annotations simultaneously [14], [15].

GFS-3DS is fundamentally constrained by the stability–plasticity dilemma: improving novel-class performance often degrades base-class knowledge [16]–[18]. This conflict is particularly acute for prototype-based formulations widely used in few-/generalized few-shot 3D segmentation [11], [14], [15], [19], [20]. First, base and novel categories share the same feature space and parameters, so few-shot updates for novel classes can directly perturb base decision boundaries. Second, segmentation is prototype-centric: predictions are governed by the relative geometry between point embeddings and class prototypes. Under sparse and biased support, novel prototypes are often noisy; updating them can warp the prototype subspace structure, making base–novel separation fragile and amplifying interference [2], [14], [15]. In short, reducing interference requires controlling both (i) the *few-shot adaptation dynamics* (*how to learn*) and (ii) the *prototype subspace structure* that shapes the decision geometry (*what to learn*).

Orthogonality provides a natural guiding principle for addressing both levels [21]–[25]. Prior work shows that orthogonal gradient projection can mitigate forgetting in continual learning [21] and that orthogonal prototypes improve base–novel separation in generalized few-shot *2D* segmentation [22]; related orthogonal-basis designs have also appeared in 3D class-incremental learning [23]. However, orthogonality must be enforced at *both* levels to be effective in GFS-3DS: gradient-space orthogonalization alone can stabilize updates but cannot prevent few-shot noise from geometrically warping the prototype subspace, while prototype-space orthogonality alone improves separability yet cannot stop novel adaptation from perturbing shared parameters and forgetting base knowledge. Therefore, a joint design that aligns projected updates with prototype subspace geometry is necessary for reliable base–novel joint recognition, but such a two-level orthogonality coupling remains underexplored in GFS-3DS.

Motivated by this, we propose **HOP3D**, a unified framework that instantiates orthogonality at both the optimization level and the representation level, together with an entropy-aware few-shot regularizer. Specifically, **HOP-Net** performs hierarchical orthogonalization via: (i) **HOP-Grad**, which projects novel gradients onto the orthogonal complement of base gradient directions to suppress harmful interference during Phase 2 adaptation; and (ii) **HOP-Rep**, which learns orthogonal prototype subspaces to induce a base/novel representation decomposition, improving separability while preserving base knowledge. To further improve robustness under extremely limited supervision, we introduce **HOP-Ent**, a dual-entropy regularizer (conditional-entropy minimization and marginal-entropy maximization) integrated into few-shot training to sharpen and balance novel predictions, avoiding extra test-time optimization [26]. Our main contributions are:

- From a unified view (*how vs. what to learn*) of GFS-3DS, we propose **HOP-Net**, which instantiates a joint orthogonality principle via gradient-space orthogonal projection (**HOP-Grad**) and prototype-space orthogonal decomposition (**HOP-Rep**) to mitigate base–novel interference.
- We introduce **HOP-Ent**, a dual-entropy regularizer integrated into Phase 2 training to improve prediction certainty and class balance.
- Extensive experiments on **ScanNet200** and **ScanNet++** demonstrate that **HOP3D** achieves state-of-the-art performance under both 1-shot and 5-shot settings.

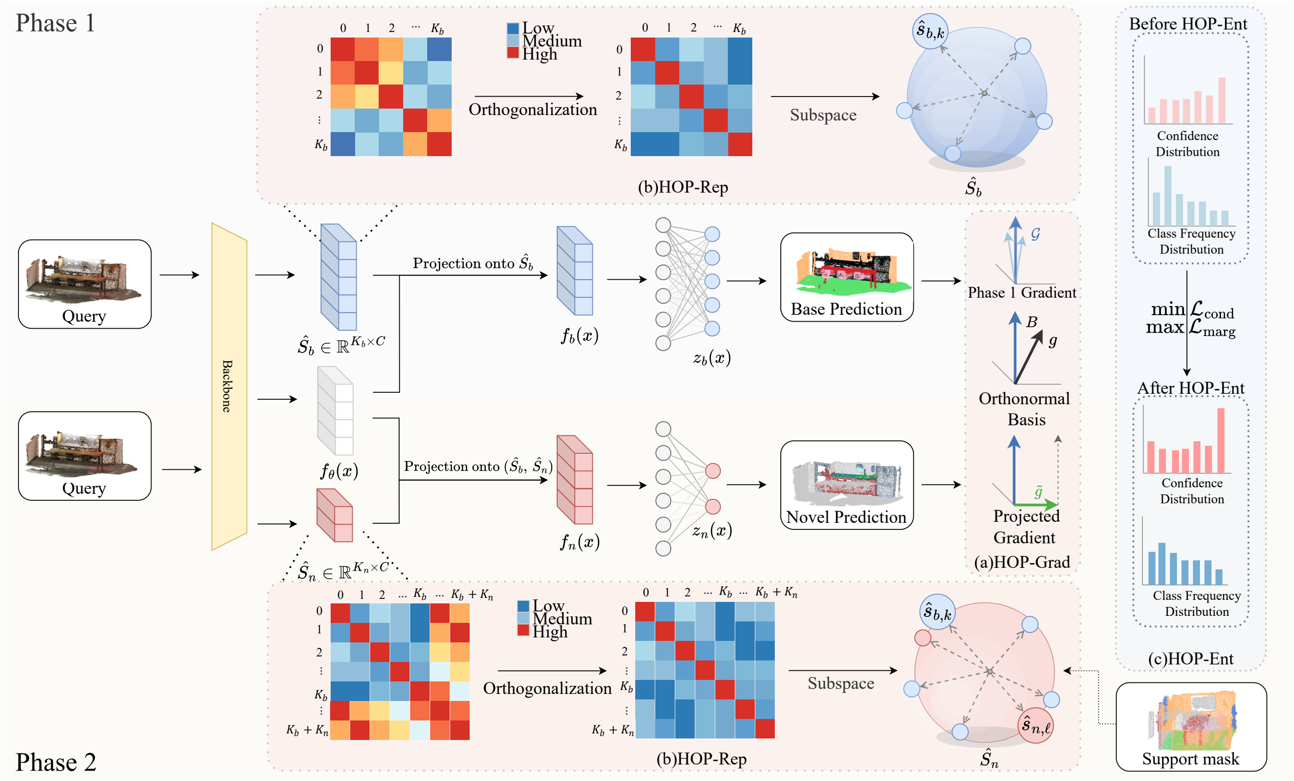

Fig. 1. Overview of the proposed HOP3D framework, integrating HOP-Net and HOP-Ent. The training pipeline consists of two phases: **Phase 1** trains base classes with **HOP-Rep** and collects base-task gradients $\mathcal{G}$ to construct an orthonormal basis B for **HOP-Grad**; **Phase 2** introduces novel classes for few-shot adaptation, applies **HOP-Grad** to project each novel gradient g as $\tilde{g}$ (removing B), continues to use **HOP-Rep** for prototype subspace orthogonalization, and integrates **HOP-Ent** for entropy-guided refinement. (a) Gradient orthogonalization in **HOP-Grad** using B and $\tilde{g}$; (b) Prototype similarity heatmaps and orthogonal prototype subspaces before/after **HOP-Rep**; (c) Confidence distribution and class-frequency distribution before/after **HOP-Ent**.

II. Methodology

A. Overview

We propose HOP3D, a unified framework for GFS-3DS that reduces base–novel interference and improves generalization under limited supervision. As shown in Fig. 1, HOP3D integrates two complementary components: (1) HOP-Net, which performs hierarchical orthogonalization, applying gradient-level orthogonal projection to avoid base-class forgetting and representation-level prototype orthogonalization to strengthen semantic separation; and (2) HOP-Ent, an entropy-based few-shot regularizer that encourages confident and balanced predictions. HOP3D is trained in two phases: base pretraining and novel adaptation. During the latter, both HOP-Net and HOP-Ent are activated to jointly enhance optimization stability and few-shot generalization.

B. Hierarchical Orthogonal Prototype Network (HOP-Net)

Orthogonal Gradient Projection (HOP-Grad): At the gradient level, HOP-Net incorporates an orthogonal projection module to stabilize few-shot adaptation while preserving base-class knowledge. Inspired by continual learning, this module prevents gradients from novel-class samples from updating directions already optimized for base classes by projecting them onto the orthogonal complement of the base gradient subspace. Let $\phi \in \mathbb{R}^d$ denote segmentation head parameters, including the class prototypes and the corresponding classifier weights, where d is the dimensionality of the vectorized parameter set.

Upon completing Phase 1 training, we extract a representative set of gradients $\mathcal{G} = \{g^{(t)}\}_{t=1}^{T}$ from the converged model by re-processing T mini-batches from the base training set, where T is chosen to balance gradient diversity and computational cost. Each $g^{(t)} \in \mathbb{R}^d$ denotes the gradient of the Phase 1 objective with respect to ϕ computed at the t-th extraction step. We then apply the Gram–Schmidt process to $\mathcal{G}$ to construct a compact orthonormal basis $B \in \mathbb{R}^{d \times r}$, where $r \leq T$ corresponds to the effective rank of the gradient set:

$$B = \mathrm{GS}(\{g^{(t)}\}_{t=1}^{T}), \quad \text{with} \quad B^\top B = I. \tag{1}$$

The basis B remains fixed throughout Phase 2, constraining gradient updates to lie within the orthogonal complement of the base optimization subspace.

During Phase 2, for any gradient $g \in \mathbb{R}^d$ with respect to ϕ that is induced by the novel-class supervision, we project it onto the orthogonal complement of the base subspace spanned by B as $\tilde{g} = g - B(B^\top g)$. Since B is orthonormal, $BB^\top g$ is the component of g within B, and $\tilde{g}$ retains only the component orthogonal to B. This removes update directions that overlap with the base optimization subspace, helping mitigate base-class forgetting.

The final projected gradient $\tilde{g}$ is then used to update parameters ϕ via the chosen optimizer. This gradient-level decoupling forms the first component of HOP-Net, ensuring stable adaptation by explicitly separating base and novel update directions.

Orthogonal Representation Decomposition (HOP-Rep): To disentangle representations, we enforce pairwise orthogonality on the parameterized projection bases, rather than raw feature representations. This ensures both intra-group separation and inter-group independence between base ($\hat{S}_b$) and novel ($\hat{S}_n$) subspaces, enabling sequential projections to resolve features into mutually decorrelated semantic components. Let $x \in \mathbb{R}^3$ denote a 3D point in the input cloud, and let $f_\theta(x) \in \mathbb{R}^C$ be its embedded feature produced by the backbone network $f_\theta(\cdot)$, where C is the feature dimension. Define the ℓ_2-normalized base prototype set for the K_b base classes as $\hat{S}_b = \{\hat{s}_{b,k} \in \mathbb{R}^C\}_{k=1}^{K_b}$.

In Phase 1, only base prototypes are used. The input feature is first projected onto the subspace spanned by base prototypes:

$$f_b(x) = \sum_{k=1}^{K_b} \langle f_\theta(x), \hat{s}_{b,k} \rangle \hat{s}_{b,k}, \tag{2}$$

$$r^{(0)}(x) = f_\theta(x) - f_b(x), \tag{3}$$

where $f_b(x) \in \mathbb{R}^C$ denotes the base-aligned component and $r^{(0)}(x) \in \mathbb{R}^C$ is the residual orthogonal to the base subspace. These two components are concatenated and passed through a shared multi-layer perceptron (MLP) to produce per-point classification scores for base classes.

In Phase 2, we introduce the novel prototype set $\hat{S}_n = \{\hat{s}_{n,\ell} \in \mathbb{R}^C\}_{\ell=1}^{K_n}$ for the K_n novel classes. The residual $r^{(0)}(x)$ is then projected onto the corresponding novel subspace:

$$f_n(x) = \sum_{\ell=1}^{K_n} \langle r^{(0)}(x), \hat{s}_{n,\ell} \rangle \hat{s}_{n,\ell}, \tag{4}$$

$$r^{(1)}(x) = r^{(0)}(x) - f_n(x), \tag{5}$$

where $f_n(x) \in \mathbb{R}^C$ is the novel-aligned projection and $r^{(1)}(x) \in \mathbb{R}^C$ is the remaining residual. The model employs two separate MLPs: $h_b(\cdot)$ for base features and $h_n(\cdot)$ for novel and residual features. The final prediction is obtained as:

$$z(x) = \Big[h_b(f_b(x)), \quad h_n(f_n(x), r^{(1)}(x)) \Big], \tag{6}$$

where $z(x) \in \mathbb{R}^{K_b+K_n}$ denotes the logits over all foreground categories. Although base prototypes in Phase 2 are initialized from those learned in Phase 1, their role is not strictly identical. In Phase 1, the residual space captures background information. In Phase 2, however, the introduction of novel prototypes redefines the residual structure, prompting further adaptation of the base prototypes to align with the new decomposition.

To encourage decorrelation among all learned prototypes, we apply a unified orthogonality regularizer to the cosine similarity between all distinct prototype pairs. In Phase 1, this regularizer is applied only to base prototypes, and in Phase 2 it is extended to the joint set of base and novel prototypes:

$$\mathcal{L}_{\text{orth}} = \sum_{i<j} \left| \hat{s}_i^\top \hat{s}_j \right|, \tag{7}$$

where the summation spans all distinct pairs of ℓ_2-normalized prototypes in the current phase.

C. Entropy-based Few-Shot Regularizer (HOP-Ent)

To enhance generalization on novel categories under limited supervision, we introduce the Entropy-based Few-Shot Regularizer (HOP-Ent). This module jointly optimizes two complementary entropy-based objectives to encourage both confident and balanced predictions. During Phase 2, we generate novel-class supervision by adopting the pseudo-label selection and adaptive infilling of GFS-VL [15] without any modification.

Let the model output logits for point x be $z(x) \in \mathbb{R}^{K_b+K_n}$, and define $p(y \mid x) = \text{softmax}(z(x))$ as the predicted class distribution, where K_b and K_n denote the number of base and novel classes. HOP-Ent consists of two loss components:

Conditional Entropy Minimization. For a selected set $\mathcal{S}$ of high-confidence pseudo-labeled points, we minimize the entropy of each prediction to improve per-sample certainty:

$$\mathcal{L}_{\text{cond}} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \left[- \sum_{c \in \mathcal{C}_n} p(c \mid x) \log p(c \mid x) \right], \tag{8}$$

where $\mathcal{C}_n$ is the set of novel class indices.

Marginal Entropy Maximization. To mitigate class imbalance among novel categories, we maximize the entropy of the batch-level marginal distribution:

$$\bar{p}(c) = \frac{1}{N} \sum_{i=1}^{N} p(c \mid x_i), \quad \mathcal{L}_{\text{marg}} = - \sum_{c \in \mathcal{C}_n} \bar{p}(c) \log \bar{p}(c), \tag{9}$$

where N is the total number of points in the batch.

Total Regularization Loss. The overall HOP-Ent loss is a weighted sum of the above terms:

$$\mathcal{L}_{\text{ent}} = \lambda_{\text{cond}} \, \mathcal{L}_{\text{cond}} + \lambda_{\text{marg}} \, \mathcal{L}_{\text{marg}}, \tag{10}$$

where λ_{cond} and λ_{marg} control the trade-off between the two objectives.

Unlike test-time adaptation techniques, HOP-Ent is integrated into Phase 2 training and jointly updates all model parameters, including the backbone, classifier heads, and prototypes. This end-to-end optimization encourages confident and diverse predictions, significantly enhancing robustness in generalized few-shot settings.

D. Training Objective

The overall training objective consists of two stages. In Phase 1, the model is trained on base classes using a segmentation loss and an orthogonality regularizer: $\mathcal{L}_{\text{P1}} = \mathcal{L}_{\text{seg}}^{\text{base}} + \lambda_{\text{orth}}^{(1)} \mathcal{L}_{\text{orth}}$. In Phase 2, the model adapts to novel classes using both labeled and pseudo-labeled data. The loss combines segmentation, dual entropy regularization, and continued orthogonality: $\mathcal{L}_{\text{P2}} = \mathcal{L}_{\text{seg}} + \mathcal{L}_{\text{ent}} + \lambda_{\text{orth}}^{(2)} \mathcal{L}_{\text{orth}}$. To prevent base forgetting, orthogonal gradient projection is applied during optimization: $\tilde{g} = g - B(B^\top g)$.

TABLE I

PERFORMANCE COMPARISON ON SCANNET200 AND SCANNET++. RESULTS UNDER 1-SHOT AND 5-SHOT SETTINGS.

Method	ScanNet200								ScanNet++							
	5-shot				1-shot				5-shot				1-shot			
	B	N	A	HM	B	N	A	HM	B	N	A	HM	B	N	A	HM
Fully Sup.	68.70	39.32	45.51	50.02	68.70	39.32	45.51	50.02	65.45	37.24	48.53	47.47	65.45	37.24	48.53	47.47
PIFS [27]	28.78	3.82	9.07	6.71	17.84	2.87	6.02	4.88	39.98	5.74	19.44	10.03	36.66	4.95	17.63	8.71
attMPTI [19]	37.13	4.99	11.76	8.79	54.84	3.28	14.14	6.17	55.89	4.19	24.87	7.78	53.16	3.55	23.40	6.66
COSeg [11]	57.67	5.21	16.25	9.54	47.03	4.03	13.09	7.42	59.34	6.96	27.91	12.45	58.49	6.24	27.14	11.26
GW [14]	59.28	8.30	19.03	14.55	55.23	6.47	16.74	11.56	51.35	11.03	27.16	18.15	46.71	6.63	22.66	11.59
GFS-VL [15]	67.17	31.18	38.76	42.59	67.25	28.89	36.97	40.42	60.49	21.40	37.04	31.61	60.02	17.90	34.75	27.56
HOP3D (ours)	**67.36**	**34.38**	**41.32**	**45.52**	**68.45**	**31.80**	**39.52**	**43.42**	**62.40**	**23.70**	**39.18**	**34.34**	**61.72**	**19.23**	**36.23**	**29.32**

III. EXPERIMENTS

A. Experimental Setup

Datasets. We evaluate our method on two large-scale benchmarks: **ScanNet200** [28], an extension of ScanNet [1] to 200 categories, and **ScanNet++** [29], which comprises 460 scenes across over 1,000 unique classes. Following the GFS-PCS protocol [15], our benchmark incorporates 57 classes for ScanNet200 and 30 for ScanNet++, with official train/test splits maintained; all unselected categories are treated as background during evaluation. Consistent with [30], raw points are voxelized with a 0.02 m grid size.

Evaluation Metrics. We evaluate performance using mean Intersection-over-Union (mIoU) on three category groups: base classes (mIoU-B), novel classes (mIoU-N), and all classes (mIoU-A). To better capture the balance between base and novel performance, we additionally report the harmonic mean (HM) of mIoU-B and mIoU-N. All results are averaged over five randomly sampled support sets for each evaluation setting.

Implementation Details. Following GFS-VL [15], we build our models on Point Transformer V3 (PTv3) [30] as backbone. Training follows a two-stage protocol. We first pretrain on base classes with AdamW (learning rate 6×10^{-3}). We then fine-tune in Phase 2 on 1-shot/5-shot support sets for 10% of the Phase 1 iterations, using a learning rate of 1×10^{-3} on ScanNet200 and 7×10^{-3} on ScanNet++. We use HOP-Rep in both phases, and enable HOP-Grad and HOP-Ent only during Phase 2 adaptation. The number of gradient samples T for HOP-Grad is set to 500 to ensure sufficient coverage of base optimization directions. A batch size of 8 is used. Experiments are run on 8 NVIDIA A100 GPUs.

B. Main Results

We compare HOP3D with representative GFS-3DS baselines, including PIFS [27], attMPTI [19], COSeg [11], and GW [14], as well as GFS-VL [15], the state-of-the-art baseline. For a fair comparison, we rerun GFS-VL under a unified setup. The results of the other baselines are quoted from the GFS-VL paper [15]. A fully supervised model trained with access to both base and novel labels is reported as the upper bound.

Quantitative results. Table I shows that HOP3D consistently outperforms the strongest baseline, GFS-VL, on ScanNet200/++ in both 1-shot and 5-shot settings. On ScanNet200, HOP3D achieves 34.38% mIoU-N and 45.52% HM in the 5-shot setting, outperforming GFS-VL by +3.20% and +2.93%. The substantial margins over COSeg (+29.17% mIoU-N and +35.98% HM) further highlight how hierarchical orthogonalization mitigates prototype collapse—a failure mode observed in prior prototype-refinement approaches. Under the 1-shot setting on ScanNet200, HOP3D achieves 31.80% mIoU-N and 43.42% HM, outperforming GFS-VL by 2.91% and 3.00%, respectively, while preserving 68.45% mIoU-B, which is nearly identical to the fully supervised performance (68.70%), thereby simultaneously improving novel-class performance and mitigating base-class forgetting under sparse supervision.

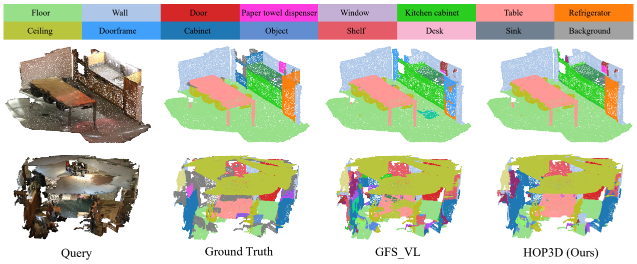

Fig. 2. Qualitative comparison between GFS-VL and our HOP3D on ScanNet200. Class color legend is shown at the top. From left to right: query input, ground-truth labels, GFS-VL prediction, and HOP3D prediction.

On ScanNet++, which features higher scene diversity and larger semantic space, HOP3D continues to maintain strong improvements. In the 5-shot setting, it achieves 23.70% mIoU-N and 34.34% HM, surpassing GFS-VL by 2.30% and 2.73%. The robust performance on this challenging benchmark suggests that HOP3D scales well with category richness and long-tail distributions. In the 1-shot scenario, HOP3D attains 19.23% mIoU-N and 29.32% HM, improving GFS-VL by 1.33% and 1.76%. This gain is attributed to the entropy-based refinement introduced by HOP-Ent, which mitigates prediction bias and promotes balanced novel-class learning. Overall, these results demonstrate that HOP3D not only enhances novel-class generalization but also preserves base-class performance by stabilizing the optimization landscape.

Qualitative results. Fig. 2 compares HOP3D with GFS-VL on ScanNet200. GFS-VL misclassifies novel objects (e.g., *refrigerator*) as base classes and distorts base-class predictions (e.g., *table* as *ceiling*), whereas HOP3D yields more consistent and accurate segmentation. These results indicate that HOP3D improves novel-class recognition while preserving base-class segmentation quality.

TABLE II
Ablation on ScanNet200 (1-shot) for HOP-Net and HOP-Ent.

HOP-Net	HOP-Ent	mIoU-B	mIoU-N	mIoU-A	HM
		67.64	28.80	36.98	40.39
✓†		68.67	29.46	37.71	41.23
✓‡		68.79	29.19	37.53	40.98
✓		**69.30**	30.52	38.69	42.37
✓	✓§	67.11	30.79	38.43	42.20
✓	✓	68.45	**31.80**	**39.52**	**43.42**

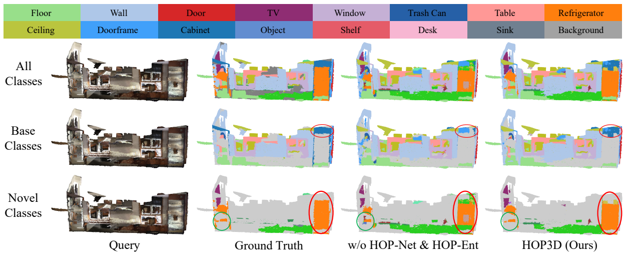

Fig. 3. Qualitative results illustrating the effect of HOP-Net and HOP-Ent. From left to right: query, ground truth, prediction without HOP-Net/HOP-Ent, and HOP3D. Circles highlight corrected base–novel misclassifications.

C. Ablation Study

Quantitative results. We evaluate HOP-Net and HOP-Ent on ScanNet200 under 1-shot, keeping the pseudo-labeling identical to [15]. As shown in Table II, HOP-Rep-only (†) and HOP-Grad-only (‡) improve mIoU-N/HM by 0.66%/0.84% and 0.39%/0.59%, respectively, reflecting complementary effects: prototype orthogonalization promotes semantic decoupling, while gradient projection stabilizes few-shot updates by removing base-conflicting directions. Combining them (full HOP-Net) further boosts mIoU-N/HM to 30.52%/42.37% (+1.72%/+1.98% over the baseline), indicating additive gains. We also report a marginal-entropy-only variant (§), which improves mIoU-N/HM by 1.99%/1.81% but is below full HOP-Net in HM, suggesting that balancing class frequency alone is insufficient. Adding full HOP-Ent yields the best trade-off, improving mIoU-N/HM by 3.00%/3.03% over the baseline with only a 0.85% mIoU-B drop vs. full HOP-Net.

Qualitative results. Fig. 3 shows that HOP3D corrects typical base–novel confusions (highlighted by circles) compared with the variant without HOP-Net and HOP-Ent.

Analysis of HOP-Net. HOP-Net introduces hierarchical orthogonalization at both the gradient and prototype levels. We analyze its behavior by varying the orthogonality weight λ_{orth} and the Phase 2 adaptation ratio (AR). As shown in Fig. 4(a)–(d), disabling orthogonalization ($\lambda_{\text{orth}} = 0$) hurts mIoU-N and HM, while a small weight improves the base–novel trade-off, with $\lambda_{\text{orth}} = 0.1$ giving the best overall balance; we therefore use $\lambda_{\text{orth}} = 0.1$ by default. Fig. 4(e) further shows that increasing AR consistently benefits both base and novel performance, with even minimal adaptation (0.625%) yielding clear gains. As shown in Fig. 5, Phase 2 adaptation without HOP-Net leads to a noticeable surge of off-diagonal prototype similarities, whereas HOP-Net preserves a more diagonal-dominant structure, indicating reduced inter-class redundancy and clearer subspace separation.

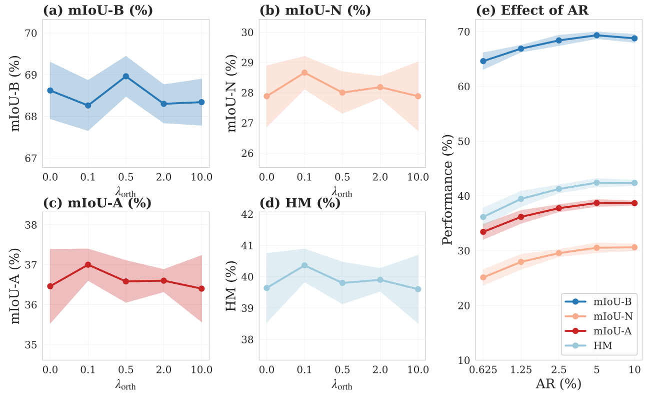

Fig. 4. HOP-Net ablation. (a)–(d) Impact of λ_{orth}. (e) Effect of AR. Shaded areas denote 95% confidence intervals.

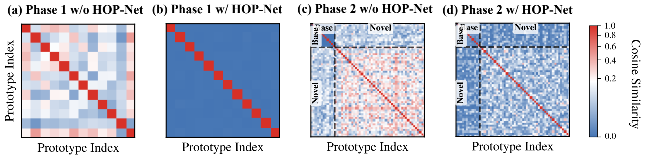

Fig. 5. Cosine-similarity matrices of ℓ_2-normalized prototypes (red: higher similarity, blue: lower). Phase 1: base prototypes only; Phase 2: joint base+novel prototypes (base first, then novel). (a) P1 w/o HOP-Net; (b) P1 w/ HOP-Net; (c) P2 w/o HOP-Net; (d) P2 w/ HOP-Net.

Analysis of HOP-Ent. HOP-Ent refines Phase 2 adaptation via conditional and marginal entropy. As shown in Fig. 6(a), conditional entropy minimization improves prediction certainty, increasing the mean confidence from 61.4% to 68.5% and raising the proportion of high-confidence predictions ($p > 0.9$) from 21.3% to 31.0%. As shown in Fig. 6(b), marginal entropy maximization improves class balance by reducing the standard deviation of novel-class prediction frequency from 0.9361% to 0.8662% and the coefficient of variation from 1.372 to 1.203. Together, these effects yield more reliable and balanced predictions, improving novel-class performance with minimal impact on base classes, consistent with Table II.

Efficiency. Measured on the same platform with AR = 10% (Fig. 4(e)), HOP3D incurs a 9.7% training-time overhead over GFS-VL [15]. Using a smaller AR further reduces the practical overhead, since HOP-Grad is only applied during Phase 2. Inference cost is unchanged, as HOP-Grad is applied exclusively during training.

Discussion and Limitations. The ablation results suggest that reducing base–novel interference in GFS-3DS requires jointly addressing optimization dynamics and representation geometry, rather than focusing on either aspect alone. HOP-Net stabilizes few-shot adaptation by constraining harmful update directions, while HOP-Ent further refines novel predictions through uncertainty-aware regularization, highlighting the importance of explicitly disentangling *how* and *what* to adapt under extremely limited supervision. Our method currently relies on the pseudo-labeling pipeline of [15] and constructs a fixed gradient basis after Phase 1; more robust pseudo-labeling strategies or adaptive gradient bases may further improve robustness. Finally, HOP-Grad introduces a small training overhead, while inference cost remains unchanged.

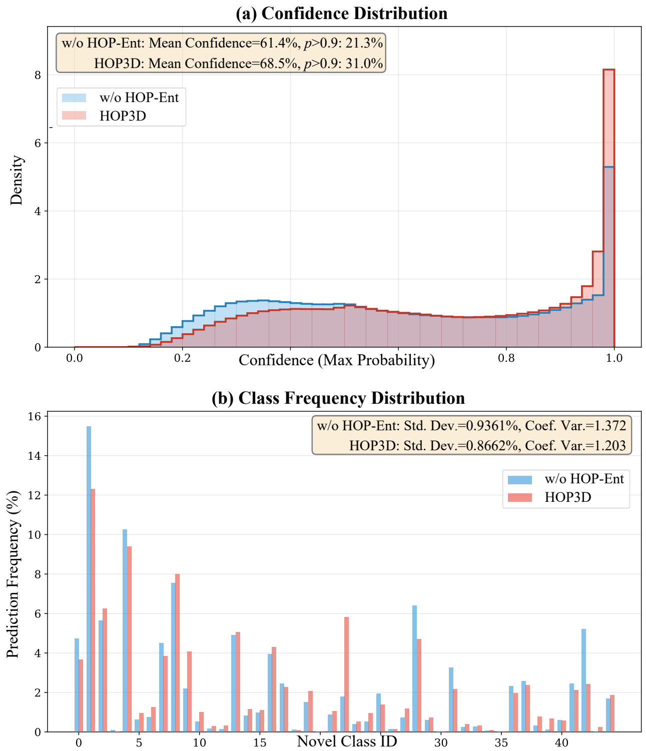

Fig. 6. HOP-Ent analysis. (a) Confidence distribution, where higher mean confidence indicates better prediction certainty. (b) Class frequency distribution, where a lower coefficient of variation indicates better class balance. HOP-Ent improves both prediction certainty and class balance.

IV. Conclusion

We presented HOP3D, a unified framework for generalized few-shot 3D segmentation that combines hierarchical orthogonal prototype learning with entropy-aware adaptation. By jointly decoupling optimization and representation (*how* vs. *what* to learn), HOP3D is, to the best of our knowledge, the first framework to introduce dual orthogonality into GFS-3DS, achieving a strong balance between base-class retention and novel-class generalization. HOP-Ent further improves confidence calibration and prediction balance without post-training or test-time procedures. Extensive experiments on ScanNet200 and ScanNet++ demonstrate state-of-the-art performance (1-shot and 5-shot settings). Future work will explore extensions to cross-modal and open-world 3D scene understanding.

References

[1] Angela Dai, Angel X. Chang, Manolis Savva, et al., “Scannet: Richly-annotated 3d reconstructions of indoor scenes,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[2] Charles R. Qi, Li Yi, Hao Su, et al., “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[3] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, et al., “Kpconv: Flexible and deformable convolution for point clouds,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019.

[4] Jonas Schult, Francis Engelmann, Alexander Hermans, et al., “Mask3d: Mask transformer for 3d semantic instance segmentation,” in *IEEE International Conference on Robotics and Automation*, 2023.

[5] Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, et al., “Oneformer3d: One transformer for unified point cloud segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

[6] Xin Lai, Jianhui Liu, Li Jiang, et al., “Stratified transformer for 3d point cloud segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.

[7] Ting Han, Yiping Chen, Jin Ma, et al., “Point cloud semantic segmentation with adaptive spatial structure graph transformer,” *International Journal of Applied Earth Observation and Geoinformation*, 2024.

[8] Yuliang Sun, Xudong Zhang, and Yongwei Miao, “A review of point cloud segmentation for understanding 3d indoor scenes,” *Visual Intelligence*, 2024.

[9] Jingyi Wang, Yu Liu, Hanlin Tan, et al., “A survey on weakly supervised 3d point cloud semantic segmentation,” *IET Computer Vision*, 2024.

[10] Shuting He, Xudong Jiang, Wei Jiang, et al., “Prototype adaption and projection for few- and zero-shot 3d point cloud semantic segmentation,” *IEEE Trans. Image Process.*, vol. 32, pp. 3199–3211, 2023.

[11] Zhaochong An, Guolei Sun, Yun Liu, et al., “Rethinking few-shot 3d point cloud semantic segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

[12] S. Xu, L. Zhang, G. Jiang, et al., “Part-whole relational few-shot 3d point cloud semantic segmentation,” *Computers, Materials & Continua*, 2024.

[13] Zhaochong An, Guolei Sun, Yun Liu, et al., “Multimodality helps few-shot 3d point cloud semantic segmentation,” in *Int. Conf. Learn. Represent. (ICLR)*, 2025.

[14] Yating Xu, Conghui Hu, Na Zhao, et al., “Generalized few-shot point cloud segmentation via geometric words,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2023.

[15] Zhaochong An, Guolei Sun, Yun Liu, et al., “Generalized few-shot 3d point cloud segmentation with vision-language model,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025.

[16] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, et al., “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the National Academy of Sciences*, 2017.

[17] German I. Parisi, Ronald Kemker, Jose L. Part, et al., “Continual lifelong learning with neural networks: A review,” *Neural Networks*, 2019.

[18] Sanghwan Kim, Lorenzo Noci, Antonio Orvieto, et al., “Achieving a better stability–plasticity trade-off via auxiliary networks in continual learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[19] 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. (CVPR)*, 2021.

[20] Shuqian Yang, Henhui Ding, and Xudong Jiang, “Generalized few-shot 3d point cloud segmentation,” in *IEEE International Symposium on Circuits and Systems*, 2024.

[21] Mehrdad Farajtabar, Navid Azizan, Alex Mott, et al., “Orthogonal gradient descent for continual learning,” in *International Conference on Artificial Intelligence and Statistics*, 2020.

[22] Sun-Ao Liu, Yiheng Zhang, Zhaofan Qiu, et al., “Learning orthogonal prototypes for generalized few-shot semantic segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[23] Townim Chowdhury, Ali Cheraghian, Sameera Ramasinghe, et al., “Few-shot class-incremental learning for 3d point cloud objects,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022.

[24] Naman Bansal, Xiaohan Chen, Zhangyang Wang, et al., “Can we gain more from orthogonality regularizations in training deep cnns?,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018.

[25] Michael Cogswell, Faruk Ahmed, Ross Girshick, et al., “Reducing overfitting in deep networks by decorrelating representations,” in *Int. Conf. Learn. Represent. (ICLR)*, 2016.

[26] Sina Hajimiri, Malik Boudiaf, Ismail Ben Ayed, et al., “A strong baseline for generalized few-shot semantic segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

[27] Fabio Cermelli, Massimiliano Mancini, Yongqin Xian, et al., “Prototype-based incremental few-shot segmentation,” in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2021.

[28] David Rozenberszki, Or Litany, and Angela Dai, “Language-grounded indoor 3d semantic segmentation in the wild,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022.

[29] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nissner, et al., “Scannet++: A high-fidelity dataset of 3d indoor scenes,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2023.

[30] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, et al., “Point transformer v3: Simpler, faster, stronger,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.