Point-PRC: A Prompt Learning Based Regulation Framework for Generalizable Point Cloud Analysis

Hongyu Sun[1,2] **Qiuhong Ke**[2] **Yongcai Wang**[1]*
Wang Chen[1] **Kang Yang**[1] **Deying Li**[1] **Jianfei Cai**[2]

[1]Department of Computer Science, Renmin University of China, China
[2]Department of Data Science & AI, Monash University, Australia

Abstract

This paper investigates the 3D domain generalization (3DDG) ability of large 3D models based on prevalent prompt learning. Recent works demonstrate the performances of 3D point cloud recognition can be boosted remarkably by parameter-efficient prompt tuning. However, we observe that the improvement on downstream tasks comes at the expense of a severe drop in 3D domain generalization. To resolve this challenge, we present a comprehensive regulation framework that allows the learnable prompts to actively interact with the well-learned general knowledge in large 3D models to maintain good generalization. Specifically, the proposed framework imposes multiple explicit constraints on the prompt learning trajectory by maximizing the mutual agreement between task-specific predictions and task-agnostic knowledge. We design the regulation framework as a plug-and-play module to embed into existing representative large 3D models. Surprisingly, our method not only realizes consistently increasing generalization ability but also enhances task-specific 3D recognition performances across various 3DDG benchmarks by a clear margin. Considering the lack of study and evaluation on 3DDG, we also create three new benchmarks, namely base-to-new, cross-dataset and few-shot generalization benchmarks, to enrich the field and inspire future research. Code and benchmarks are available at https://github.com/auniquesun/Point-PRC.

1 Introduction

3D point cloud data is widely adopted in many industrial and civil areas, such as autonomous driving [47], robotics [27, 3], geospatial mapping [39] and entertainment games [46]. Recognizing 3D objects from point cloud data is a basic need of these applications. Relevant research topics have been explored for a long time and their development can be summarized in three stages. In the early phase, PointNet series [48, 49] sparked a wave of directly operating raw point cloud data using deep learning techniques. Later methods improved upon PointNet and PointNet++ in terms of local information aggregation [33, 67, 60, 63, 69, 37], optimization techniques [50], geometry prior [55], model architecture [18, 82, 68, 44, 14], *etc*. Although remarkable progress has been made, these works tend to design specific architectures targeting downstream benchmarks while paying little attention to the model generalization, resulting in disappointed performances when deploying in the wild, especially in the case of unseen domains and corrupted data. On the other hand, training point cloud recognition models on each benchmark is not always feasible due to the narrow set of 3D visual concepts and expensive labeled data.

*Corresponding author. {sunhongyu, ycw}@ruc.edu.cn, {qiuhong.ke, jianfei.cai}@monash.edu

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

The above factors call for the investigation of the domain generalization routes for the deep point cloud models so that they can learn robust and transferable representations. Related studies have been extensively conducted in image recognition [28, 29, 31, 30, 32, 87, 13, 84] while to our best knowledge, there are only a few methods to discuss the domain adaptation and domain generalization in 3D. Several years ago, PointDAN [51] first investigated domain adaptation for point cloud classification models by aligning multi-scale features of 3D objects across the source and target domains. MetaSets [20] proposed to meta-learn on a group of transformed point sets to obtain generalizable representations to handle the sim-to-real geometry shifts. PDG [64] decomposed 3D objects into shared part space to reduce domain gap and developed a part-level domain generalization model for 3D point cloud classification.

However, the above methods are all built on small models (e.g., PointNet with 1.2M parameters) and small datasets (e.g., ModelNet with 9,843 training samples) and the overall transferability is still suppressed compared to prevalent large 3D foundation models [76, 90, 71, 78, 77, 79, 72], which have been pre-trained on numerous volume of 3D data [11, 10] and demonstrated promising zero-shot capability. Recent works stand on the shoulder of large 3D foundation models and push the boundary of downstream 3d tasks by parameter-efficient adaptation, such as prompt learning [74, 58], adapter [59, 88], and their combination. They insert learnable prompts in the inputs or adapter inside the Transformer [62] blocks to adapt the foundation models to specific 3D tasks. However, optimizing the newly introduced small modules targeting downstream benchmarks is prone to overfitting, thus disturbing the internal representations and compromising the inherent generalization of the foundation models [85, 36, 89, 26, 24, 25]. As Fig. 1 demonstrates, lightweight prompt tuning can notably lift the recognition accuracy of representative large 3D models on seen classes while hindering the generalization on unseen new classes, where the performances consistently lag behind corresponding zero-shot predictions of these models.

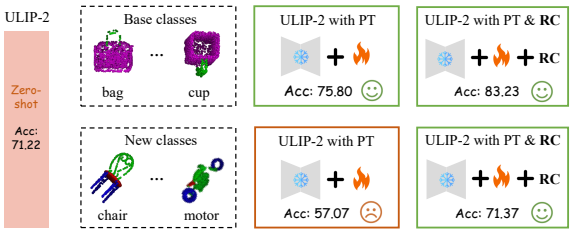

Figure 1: **Motivation of our research: to promote the performances on downstream 3D tasks while maintaining good generalization of large 3D models.** The experiments are conducted on ShapeNetCoreV2. ULIP-2 can reach 71.22% zero-shot recognition accuracy on this dataset. Recent works built on ULIP-2 introduce lightweight prompt tuning (PT) to further boost target tasks (75.80% accuracy). However, we observe the improvements come at the expenses of a severe drop in 3D domain generalization (e.g., 57.07% accuracy on new classes, much behind 71.22%), and develop a systematic regulation constraint (**RC**) framework to address this challenge.

In this paper, we develop our approach based on large 3D foundation models through lightweight prompt learning and propose a comprehensive framework that consists of three regulation constraints to allow the learning trajectory to interact with the well-learned knowledge in large 3D models actively, achieving better task-specific performances and task-agnostic generalization at the same time. Specifically, we propose the mutual agreement constraint to regulate the learnable prompts to produce consistent feature distributions and predictions with the pre-trained foundation models. Then, we exploit the flexible and diverse text descriptions derived from LLMs or manual templates to reflect the attributes of different classes of point clouds and enhance the generalization. Finally, we develop a weighted model ensemble strategy to update the learnable prompts smoothly and predictably, avoiding giant and unexpected leaps toward overfitting the downstream datasets. Some recent works also explore parameter-efficient tuning for point cloud analysis [74, 59, 58, 88], they focus on the performances of downstream tasks while failing to take the model generalization into account. As far as we are aware, our work initiates the first attempt to impose explicit regulation constraints and improve the 3D domain generalization based on large 3D models.

In addition, we argue existing 3D domain generalization evaluation benchmarks, such as PointDA [51] and Sim-to-Real [20], may not be comprehensive to evaluate common generalization capabilities. Only $\sim$10 point cloud object classes are included in these benchmarks. They emphasize the generalization among the shared categories between the source and target domain, without considering transferring to unseen new classes, corrupted data, *etc*, which are frequent in real-world scenarios.

In this paper, three new benchmarks are created to enrich 3D domain generalization evaluation, including *base-to-new class* generalization, *cross-dataset* generalization and *few-shot* generalization. We will dissect the details of benchmark curation and usage in Section A.1. We supply comprehensive experiments and analysis to examine the proposed regulation constraint framework, ablate the effectiveness of distinct components, and draw some new insights from our newly introduced 3DDG evaluation benchmarks. The results verify the proposed method not only enhances the task-specific 3D point cloud recognition but also extends the task-agnostic generalization ability by a clear margin.

In short, the contributions of this work are threefold. ***Firstly***, to our knowledge, we firstly bring the 3DDG problem in front of large multi-modal 3D models and present an effective regulation framework based on lightweight prompt tuning, which not only strengthens downstream 3D task performances but also lifts the domain generalization capability remarkably. ***Secondly***, we implement our regulation framework as a plug-and-play module to seamlessly integrate into the existing large multi-modal 3D models. Consistent improvements are obtained over representative large 3D models, indicating the proposed regulation framework is general and model-agnostic. ***Thirdly***, we carefully craft three new benchmarks to enrich the evaluation of 3D domain generalization. Our benchmarks introduce new evaluation dimensions for 3DDG which are vital in the real world but absent in existing ones, including base-to-new, cross-dataset, and few-shot generalization. These new and more challenging benchmarks will drive the future research of 3D domain generalization.

2 Related Work

3D domain generalization. Although domain generalization has been widely studied in image recognition [17, 28, 34, 29, 31, 30, 32, 83, 21, 87, 5, 38, 13, 84, 80], it is still not the case for 3D. A large body of works in point cloud recognition focuses on improving the performances on specific benchmarks by supervised [48, 49, 33, 67, 60, 63, 18, 69, 82, 37, 68, 50, 14, 44] or self-supervised [73, 43, 75, 57] learning. However, they lack systematic strategies to address the generalization challenge and related evaluation is absent. Only a few methods investigate the 3D domain adaptation [51] and domain generalization [20, 64] problem. They either create a common feature space between the source and target domain (e.g., PointDAN [51], PDG [64]), or utilize the meta-learning framework (e.g., MetaSets [20]) to obtain robust representations to handle the domain shifts. Nevertheless, those methods are all built on small-size point cloud encoders targeting small-scale datasets and the overall generalization is unsatisfactory. In contrast, we explore the 3D domain generalization based on representative large multi-modal 3D models, like PointCLIP series [76, 90] and ULIP series [71, 72]. Meanwhile, we do not touch the backbone and only conduct lightweight prompt tuning on those large 3D models.

Prompt learning for large 3D models. Prompt learning for 3D point cloud understanding has been studied in recent works [74, 58, 59, 88]. IDPT [74], Point-PEFT [59] and DAPT [88] explore this problem in pure point cloud modality and do not establish connections with flexible language descriptions, thus these methods cannot conduct open-vocabulary 3D recognition. PPT [58] firstly constructs a prompt learning pipeline based on the multi-modal framework ULIP [71]. It achieves open-vocabulary recognition with promising performances and relatively small costs. Our work is closely related to PPT but distinguished in the following aspects: First, PPT focuses on optimizing specific 3D tasks with learnable prompts and fails to consider generalization on unseen data. Instead, our work develops systematic strategies to regulate prompt learning to boost generalization as well as target tasks. Second, PPT only introduces learnable prompts in the text branch while our method conducts multi-modal prompt tuning on both text and 3D branches.

3 Method

We firstly revisit lightweight prompt learning for existing large 3D models in section 3.1. Then, a comprehensive regulation framework is proposed to promote the generalization capability of large 3D models based on the plug-and-play prompt tuning strategy in section 3.2. Finally, we introduce the implementation details of the devised method in section 3.3. The overall pipeline of our method is visualized in Fig. 2. The creation and analysis of our new 3DDG benchmarks are elaborated in Appendix A.1 due to space limitation.

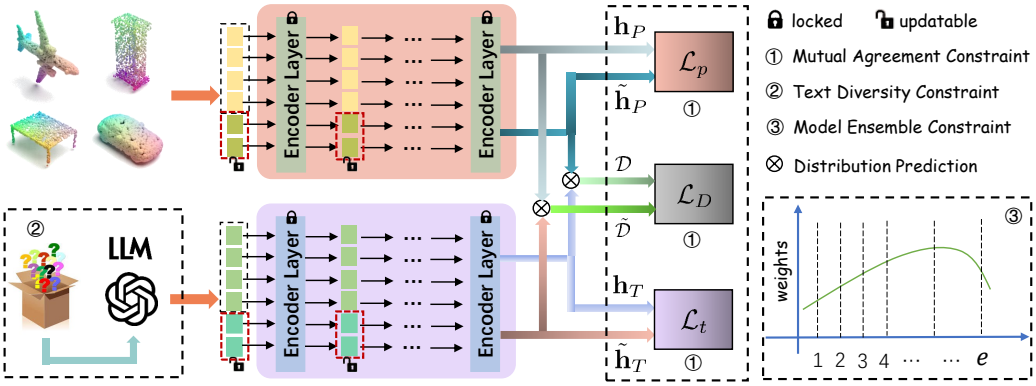

Figure 2: **The overall architecture of our point cloud analysis prompt regulation constraint framework**, namely **Point-PRC**, consisting of three core components as in the figure.

3.1 Preliminary

Existing large multi-modal 3D models [76, 90, 71, 72] have different branches that encode the inputs from point cloud and text. In the 3D branch, a point cloud $P \in \mathbb{R}^{N\times 3}$ is divided and projected into u point patches. Then, a class token $p_{cls} \in \mathbb{R}^d$ is inserted before the patches to form the input $\boldsymbol{P} = \{p_{cls}, p_1, p_2, \ldots, p_u\} \in \mathbb{R}^{(1+u)\times d}$ of the 3D encoder $f_P(\cdot, \theta_p)$, where θ_p represents the encoder parameters. In the text branch, the descriptions of each 3D category are converted into the sequence $\boldsymbol{T} = \{t_{sos}, t_1, t_2, \ldots, t_v, t_c, t_{eos}\} \in \mathbb{R}^{(3+v)\times d}$ for the text encoder $f_T(\cdot, \theta_t)$. Here t_c is the embedding of `{class}`, t_{sos} and t_{eos} stands for the the start and end flag token of a sentence. So we can obtain the 3D features $\mathbf{h}_P = f_P(\boldsymbol{P}, \theta_p)$ and text features $\mathbf{h}_T = f_T(\boldsymbol{T}, \theta_t)$. When executing zero-shot recognition for a downstream 3D dataset of C categories, θ_p and θ_t are frozen, the model outputs the class probability distribution $\mathcal{D}$ for point cloud P by computing $\frac{\exp(sim(\mathbf{h}_P,\, \mathbf{h}_T)/\tau)}{\sum_{j=1}^{C}\exp(sim(\mathbf{h}_P,\, \mathbf{h}_T^j)/\tau)}$, where $sim(\cdot,\cdot)$ measures the cosine similarity of the inputs and τ is a temperature coefficient.

Although zero-shot inference is flexible, the performances on target tasks may not be satisfactory. Multi-modal prompt learning introduces learnable prompts in the inputs of different branches. Specifically, we insert r learnable prompts $\boldsymbol{E}^P = \{e_1^P, e_2^P, \ldots, e_r^P\} \in \mathbb{R}^{r\times d}$ into $\boldsymbol{P}$ and s learnable prompts $\boldsymbol{E}^T = \{e_1^T, e_2^T, \ldots, e_s^T\} \in \mathbb{R}^{s\times d}$ into $\boldsymbol{T}$, respectively. Thereupon, the modified inputs for point cloud and text encoder become $\tilde{\boldsymbol{P}} = \{p_{cls}, p_1, \ldots, p_u, e_1^P, \ldots, e_r^P\}$ and $\tilde{\boldsymbol{T}} = \{t_{sos}, t_1, t_2, \ldots, t_v, t_c, e_1^T, \ldots, e_s^T, t_{eos}\}$. After transforming by the encoders, we obtain the new point cloud and text representations denoted with $\tilde{\mathbf{h}}_P = f_P(\tilde{\boldsymbol{P}}, \tilde{\theta}_p)$ and $\tilde{\mathbf{h}}_T = f_T(\tilde{\boldsymbol{T}}, \tilde{\theta}_t)$, where $\tilde{\theta}_p = \{\theta_p, \boldsymbol{E}^P\}$ and $\tilde{\theta}_t = \{\theta_t, \boldsymbol{E}^T\}$. Similarly, the predicted class distribution $\tilde{\mathcal{D}}$ and optimization objective can be formulated by Eq. 1.

$$\tilde{\mathcal{D}} = \frac{\exp(sim(\tilde{\mathbf{h}}_P, \tilde{\mathbf{h}}_T)/\tau)}{\sum_{j=1}^{C}\exp(sim(\tilde{\mathbf{h}}_P, \tilde{\mathbf{h}}_T^j)/\tau)}, \qquad \{\boldsymbol{E}^{P*},\, \boldsymbol{E}^{T*}\} = \underset{\{\boldsymbol{E}^P,\, \boldsymbol{E}^T\}}{\arg\min}\, \mathbb{E}_{(P,y)\sim\mathcal{D}_{gt}}\mathcal{L}_{CE}(\tilde{\mathcal{D}}, y) \quad (1)$$

where $\mathcal{D}_{gt}$ is the ground truth distribution of point cloud data and y is the category of point cloud P. Note that θ_p and θ_t are still frozen and only E^P and E^T are updatable with the cross entropy (CE) loss. Also, the learnable prompts can be inserted at each layer of the 3D and text encoder, not only at the very first layer. We call this scheme deep multi-modal prompt learning that will be regarded as an important baseline in our experiment settings.

3.2 Our Regulation Framework

Prompt learning aims to elicit well-learned knowledge of pre-trained large models by introducing a small number of learnable parameters in the input space. But optimizing the learnable prompts targeting specific datasets easily compromises the general knowledge. To handle the above problems, we propose a comprehensive regulation framework consisting of three components: mutual agreement constraint, text diversity constraint, and model ensemble constraint, as elaborated below.

3.2.1 Mutual Agreement Constraint (MAC)

Large foundation models unfold overall better robustness and transferability on a broad spectrum of evaluations than conventional models learned on specific datasets, supported by representative works in vision [9], language [53, 54, 4], and multi-modal understanding [52, 23, 15, 45, 86, 85, 90, 71, 19]. The first component of the proposed framework is to interact with large 3D models actively by maximizing the mutual agreement between learnable prompts and pre-trained knowledge.

Specifically, we engage with large 3D models by aligning extracted features and predicted distributions simultaneously. Let us denote the frozen point cloud feature extracted by the 3D foundation model as $\mathbf{h}_P$, the point cloud feature containing learnable prompts as $\tilde{\mathbf{h}}_P$. Now we compute the difference between $\mathbf{h}_P$ and $\tilde{\mathbf{h}}_P$ and mark it as $\mathcal{L}_p$. Similarly, in the text modality, we have $\mathcal{L}_t$ which measures the difference between $\mathbf{h}_T$ and $\tilde{\mathbf{h}}_T$. On the other side, $\mathcal{D}$ and $\tilde{\mathcal{D}}$ are two class distributions given by the frozen and promptable large 3D models, respectively. The difference between $\mathcal{D}$ and $\tilde{\mathcal{D}}$ is denoted as $\mathcal{L}_D$. Our mutual agreement constraint aims to minimize the feature and prediction distribution discrepancy to ensure the learning trajectory not to forget the task-agnostic knowledge in large pre-trained models.

$$\mathcal{L}_p = \sum_i |\, \mathbf{h}_P^i - \tilde{\mathbf{h}}_P^i \,|, \quad \mathcal{L}_t = \sum_i |\, \mathbf{h}_T^i - \tilde{\mathbf{h}}_T^i \,|, \quad \mathcal{L}_D = \sum_i \mathcal{D}_{KL}(\mathcal{D}_i \,||\, \tilde{\mathcal{D}}_i) \tag{2}$$

As formulated in Eq. 2, L_1 distance is employed to compute $\mathcal{L}_p$ and $\mathcal{L}_t$, and Kullback-Leibler (KL) divergence is used to characterize the distribution discrepancy. We will examine the design choices in the ablation study.

3.2.2 Text Diversity Constraint (TDC)

Inspired by the flexibility and versatility of language expressions, we propose to leverage diverse text descriptions to guide the lightweight prompt tuning to produce transferrable features. Specifically, we obtain multiple text descriptions for each point cloud object category by prompting LLMs ($e.g.$, GPT-3.5 [1], GPT-4 [41], PointLLM [70]) or utilizing manual templates. Then, we aggregate the text feature of all descriptions for each single category by pooling operation, $\mathbf{h}_T = \text{AvgPool}(\sum_j \mathbf{h}_T^j)$, which will integrate rich semantic information extracted by powerful large models, prevent a point cloud category biasing towards some specific descriptions and finally enhance the model transferability. In the case of describing point clouds with LLMs, we design three kinds of prompts, including question answering, caption generation, and making sentences using keywords, as demonstrated in Fig. 3. For each instruction to the LLM, we acquire $N_t = 10$ responses.

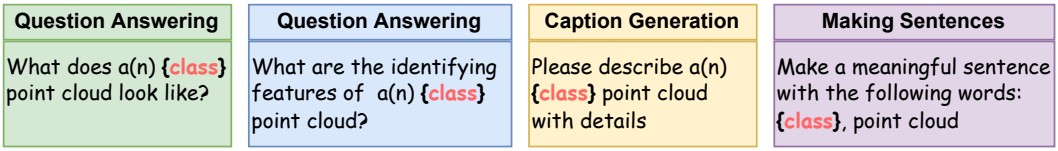

Figure 3: **Illustration of diverse questions to LLMs**, including GPT-3.5, GPT-4 and PointLLM. The responses given by LLMs are regarded as the text descriptions to the point cloud and fed into the text encoder.

3.2.3 Model Ensemble Constraint (MEC)

The model ensemble constraint aims to synthesize the opinions from different models by weighted voting to avoid some extreme and failure cases of a single model. The idea has been widely discussed in statistical machine learning [42, 12] and deep learning [16, 40]. Robust tuning of multi-modal large models by ensemble learning also has been studied in recent literature [65, 22]. The ensemble strategy mainly involves interpolating weights between zero-shot and fully fine-tuned large models. But it has not been investigated in the context of prompt tuning for large 3D models and its effectiveness is unknown. In this paper, we propose to ensemble models by aggregating the model parameters in different training epochs with a Gaussian weighted strategy. The basic idea is that in the initial learning stage, the prompts are randomly initialized and not well optimized so we distribute them very

small weights. As the training iterates, the model gradually gets a sense of downstream tasks; thus, increasing weights are assigned to the model parameters in these epochs. As the training ends, the learnable prompts are adjusted well to adapt downstream datasets while having the risk of overfitting, so we decrease the weights to the model parameters. The varying weights of the above process can be approximated by a gaussian curve. Finally, the weighted models in different epochs are ensembled to generate the model parameters $\tilde{\theta}_t$ and $\tilde{\theta}_p$, shown in Eq. 3.

$$\tilde{\theta}_p = \sum_{i=1}^{e} w_i \tilde{\theta}_p^i, \quad \tilde{\theta}_t = \sum_{i=1}^{e} w_i \tilde{\theta}_t^i \tag{3}$$

where e is the number of epochs and $w_i = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(i-\mu)^2}{2\sigma^2})$. μ and σ^2 represent the mean and variance of a gaussian distribution. $\tilde{\theta}_p^i = \{\theta_t^i, E^{P_i}\}$ and $\tilde{\theta}_t^i = \{\theta_t^i, E^{T_i}\}$ indicate the model parameters after the ith epoch of training in the text and point cloud branch, respectively. Note that a simple accumulated addition can implement Eq. 3 and we do not need to store all e copies of the parameters, referring to Appendix for details.

Optimization. The overall optimization objective consists of two parts, the task-specific cross entropy loss $\mathcal{L}_{CE}$ and the task-agnostic regulation constraint loss $\mathcal{L}_{RC}$, displayed in Eq. 4, where α, β, γ are hyperparameters. Unlike trivial prompt tuning a multi-modal large model on downstream tasks, this design allows the learnable prompts to actively interact and align with the general knowledge in a pre-trained large model while learning on specific 3D tasks.

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{RC}, \qquad \mathcal{L}_{RC} = \alpha\mathcal{L}_p + \beta\mathcal{L}_t + \gamma\mathcal{L}_D \tag{4}$$

3.3 Implementation Details

We choose PointCLIP [71], PointCLIP V2 [71], ULIP [71], and ULIP-2 [72] as the 3D foundation models for experiments. All experiments are running with three random seeds and we report the mean and standard deviation. The learnable prompts are inserted into the inputs of first 9 Transformer layers in these models and the prompt length is set to 2. Unless specified, the prompts are optimized using 16-shot learning. Note that previous 3DDG methods [51, 20, 64] use the full training set. We set $\alpha = 10$, $\beta = 25$ and $\gamma = 1$. The optimizer is SGD, the initial lr is 0.0025 and we use cosine scheduler to update it. More details about the model configuration can be found in Appendix. We will justify the design choices in the ablation study.

4 Experiments

In this section, we first explain the evaluation settings of our newly curated and existing benchmarks. Then, comprehensive comparison and analysis across various generalization settings are presented to show the advantages of the proposed method. Finally, we justify the effectiveness of different components in our regulation framework through systematic controlled experiments.

4.1 3DDG Evaluation Settings

Base-to-New. This benchmark includes 5 point cloud datasets which are ModelNet40 [2], three variants of ScanObjectNN [61] (S-PB_T50_RS, S-OBJ_BG, S-OBJ_ONLY) and ShapeNetCoreV2 [6]. Each dataset is equally split into base and new classes, where the former is used for prompt tuning while the latter only serves the test purpose. **Cross-Dataset**. This benchmark has four types of evaluation, including *OOD generalization*, *data corruption*, *PointDA* [51] and *Sim-to-Real* [20]. We established the first two assessments and the latter two already existed. For *OOD generalization*, models are trained on the source domain and evaluated on the target domains. For *data corruption*, models are trained on clean ModelNet [2] and tested on the corrupted data in ModelNet-C[56]. **Few-Shot**. This setting inspects the model generalization in an extremely low-data regime, where 1, 2, 4, 8, and 16 shots are randomly sampled for prompt learning, and the recognition accuracy is calculated on the whole test set, respectively. The explanations are brief and we encourage the readers to check the details in Appendix.

Table 1: **Base-to-new class generalization comparison for representative large 3D models based on prompt learning**. Each number here is the mean of three runnings. Base: base class accuracy (in %, same below). New: new class accuracy. HM: harmonic mean of base and new class accuracy. **+RC** demonstrates the models with our regulation constraint framework.

(a) **Average over 5 datasets**

Method	Base	New	HM
P-CLIP [76]	75.66	23.45	35.80
P-CLIP2 [90]	74.11	37.84	50.10
ULIP [71]	77.32	49.01	59.99
+**RC**(Ours)	**82.19**	**61.93**	**70.64**
ULIP-2 [72]	77.91	67.91	72.57
+**RC**(Ours)	**83.18**	**76.10**	**79.48**

(b) ModelNet40

Method	Base	New	HM
P-CLIP [76]	93.23	20.22	33.23
P-CLIP2 [90]	93.98	45.21	61.05
ULIP [71]	92.80	50.07	65.05
+**RC**(Ours)	**95.03**	**55.27**	**69.89**
ULIP-2 [72]	91.77	56.47	69.92
+**RC**(Ours)	**95.30**	**64.83**	**77.17**

(c) S-PB_T50_RS

Method	Base	New	HM
P-CLIP [76]	61.25	19.87	30.01
P-CLIP2 [90]	56.84	29.92	39.20
ULIP [71]	56.73	25.80	35.47
+**RC**(Ours)	**64.20**	**49.17**	**55.69**
ULIP-2 [72]	66.40	66.47	66.43
+**RC**(Ours)	**73.67**	**74.27**	**73.97**

(d) S-OBJ_BG

Method	Base	New	HM
P-CLIP [76]	72.82	23.00	34.96
P-CLIP2 [90]	70.07	35.08	46.75
ULIP [71]	73.20	47.17	57.37
+**RC**(Ours)	**79.47**	**55.20**	**65.15**
ULIP-2 [72]	77.00	83.27	80.01
+**RC**(Ours)	**80.10**	**88.93**	**84.28**

(e) S-OBJ_ONLY

Method	Base	New	HM
P-CLIP [76]	76.23	20.23	31.97
P-CLIP2 [90]	71.40	44.39	54.74
ULIP [71]	74.13	50.80	60.29
+**RC**(Ours)	**79.23**	**65.93**	**71.97**
ULIP-2 [72]	78.60	76.27	77.42
+**RC**(Ours)	**83.60**	**81.10**	**82.33**

(f) ShapeNetCoreV2

Method	Base	New	HM
P-CLIP [76]	74.78	33.92	46.61
P-CLIP2 [90]	78.27	34.58	47.97
ULIP [71]	89.73	71.20	79.40
+**RC**(Ours)	**93.03**	**84.10**	**88.34**
ULIP-2 [72]	75.80	57.07	65.38
+**RC**(Ours)	**83.23**	**71.37**	**76.85**

4.2 Base-to-new Class Generalization

In this benchmark, models are learned on the base classes and evaluated on the test sets of base and novel classes. In addition to ULIP and ULIP-2, we also implement the same prompt tuning for PointCLIP [76] (P-CLIP) and PointCLIP V2 [90] (P-CLIP2) for comparison, shown in Tab. 1.

Loss of Generalization in P-CLIP and ULIP Series. We observe notable gaps occur between base and new class recognition accuracy of P-CLIP, P-CLIP2, ULIP, ULIP-2 when prompt tuning without the proposed regulation constraints. For instance, P-CLIP2 achieves 93.98% accuracy on the base classes of ModelNet40 while dropping by 48.77% absolute points on the whole test set of the new classes, which even lags behind the zero-shot accuracy of the frozen P-CLIP2 (64.22%). The results are consistent across five datasets, suggesting the loss of generalization of original models.

Lifting the Generalization by Our Framework. As shown in Tab. 1, the proposed framework composite of three regulation constraints boosts the unseen class recognition accuracy across different models and datasets by a clear margin, thanks to the active communication and alignment with the general knowledge in large 3D models. For example, the improvement of the harmonic mean on ULIP reaches 10.65% absolute points averaged over 5 datasets.

Lifting the Specific 3D Tasks by Our Framework. Surprisingly, the task-specific performances are not be hindered by the regulation constraints while enhancing the task-agnostic generalization, referring to the base class accuracy of ULIP+**RC** and ULIP-2+**RC** averaged over 5 datasets, increasing by 4.87% and 5.27%, respectively.

4.3 Cross-Dataset Generalization

This setting differs from the base-to-new counterpart where the base and new classes belong to the same dataset. We present the analysis for *OOD generalization* and *data corruption* as below, and put the comparison on *Sim-to-Real* and *PointDA* in Appendix.

OOD Generalization demonstrates the models' transferability to other unseen domains by learning from an existing domain. To evaluate on this benchmark, we implement the lightweight prompt learning for ULIP and ULIP-2 then impose the proposed regulation constraints on them. Prompt learning for P-CLIP [76] and P-CLIP2 [76] with same settings are also implemented for comparison. The results are reported in Tab. 2 . By wrapping ULIP and ULIP-2 with the devised framework, we achieve consistent positive gains on each of the five target domains. The average gains over them are enlarged with increasing ability of ULIP, *e.g.*, +6.20% for ULIP-2 vs. +1.79% for ULIP. Meanwhile, we notice that the performances on Omni3D [66] are rather limited and the methods here seem not to

Table 2: **Comparison of OOD generalization in cross-dataset benchmark**. ShapeNetV2 serves as the source domain and the other five datasets are deployed as the target domain. ShapeNetV2: 55 classes, ModelNet40: 40 classes, SONN: 15 classes, Omni3D: 216 classes. Some common categories are shared between the source and target domain. Note that Omni3D has much more new 3D object concepts than others. The last column indicates the average over five target datasets.

Method	Source	Target					Avg.
	ShapeNetV2	ModelNet40	S-PB_T50_RS	S-OBJ_BG	S-OBJ_ONLY	Omni3D	
P-CLIP [76]	67.41(0.09)	33.20(1.86)	15.51(0.58)	18.59(1.40)	22.89(2.32)	0.48(0.17)	22.55(1.54)
P-CLIP2 [90]	68.93(1.43)	54.73(1.48)	39.53(4.22)	**34.30**(1.28)	**25.63**(1.16)	8.63(2.52)	32.56(2.13)
+**RC**(Ours)	**69.80**(2.86)	**55.37**(1.78)	**39.77**(0.45)	34.20(0.54)	24.50(1.26)	**10.20**(0.40)	**32.81**(0.89)
ULIP [71]	87.33(0.95)	56.17(1.15)	26.83(2.15)	39.43(2.17)	43.53(1.32)	6.37(0.90)	34.47(1.54)
+**RC**(Ours)	**90.43**(0.86)	**58.00**(0.57)	**28.43**(0.68)	**40.33**(0.71)	**46.33**(1.54)	**8.20**(0.50)	**36.26**(0.80)
ULIP-2 [72]	76.70(1.37)	65.27(0.66)	40.07(0.34)	53.80(1.78)	48.53(1.72)	17.27(0.54)	44.99(1.01)
+**RC**(Ours)	**76.70**(1.59)	**72.10**(0.93)	**46.77**(2.43)	**59.03**(3.02)	**56.27**(0.97)	**21.80**(0.49)	**51.19**(1.57)

Table 3: **Comparison of corruption generalization on ModelNet-C[56] when trained on clean data**. The results are reported for the corruption severity=2 in ModelNet-C.

Method	Clean Data	Corruption Type							Avg.
	ModelNet	Add Global	Add Local	Drop Global	Drop Local	Rotate	Scale	Jitter	
P-CLIP [76]	80.97(1.02)	80.97(1.02)	80.97(1.02)	64.95(1.08)	68.31(1.93)	65.75(1.19)	72.04(1.33)	52.09(1.28)	69.30(1.26)
P-CLIP2 [90]	83.49(0.51)	83.49(0.51)	83.49(0.51)	68.85(3.22)	66.67(1.96)	70.13(1.33)	75.68(0.15)	61.21(2.16)	72.79(1.41)
ULIP [71]	82.43(1.25)	82.50(0.99)	82.27(1.17)	80.77(1.03)	65.43(1.02)	72.27(1.56)	74.67(1.58)	**45.60**(0.65)	71.93(1.14)
+**RC**(Ours)	**83.87**(0.34)	**83.83**(0.40)	**83.93**(0.19)	**81.83**(0.52)	**67.37**(1.72)	**79.10**(0.36)	**76.37**(0.09)	41.67(4.79)	**73.44**(1.15)
ULIP-2 [72]	85.07(0.21)	81.97(0.79)	82.03(0.96)	79.93(0.92)	60.03(1.21)	80.30(0.93)	75.77(0.74)	44.27(2.13)	72.04(1.10)
+**RC**(Ours)	**86.47**(0.56)	**86.57**(0.48)	**86.30**(0.51)	**84.87**(0.48)	**67.80**(1.20)	**84.60**(0.22)	**81.17**(1.05)	**46.43**(2.45)	**76.82**(0.91)

work, especially for P-CLIP series and ULIP (less than 10% accuracy). This dataset contains a large vocabulary of real 3D objects (216 categories) and exhibits the long-tail attribute. When transferring the models that learn from a narrow set of 3D object concepts (55 classes in ShapeNetV2) to Omni3D, they suffer from new 3D concepts thus perform poorly.

Data Corruption are common in point clouds due to complex geometry, sensor inaccuracy and processing imprecision. We investigate the generalization of the proposed framework on ModelNet-C [56], which includes common corruptions, such as dropping some parts or adding global outliers. The compared methods are same as those in OOD generalization and the results are exhibited in Tab. 3. Our method not only boosts the recognition accuracy on clean data (+1.44% for ULIP and +1.40% for ULIP-2), but also strengthen the robustness of representative large 3D models against collapsed data. By averaging on 7 types of corruption, we receive +1.51% and +4.78% gains for ULIP and ULIP-2, respectively.

4.4 Few-shot Generalization

In this setting, ULIP and ULIP-2 with (w.) and without (w.o.) our regulation constraints (RC) are compared. As visualized in Fig. 4, the solid lines of ULIP and ULIP-2 exceed the corresponding dashed lines by clear margins average over 5 datasets, indicating the devised framework strengthens the 3DDG capability considerably. The advantages are enlarged especially for the extreme 1-shot learning, *e.g.*, +8.05% acc. for ULIP and +5.39% acc. for ULIP-2. Note that in some cases, *e.g.*, on ModelNet40, ULIP-2 w.o. RC (1-shot, 66.63%) even lags behind zero-shot ULIP-2 (71.23%), implying that simple prompt tuning disturbs the well-learned representations of ULIP-2. In contrast, the developed framework brings 2.4% absolute improvements over the zero-shot ULIP-2, obtaining 73.63% acc. under the 1-shot setting.

4.5 Ablation Study

In this section, we examine the effectiveness of several critical components in the proposed framework via a series of controlled experiments. ULIP-2 is adopted as the baseline and we compare the variants on the base-to-new benchmark and report the harmonic mean (HM) averaged over 5 datasets.

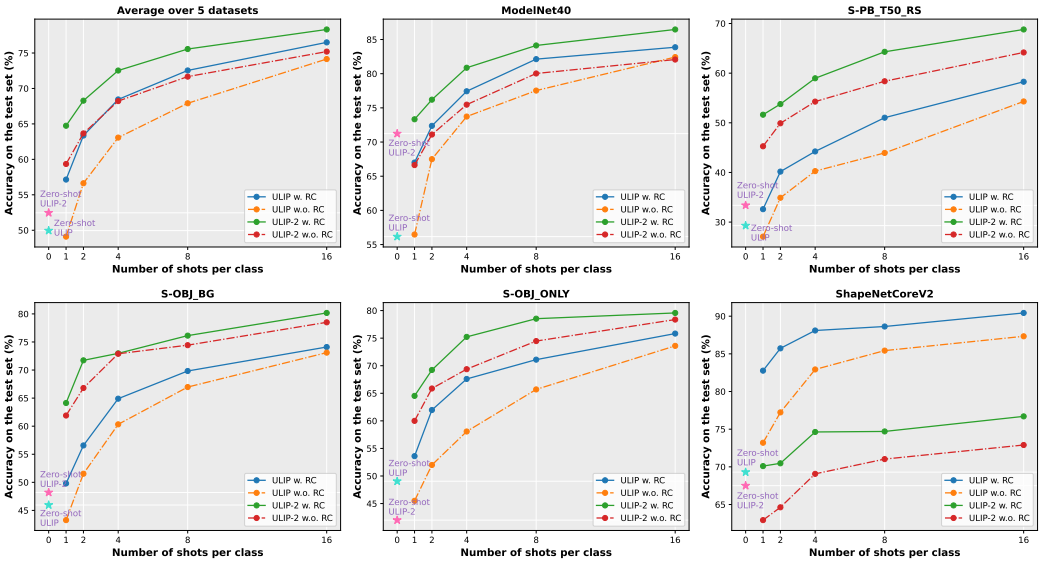

Figure 4: **Comparison of few-shot generalization**. The solid and dashed lines represent the models with and without our framework. Zero-shot performances of ULIP and ULIP-2 are marked with star symbols. The figure in the upper left presents the average results over 5 datasets.

Regulation constraints. Three components in our framework are vital for generalization enhancement. We verify their effectiveness by adding/deleting the components on ULIP-2. The results in Tab. 4 indicate there exists a notable performance gap (6.91%) between ULIP-2 with and without the regulation constraints. And the gap can be gradually narrowed down by inserting different components. For instance, ULIP-2 with the model ensemble constraint lifts the HM from 72.57% to 78.89%, a 6.32% absolute increase. Although a single mutual agreement or text diversity constraint may not bring adequate gains, their combination contributes to the generalization improvement significantly, achieving 79.26% that is close to the performance of the full version of our framework.

Distance metrics in MAC. The mutual agreement among the extracted features with learnable prompts ($e.g.$ $\tilde{\mathbf{h}}_T$ and $\tilde{\mathbf{h}}_P$) and the general knowledge in large 3D models ($e.g.$ $\mathbf{h}_T$ and $\mathbf{h}_P$) can be implemented in different distance metrics. Here we explore their effect and report the results in Tab. 5a. As observed, MAC with MSE distance attains the best recognition acc. on the base classes. But when incorporating the performances of new 3D categories into account, the same model with L1 distance demonstrates overall better generalization (76.10% new acc. and 79.48% HM). Thus we choose L1 as the distance metric in MAC by default.

Table 4: **Ablation study for the three regulation constraints in our framework**. The results are averaged on 5 datasets.

MAC	TDC	MEC	Base	New	HM
✗	✗	✗	77.91	67.91	72.57
✓	✗	✗	78.02	68.91	73.18
✗	✓	✗	78.89	70.63	74.53
✗	✗	✓	84.54	72.19	78.89
✗	✓	✓	**84.92**	71.17	77.44
✓	✗	✓	83.19	73.35	77.96
✓	✓	✗	83.30	75.59	79.26
✓	✓	✓	83.18	**76.10**	**79.48**

Table 5: Ablation studies. The results are averaged over 5 datasets in the base-to-new benchmark.

(a) The distance metrics in MAC. L1: L1 norm, MSE: mean square error, Cosine: cosine distance.

Metric	Base	New	HM
Cosine	78.63	73.73	76.10
MSE	**83.91**	72.81	77.97
L1	83.18	**76.10**	**79.48**

(b) Here GPT-3.5 is short for GPT-3.5-turbo.

Text Description	Base	New	HM
GPT-3.5 [1]	83.30	71.44	76.92
GPT-4 [41]	**83.46**	71.55	77.05
PointLLM [70]	83.27	73.83	78.27
Manual	83.18	**76.10**	**79.48**

Point cloud descriptions from different sources. We hope to exploit flexible and diverse text descriptions to reflect some vital characteristics of the point clouds in different classes. The following

experiments investigate the effect of the point cloud descriptions generated from different sources, including large language models like GPT-3.5 [1], GPT-4 [41], PointLLM [70] and manual templates (see Appendix for details). As shown in Tab. 5b, point cloud descriptions from general-purpose LLMs, such as GPT-3.5 and GPT-4, bring decent performances on base classes. However, they lag behind PointLLM regarding new class recognition accuracy by a clear margin (-2.39% for GPT-3.5 and -2.28% for GPT-4). We infer it is due to the fact that PointLLM has seen massive point cloud data and related text descriptions thus generates more accurate and domain-related responses. Surprisingly, by combining 64 simple sentences written by human beings [71], ULIP-2 achieves decent base class accuracy and the best performance on new classes, resulting in even better HM than that of ULIP-2 with LLMs' descriptions.

The depth and length of learnable prompts. Two variables that should be determined for the learnable prompts $\{\boldsymbol{E}^P, \boldsymbol{E}^T\}$ are the depth of prompt layers D and the length of prompt tokens L. For simplicity, the prompt depth is kept the same in the point cloud and text encoders, and similarly for the prompt length. We ablate the two variables and visualize the results in Fig. 5. In general, increasing the prompt layers promotes the harmonic mean. But it is not always beneficial to deepen the learnable prompts, $e.g.$ ULIP-2 with $D = 12$ achieves 78.26% HM, slightly lower than 78.67% HM of ULIP-2 with $D = 9$. We also find that it is not necessary to construct very long prompt tokens to achieve better generalization, $e.g.$, ULIP-2 with $L = 2$ surpasses other variants average on 5 datasets clearly. Thus we let $D = 9$ and $L = 2$ by default.

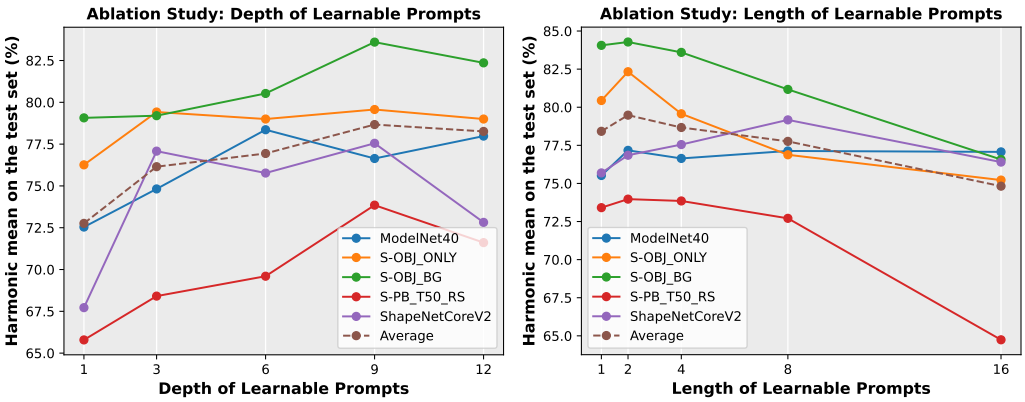

Figure 5: **Ablation study for the prompt depth and length**. We compare the harmonic mean on five datasets of the base-to-new benchmark and the average results are displayed in dashed lines.

5 Conclusion

This paper initializes the efforts of addressing the corrupted generalization of large 3D models when adapting to downstream 3D tasks by a comprehensive regulation framework. The framework enables the learnable prompts to actively engage with large 3D models by maximizing the mutual agreement between task-specific prediction and general knowledge. Consistent generalization gains are obtained over different large 3D models, suggesting the model-agnostic attribute of the proposed framework. We also contribute to the study of 3DDG by developing new and more challenging evaluation benchmarks that will drive further investigation. Nevertheless, this work focuses on the point cloud recognition, and we plan to discuss the segmentation and detection tasks in future work.

Limitations and Broader Impacts. The proposed framework has demonstrated effectiveness and scalability on the object-level recognition task but not been validated on the scene-level tasks, such as 3D semantic segmentation and object detection. Different solutions may be required to handle scene-level point cloud data. On the other hand, when exploiting the power of LLMs to reflect critical characteristics of 3D objects, we simply ensemble multiple descriptions through the pooling operation, more sophisticated prompting and fusion strategy can be developed. For broader impacts, we are the first to investigate the generalization ability of large multi-modal 3D models, which mirrors the progress of the vision-language field (CLIP-based image recognition) and probably inspires a series of follow-up works. We do not perceive the potential negative impacts of this work.

Acknowledgments and Disclosure of Funding

We thank all anonymous reviewers and area chairs for their time and valuable feedback. This work was partially supported by China Scholarship Council (CSC) under the Grant No. 202306360147 and partially supported by NeurIPS 2024 Student Travel Grant.

References

[1] Openai chatgpt. https://chatgpt.com/. Accessed: 2024-05-06.

[2] The princeton modelnet. https://modelnet.cs.princeton.edu/. Accessed: 2012-12-27.

[3] J. Behley and C. Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. *Robotics: Science and Systems XIV*, 2018.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[5] M.-H. Bui, T. Tran, A. Tran, and D. Phung. Exploiting domain-specific features to enhance domain generalization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21189–21201. Curran Associates, Inc., 2021.

[6] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015.

[7] G. Chen, W. Yao, X. Song, X. Li, Y. Rao, and K. Zhang. Prompt learning with optimal transport for vision-language models. In *ICLR*, 2023.

[8] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. P. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, R. Jenatton, L. Beyer, M. Tschannen, A. Arnab, X. Wang, C. Riquelme Ruiz, M. Minderer, J. Puigcerver, U. Evci, M. Kumar, S. V. Steenkiste, G. F. Elsayed, A. Mahendran, F. Yu, A. Oliver, F. Huot, J. Bastings, M. Collier, A. A. Gritsenko, V. Birodkar, C. N. Vasconcelos, Y. Tay, T. Mensink, A. Kolesnikov, F. Pavetic, D. Tran, T. Kipf, M. Lucic, X. Zhai, D. Keysers, J. J. Harmsen, and N. Houlsby. Scaling vision transformers to 22 billion parameters. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7480–7512. PMLR, 23–29 Jul 2023.

[10] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 35799–35813. Curran Associates, Inc., 2023.

[11] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153, June 2023.

[12] T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, page 1–15, Berlin, Heidelberg, 2000. Springer-Verlag.

[13] Y. Ding, L. Wang, B. Liang, S. Liang, Y. Wang, and F. Chen. Domain generalization by learning and removing domain-specific features. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24226–24239. Curran Associates, Inc., 2022.

[14] L. Duan, S. Zhao, N. Xue, M. Gong, G.-S. Xia, and D. Tao. Condaformer: Disassembled transformer with local structure enhancement for 3d point cloud understanding. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[15] A. Fang, G. Ilharco, M. Wortsman, Y. Wan, V. Shankar, A. Dave, and L. Schmidt. Data determines distributional robustness in contrastive language image pre-training (CLIP). In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6216–6234. PMLR, 17–23 Jul 2022.

[16] M. Ganaie, M. Hu, A. Malik, M. Tanveer, and P. Suganthan. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.*, 115(C), oct 2022.

[17] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2551–2559, 2015.

[18] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu. Pct: Point cloud transformer, 2020.

[19] Z. Guo, R. Zhang, X. Zhu, Y. Tang, X. Ma, J. Han, K. Chen, P. Gao, X. Li, H. Li, and P.-A. Heng. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following, 2023.

[20] C. Huang, Z. Cao, Y. Wang, J. Wang, and M. Long. Metasets: Meta-learning on point sets for generalizable representations. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8859–8868, 2021.

[21] Z. Huang, H. Wang, E. P. Xing, and D. Huang. Self-challenging improves cross-domain generalization. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part II 16*, pages 124–140. Springer, 2020.

[22] G. Ilharco, M. Wortsman, S. Y. Gadre, S. Song, H. Hajishirzi, S. Kornblith, A. Farhadi, and L. Schmidt. Patching open-vocabulary models by interpolating weights. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 29262–29277. Curran Associates, Inc., 2022.

[23] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 18–24 Jul 2021.

[24] M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19113–19122, June 2023.

[25] M. U. khattak, S. T. Wasim, N. Muzzamal, S. Khan, M.-H. Yang, and F. S. Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

[26] D. Lee, S. Song, J. Suh, J. Choi, S. Lee, and H. J. Kim. Read-only prompt optimization for vision-language few-shot learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1401–1411, 2023.

[27] J. Levinson and S. Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *2010 IEEE International Conference on Robotics and Automation*, pages 4372–4378, 2010.

[28] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[29] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Learning to generalize: Meta-learning for domain generalization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[30] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. Hospedales. Episodic training for domain generalization. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1446–1455, 2019.

[31] H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] H. Li, Y. Wang, R. Wan, S. Wang, T.-Q. Li, and A. Kot. Domain generalization for medical imaging classification with linear-dependency regularization. In *Advances in Neural Information Processing Systems*, volume 33, pages 3118–3129. Curran Associates, Inc., 2020.

[33] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[34] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[35] Z. Li, X. Li, X. Fu, X. Zhang, W. Wang, S. Chen, and J. Yang. Promptkd: Unsupervised prompt distillation for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[36] Y. Lu, J. Liu, Y. Zhang, Y. Liu, and X. Tian. Prompt distribution learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5196–5205, 2022.

[37] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations*, 2022.

[38] D. Mahajan, S. Tople, and A. Sharma. Domain generalization using causal matching. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7313–7324. PMLR, 18–24 Jul 2021.

[39] K. S. McCurley. Geospatial mapping and navigation of the web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW'01, page 221–229, New York, NY, USA, 2001. Association for Computing Machinery.

[40] A. Mohammed and R. Kora. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*, 35(2):757–774, 2023.

[41] OpenAI. GPT-4. https://openai.com/research/gpt-4, 2023. Accessed on January 22, 2024.

[42] D. Opitz and R. Maclin. Popular ensemble methods: an empirical study. *J. Artif. Int. Res.*, 11(1):169–198, jul 1999.

[43] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision – ECCV 2022*. Springer International Publishing, 2022.

[44] J. Park, S. Lee, S. Kim, Y. Xiong, and H. J. Kim. Self-positioning point-based transformer for point cloud understanding. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21814–21823, 2023.

[45] H. Pham, Z. Dai, G. Ghiasi, K. Kawaguchi, H. Liu, A. W. Yu, J. Yu, Y.-T. Chen, M.-T. Luong, Y. Wu, M. Tan, and Q. V. Le. Combined scaling for zero-shot transfer learning, 2023.

[46] A. P. Placitelli and L. Gallo. Low-cost augmented reality systems via 3d point cloud sensors. In *2011 Seventh International Conference on Signal Image Technology and Internet-Based Systems*, pages 188–192, 2011.

[47] C. R. Qi, O. Litany, K. He, and L. J. Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

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

[49] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[50] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[51] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[52] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[53] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[54] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[55] H. Ran, J. Liu, and C. Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18942–18952, June 2022.

[56] J. Ren, L. Pan, and Z. Liu. Benchmarking and analyzing point cloud classification under corruptions. In *International Conference on Machine Learning (ICML)*, 2022.

[57] H. Sun, Y. Wang, X. Cai, X. Bai, and D. Li. Vipformer: Efficient vision-and-pointcloud transformer for unsupervised pointcloud understanding. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[58] H. Sun, Y. Wang, W. Chen, H. Deng, and D. Li. Parameter-efficient prompt learning for 3d point cloud understanding. In *IEEE International Conference on Robotics and Automation*, 2024.

[59] Y. Tang, R. Zhang, Z. Guo, X. Ma, B. Zhao, Z. Wang, D. Wang, and X. Li. Point-peft: Parameter-efficient fine-tuning for 3d pre-trained models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6):5171–5179, Mar. 2024.

[60] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[61] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[63] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[64] X. Wei, X. Gu, and J. Sun. Learning generalizable part-based feature representation for 3d point clouds. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 29305–29318. Curran Associates, Inc., 2022.

[65] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, and L. Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971, June 2022.

[66] T. Wu, J. Zhang, X. Fu, Y. Wang, J. Ren, L. Pan, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 803–814, June 2023.

[67] W. Wu, Z. Qi, and L. Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[68] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022.

[69] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 915–924, October 2021.

[70] R. Xu, X. Wang, T. Wang, Y. Chen, J. Pang, and D. Lin. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023.

[71] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese. Ulip: Learning unified representation of language, image and point cloud for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[72] L. Xue, N. Yu, S. Zhang, J. Li, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

[73] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19313–19322, June 2022.

[74] Y. Zha, J. Wang, T. Dai, B. Chen, Z. Wang, and S.-T. Xia. Instance-aware dynamic prompt tuning for pre-trained point cloud models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[75] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, and H. Li. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27061–27074. Curran Associates, Inc., 2022.

[76] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8552–8562, June 2022.

[77] R. Zhang, X. Hu, B. Li, S. Huang, H. Deng, Y. Qiao, P. Gao, and H. Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15211–15222, June 2023.

[78] R. Zhang, L. Wang, Y. Qiao, P. Gao, and H. Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21769–21780, June 2023.

[79] R. Zhang, L. Wang, Y. Wang, P. Gao, H. Li, and J. Shi. Parameter is not all you need: Starting from non-parametric networks for 3d point cloud analysis. June 2023.

[80] Y. Zhang, M. Zhang, W. Li, S. Wang, and R. Tao. Language-aware domain generalization network for cross-scene hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–12, 2023.

[81] Z. Zhang, X. Gao, and W. Hu. Invariantoodg: Learning invariant features of point clouds for out-of-distribution generalization, 2024.

[82] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16259–16268, October 2021.

[83] S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao. Domain generalization via entropy regularization. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16096–16107. Curran Associates, Inc., 2020.

[84] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2023.

[85] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[86] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022.

[87] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021.

[88] X. Zhou, D. Liang, W. Xu, X. Zhu, Y. Xu, Z. Zou, and X. Bai. Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[89] B. Zhu, Y. Niu, Y. Han, Y. Wu, and H. Zhang. Prompt-aligned gradient for prompt tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15613–15623, 2023.

[90] X. Zhu, R. Zhang, B. He, Z. Guo, Z. Zeng, Z. Qin, S. Zhang, and P. Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

A Appendix

A.1 Our New 3DDG Benchmarks

To our knowledge, PointDA [51] and Sim2Real [20] are rare benchmarks in 3DDG. We perceive that existing 3DDG benchmarks may not be sufficient to cover common generalization evaluation scenarios. For instance, PointDA [51] selected 10 shared classes among three popular point cloud datasets [2, 8, 6] for generalization evaluation. Sim2Real [20] picked 9 same categories among ShapeNet [6] and ScanObjectNN [61] and 11 shared classes in ModelNet [2] and ScanObjectNN [61]. Both of them emphasize the generalization among shared 3D object classes in different datasets. But they fail to consider how to transfer to unseen 3D object classes and other out-of-distribution scenarios. To alleviate the drawbacks, we develop three new benchmarks, including base-to-new, cross-dataset, and few-shot generalization to enrich 3DDG evaluation and drive future research.

A.1.1 Construction

Base-to-new Class Benchmark. Inspired by the evaluation settings in 2D vision [85], we take five 3D datasets, including ModelNet40 [2], three variants of ScanObjectNN [61] (S-PB_T50_RS, S-OBJ_BG, S-OBJ_ONLY), and ShapeNetCoreV2 [6] to construct the base-to-new benchmark. Each of them are equally divided into two halves object classes, called base and new classes. Specifically, the first half is regarded as base and the second half is treated as new. The train, val, test sets are split for the base classes while the new classes only serve the test purpose. The splitting adopts the official standards of the released datasets if available (e.g., ModelNet40 and ShapeNetCoreV2), otherwise (e.g., 3 variants of ScanObjectNN) we randomly selected 20% samples in the original training set to be the validation set and keep the remaining 80% as new training set. The original test set is unchanged to serve as the test set. This arrangement allows the model to be trained on base classes then tested on new classes so that can measure its generalization on unseen new categories and data.

Table 6: Statistics of the Base-to-New benchmark.

Item	ModelNet40		S-OBJ_ONLY		S-OBJ_BG		S-PB_T50_RS		ShapeNetCoreV2	
	base	new	base	new	base	new	base	new	base	new
#classes	20	20	8	7	8	7	8	7	28	27
#train	4,084	N/A	1,055	N/A	1,062	N/A	5,321	N/A	23,584	N/A
#val	1,028	N/A	281	N/A	274	N/A	1,268	N/A	3,401	N/A
#test	1,202	1,266	352	229	352	229	1,741	1,141	6,960	3,301

Cross-Dataset Benchmark. Previous Sim2Real [20] and PointDA [51] evaluate the generalization across totally shared categories in different datasets. But in real world the source and target domains do not necessarily have common classes. In this work, we construct the cross-dataset benchmark to expand the generalization evaluation to broader scopes, which incorporates two newly introduced settings, *OOD generalization* and *data corruption*, and another *Sim2Real* [20] and *PointDA* [51]. Note that the last two datasets are released by other researchers for 3DDG evaluation and we just follow their default settings.

OOD Generalization. In this setting, the source and target domain may not necessarily share common categories. Also, the number of object classes can be different. ShapeNetCoreV2 [6] is arranged as the source dataset while ModelNet40 [2], the three variants of ScanObjectNN [61] and Omni3D [66] serve as the target evaluation places. Apparently, this design is specially for large 3D models that have open-vocabulary recognition ability since traditional 3DDG methods like MetaSets [20], PDG [64] are only able to recognize a fixed set of point cloud classes.

Data Corruption. Point cloud corruption is inevitable due to irregular geometry structures, inaccurate sensors or processing errors. Existing 3DDG benchmarks fail to take this factor into account. We utilize the off-the-shelf point cloud corruption dataset ModelNet-C [56] as a place to measure the robustness and generalization of point cloud recognition methods against common data corruptions, such as losing local parts, global noise, *etc.*

Few-shot Benchmark. This benchmark incorporates same datasets with those in the *Base-to-new Class* Benchmark. But we do not distinguish the base and new classes and treat them as a whole. During prompt learning on each dataset of this benchmark, we randomly sample 1, 2, 4, 8 and 16 shots from each category to tune the learnable prompts, then observe the generalization on the whole test set.

A.1.2 The Evaluation Settings

Base-to-New. This benchmark includes five point cloud datasets as described above. Each dataset is equally split into base (known) and new (unseen) classes. We compute the recognition accuracy (Acc.) on the two types of classes, respectively. Although the accuracy on the new classes reflects how well a model can learn from known point cloud categories to generalize other unseen data, we also want decent accuracy on the base classes. Consequently, the harmonic mean (HM) between the accuracies of base and new classes is chosen to balance the

Table 7: Statistics of the Cross-Dataset benchmark.

(a) *OOD Generalization*

Item	Source	Target				
	ShapeNetCoreV2	ModelNet40	S-OBJ_ONLY	S-OBJ_BG	S-PB_T50_RS	Omni3D
#classes	55	40	15	15	15	216
#train	35,708	7,861	1,847	1,847	9,132	N/A
#val	5,158	1,979	462	462	2,284	N/A
#test	10,261	2,468	581	581	2,882	5,910

(b) *Data Corruption*

Item	Add Global	Add Local	Drop Global	Drop Local	Rotate	Scale	jitter
#classes	40	40	40	40	40	40	40
#test	2,468	2,468	2,468	2,468	2,468	2,468	2,468

(c) *Sim-to-Real*

Item	Source	Target		
	ModelNet	S-OBJ_ONLY	S-OBJ_BG	S-PB_T50_RS
#classes	11	11	11	11
#train	4,844	N/A	N/A	9,132
#test	972	475	475	2,882

(d) *PointDA*

Item	ModelNet	ShapeNet	ScanNet
#classes	10	10	10
#train	4,183	17,378	6,110
#test	856	2,492	1,769

Table 8: Statistics of the Few-shot benchmark.

Item	ModelNet40	S-OBJ_ONLY	S-OBJ_BG	S-PB_T50_RS	ShapeNetCoreV2
#classes	40	15	15	15	55
#train	7,861	1,847	1,847	9,132	35,708
#val	1,979	462	462	2,284	5,158
#test	2,468	581	581	2,882	10,261

two factors. Similar evaluation settings are also adopted in vision-language community [85, 36, 7, 24, 25, 89, 35]. Note that the models trained on a fixed set of point cloud categories cannot be evaluated on this benchmark since they do not have the open-vocabulary recognition ability, such as IDPT [74], Point-PEFT [59], DAPT [88].

Cross-Dataset. This benchmark has four types of evaluation settings and we will focus on the first two settings which are, *OOD generalization* and *data corruption*, newly introduced in 3DDG by us. The last two correspond to the default settings in PointDA [51] and Sim-to-Real [20]. For *OOD generalization*, models are trained on the source domain and directly transferred to the target domains for evaluation. Recognition accuracy is a major metric. We will average the accuracies across five target domains to measure the final generalization ability. For *data corruption*, models are trained on clean ModelNet [2] and tested on various point cloud corruption scenarios in ModelNet-C[56], such as add global noises, losing local parts, geometry transformations, *etc.* The average accuracies on 7 types of atomic corruptions are computed to measure the model robustness against point cloud data corruption.

Few-Shot. This setting inspects the model generalization in extremely low-data regime, where only a few samples of each class are offered to train a model then it is evaluated on the whole test set. Here we take 1, 2, 4, 8, and 16 shots for the model training and the recognition accuracy on the whole test set is compared.

We insert the proposed explicit constraints to large 3D models, then conduct lightweight prompt tuning on downstream 3D tasks, finally observe whether positive gains will appear in above 3DDG evaluation settings compared to the same prompt learning but without our regulation framework.

A.2 Implementation Details

In gaussian model weighting, we take $\mu = 15$ and $\sigma = 1$. To avoid store e separate copies of the model parameters, we implement Eq. 3 by iteratively adding current and previous epoch of weighted model parameters. Note the learnable parameters are randomly initialized before training. The design ensures our model absorbing prior knowledge and reduces disk consumption effectively. Both ULIP and ULIP-2 exploit the Point-BERT [73] as the backbone.

We use two RTX 4090 GPUs to run the experiments. On the base-to-new benchmark, we conduct prompt learning for 20 epochs. On the cross-dataset and few-shot benchmark, models are trained for 50 epochs, and the prompt depth D and length L are set to 12 and 4, respectively. For the evaluation on ModelNet-C, we report the results when the corruption severity is 2. The 64 manual templates and other 3D object classes descriptions generated by LLMs are available at our provided codebase.

A.3 Additional Results and Analysis

A.3.1 Base-to-new Class Generalization

Table 9 enriches the *base-to-new class* generalization evaluation by reporting the mean and standard deviation of three runnings on the *base-to-new class* benchmark. The standard deviation in the bracket follows the mean. Note that ULIP and ULIP-2 with our regulation framework have much smaller deviation on novel classes, $e.g.$, 2.38% vs 3.77%, suggesting better generalization and robustness against the variations between seen and unseen point cloud data.

Table 9: **Base-to-new class generalization comparison for representative large 3D models based on prompt learning**. Each number here is the mean of three runnings. Base: base class accuracy (in %, same below). New: new class accuracy. HM: harmonic mean of base and new class accuracy. **+RC** demonstrates the models with our regulation constraint framework.

(a) **Average over 5 datasets**

Method	Base	New	HM
P-CLIP [76]	75.66	23.45	35.80
P-CLIP2 [90]	74.11	37.84	50.10
ULIP [71]	77.32(1.41)	49.01(3.77)	59.99
+**RC**(Ours)	**82.19**(1.22)	**61.93**(2.38)	**70.64**
ULIP-2 [72]	77.91(0.95)	67.91(3.75)	72.57
+**RC**(Ours)	**83.18**(0.62)	**76.10**(1.14)	**79.48**

(b) ModelNet40

Method	Base	New	HM
P-CLIP [76]	93.23	20.22	33.23
P-CLIP2 [90]	93.98	45.21	61.05
ULIP [71]	92.80(0.93)	50.07(3.52)	65.05
+**RC**(Ours)	**95.03**(0.52)	**55.27**(3.03)	**69.89**
ULIP-2 [72]	91.77(0.41)	56.47(2.78)	69.92
+**RC**(Ours)	**95.30**(0.36)	**64.83**(0.26)	**77.17**

(c) S-PB_T50_RS

Method	Base	New	HM
P-CLIP [76]	61.25	19.87	30.01
P-CLIP2 [90]	56.84	29.92	39.20
ULIP [71]	56.73(0.84)	25.80(2.33)	35.47
+**RC**(Ours)	**64.20**(0.99)	**49.17**(2.55)	**55.69**
ULIP-2 [72]	66.40(1.39)	66.47(2.40)	66.43
+**RC**(Ours)	**73.67**(0.56)	**74.27**(1.27)	**73.97**

(d) S-OBJ_BG

Method	Base	New	HM
P-CLIP [76]	72.82	23.00	34.96
P-CLIP2 [90]	70.07	35.08	46.75
ULIP [71]	73.20(2.32)	47.17(1.76)	57.37
+**RC**(Ours)	**79.47**(1.92)	**55.20**(2.89)	**65.15**
ULIP-2 [72]	77.00(1.04)	83.27(3.76)	80.01
+**RC**(Ours)	**80.10**(0.99)	**88.93**(0.24)	**84.28**

(e) S-OBJ_ONLY

Method	Base	New	HM
P-CLIP [76]	76.23	20.23	31.97
P-CLIP2 [90]	71.40	44.39	54.74
ULIP [71]	74.13(0.60)	50.80(7.71)	60.29
+**RC**(Ours)	**79.23**(1.44)	**65.93**(2.21)	**71.97**
ULIP-2 [72]	78.60(0.37)	76.27(4.72)	77.42
+**RC**(Ours)	**83.60**(0.37)	**81.10**(2.69)	**82.33**

(f) ShapeNetCoreV2

Method	Base	New	HM
P-CLIP [76]	74.78	33.92	46.61
P-CLIP2 [90]	78.27	34.58	47.97
ULIP [71]	89.73(2.38)	71.20(3.51)	79.40
+**RC**(Ours)	**93.03**(1.23)	**84.10**(1.24)	**88.34**
ULIP-2 [72]	75.80(1.56)	57.07(5.09)	65.11
+**RC**(Ours)	**83.23**(0.80)	**71.37**(1.23)	**76.85**

A.3.2 Cross-dataset Generalization

Sim-to-Real evaluation measures the 3D domain generalization from simulating data to real world. This evaluation was first introduced by MetaSets [20] then followed by PDG [64]. In this setting, ModelNet [2]

and ShapeNet [6] are regarded as synthetic point cloud and ScanObjectNN is constructed based on real-scan data. Here we implement the proposed regulation framework upon P-CLIP2 [90], ULIP [71], ULIP-2 [72] and compare with prior state-of-the-art. Note that MetaSets and PDG use the whole training set in the source domain for supervised learning, whereas our methods only exploit 16-shot prompt tuning. As Tab. 10 shows, our framework brings consistent generalization improvement on different large 3D models, +0.52% for P-CLIP2, +5.88% for ULIP and +2.48% for ULIP-2 average on 6 datasets. The enhanced ULIP-2 by the devised framework also outreaches prior best-performing PDG-D by 5.82%, demonstrating better generalization to real-world point cloud data.

Table 10: **Comparison of cross dataset generalization on *Sim-to-Real***. There are two evaluation settings, MN_11 → SONN_11, SN_9 → SONN_9. The left side of → stands for simulating data and the right side indicates real-world data. 11 classes are shared between ModelNet and ScanObjectNN, 9 classes are common in ShapeNet and ScanObjectNN. In the experiments, a point cloud contains 2048 points. -P: PointNet, -D: DGCNN.

Method	MN_11 → SONN_11			SN_9 → SONN_9			**Avg.**
	OBJ	OBJ_BG	PB_T50_RS	OBJ	OBJ_BG	PB_T50_RS	
MetaSets-P [90]	60.3	52.4	47.4	51.8	44.3	45.6	50.3
MetaSets-D [90]	58.4	59.3	48.3	49.8	47.4	42.7	51.0
PDG-P [64]	67.6	58.5	56.6	57.3	51.3	51.3	57.1
PDG-D [64]	65.3	65.4	55.2	59.1	59.3	51.0	59.2
P-CLIP2 [90]	18.67(1.68)	**15.57**(3.23)	**15.63**(1.63)	53.00(3.06)	47.83(1.84)	35.83(0.19)	31.09(1.94)
+**RC**(Ours)	**19.23**(4.00)	15.50(3.88)	14.37(2.57)	**56.60**(3.70)	**47.83**(3.80)	**36.10**(2.81)	**31.61**(3.46)
ULIP [71]	21.60(2.50)	18.03(2.03)	13.63(1.51)	54.83(1.66)	54.17(2.46)	40.87(1.27)	33.86(1.91)
+**RC**(Ours)	**29.90**(1.15)	**24.07**(2.03)	**18.87**(2.52)	**63.13**(0.74)	**58.87**(0.49)	**43.60**(1.51)	**39.74**(1.41)
ULIP-2 [72]	62.73(0.95)	68.23(0.86)	52.83(1.10)	66.90(2.77)	70.50(2.48)	54.03(2.75)	62.54(1.82)
+**RC**(Ours)	**68.43**(1.07)	**69.47**(0.95)	**55.30**(2.00)	**65.83**(1.35)	**72.53**(0.47)	**58.57**(1.17)	**65.02**(1.17)

PointDA is a 3D domain adaptation benchmark introduced by PointDAN [51], which includes 6 evaluation settings as displayed in Tab. 11. Previous methods like MetaSets [20], PDG [64], I-OODG [81] exploit the full training data in each setting while we adopt few-shot learning (16 shots). The results suggest the proposed framework contributes to the enhanced domain adaptation significantly, $e.g.$, almost 10% absolute improvements for ULIP-2 that leads prior state-of-the-art I-OODG by 6.94%.

Table 11: **Comparison of cross-dataset generalization on *PointDA***. M: ModelNet, S: ShapeNet, S*: ScanNet. The last column is the average over 6 evaluation settings.

Method	M → S	M → S*	S → M	S → S*	S* → M	S* → S	**Avg.**
P-DAN [51]	64.2	33.0	47.6	33.9	49.1	64.1	48.7
MetaSets [20]	86.0	52.3	67.3	42.1	69.8	69.5	64.5
PDG [64]	85.6	57.9	73.1	50.0	70.3	66.3	67.2
I-OODG [81]	83.7	56.4	71.7	57.6	69.5	73.5	67.8
P-CLIP2 [90]	40.53	26.40	31.33	35.57	16.30	24.97	29.18
+**RC**(Ours)	**43.27**	**37.20**	**32.47**	**36.97**	**21.70**	**43.90**	**35.92**
ULIP [71]	74.33(8.63)	38.23(2.12)	35.17(3.99)	36.17(5.67)	24.70(5.16)	60.67(4.72)	44.88(5.05)
+**RC**(Ours)	**78.80**(1.49)	**41.63**(0.87)	**43.03**(4.28)	**41.60**(4.02)	**25.23**(4.68)	**63.60**(9.01)	**48.98**(4.06)
ULIP-2 [72]	84.80(2.69)	48.10(2.13)	83.20(4.17)	42.00(4.18)	60.43(4.83)	70.50(6.22)	64.84(4.04)
+**RC**(Ours)	**89.00**(1.18)	**51.37**(1.03)	**89.87**(2.38)	**49.57**(2.50)	**85.57**(3.80)	**83.07**(4.21)	**74.74**(2.52)

A.3.3 The Role of MEC

Selecting the best checkpoint through a validation set is a common way. In theory, this greedy strategy favors highest performances on the downstream tasks, which means the small number of learnable prompts/parameters are well adapted to these tasks. It is equivalent to our framework without *Model Ensemble Constraint* (MEC).

However, purely optimizing the small number of learnable prompts toward target tasks will inevitably hinder the generalization ability of the large 3D models, as we analyzed in the paper.

We also provide the ablation study to this problem, as the results in Tab. 12 indicates, the method without MEC has slightly lower accuracy on new classes (75.59% vs 76.10%) and harmonic mean. However, when removing the factors of MAC and TDC, the role of MEC becomes prominent. It raise the overall performance remarkably, especially for unseen new classes (5.28% absolute points).

Table 12: **Ablation study for the framework without model ensembling constraint**. The results are averaged on 5 datasets. MAC: mutual agreement constraint, TDC: text diversity constraint, MEC: model ensemble constraint. HM: harmonic mean of the Base and New class accuracies.

Model	Variant			Performance		
	MAC	TDC	MEC	Base	New	HM
ULIP-2	✗	✗	✗	77.91	67.91	72.51
	✗	✗	✓	82.42	73.19	77.53
	✓	✓	✗	83.30	75.59	79.26
	✓	✓	✗	83.18	76.10	79.48

A.3.4 Running Time

We compare the training time between our method and the baselines. The results are shown in Tab. 13. The proposed method consumes a similar amount of time per epoch compared to the baseline, with a slight increase due to the inclusion of our framework.

Table 13: **Running time comparison of a strong baseline ULIP-2 and the proposed approach**. We conduct prompt learning based on ULIP-2 for 20 epochs on the base-to-new class benchmark, and the experiments are run three times with different seeds. The settings are consistent with those in the main paper. Time is counted in *seconds* for all 20 epochs using a RTX 4090.

Method	seed	Dataset					**Avg.**
		MN40	S-PB_T50_RS	S-OBJ_BG	S S-OBJ_ONLY	SNV2	
ULIP-2	1	132	106	48	53	307	129.2
	2	132	106	48	53	305	128.8
	3	133	108	48	51	305	129.6
+RC(Ours)	1	159	112	60	60	344	147.0
	2	159	114	60	59	345	147.4
	3	159	113	59	60	345	147.2

The number of learnable parameters of our framework is 16,896 while full fine-tuning ULIP-2 has 82.3M learnable parameters (only in text and 3D encoder). According to the reported details of ULIP-2, pre-training on Objaverse [11] utilizes 8 A100 GPUs and takes 1.5 days. So full fine-tuning ULIP-2 is also expensive.

A.3.5 Scalability

We further test our framework on a larger dataset named Objaverse-LVIS and the results are promising. This dataset is a subset of the recently released Objaverse and only serves as a test set (target domain). Objaverse-LVIS contains 46,205 point clouds distributed in 1,156 classes, and some classes only have a single object, posing great challenges to existing point cloud recognition methods. In the experiments, we select representative ULIP and ULIP-2 as baselines and compare them with the models with our regulation framework. The results in the Tab. 14 verify the proposed approach can also bring considerable gains (+3.27% absolute points for ULIP-2) on such a larger and challenging dataset.

Table 14: **Analysis of scalability**.

Method	**Source**	**Target**
	ShapeNetV2	Objaverse-LVIS
ULIP	87.33(0.95)	0.83(0.05)
+RC(Ours)	**90.43**(0.86)	**1.10**(0.08)
ULIP-2	76.70(1.37)	14.80(0.22)
+RC(Ours)	**76.70**(1.59)	**18.07**(0.49)

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: Refer to Abstract and Section 1.

   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: Refer to Section 5.

   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: [NA]

   Justification: This paper does not present a new theory or proof.

   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: Refer the implement details and Section 4.

   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: [Yes]

   Justification: The data and code are released at https://github.com/auniquesun/Point-PRC.

   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: Refer to the implementation details in Section 3 and Appendix.

   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 run each experiment three times and report the mean and standard deviation, referring to the tables in Section 4.

   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 use two RTX 4090 GPUs to run the experiments, referring to Appendix.

   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: Before submission, we read the NeurIPS Code of Ethics and obey them during this study.

   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: We discuss the broader impacts of this work in the last section.

    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: To our knowledge, we do not perceive the potential risks of this work.

    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: We follow the same licenses of the used datasets and cite the original paper, referring to Section 4 and Appendix.

    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: [Yes]

    Justification: We release the new benchmarks and provided related README at `https://github.com/auniquesun/Point-PRC`.

    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: Our work does not relate to crowdsourcing and 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: Our work does not relate to the 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.