Supplementary Material of CLIPoint3D: Language-Grounded Few-Shot Unsupervised 3D Point Cloud Domain Adaptation

*Mainak Singha[1] *Sarthak Mehrotra[2,3] Paolo Casari[1] Subhasis Chaudhuri[3]
Elisa Ricci[1,4] Biplab Banerjee[3]
[1] University of Trento, Italy [2] MDSR Labs Adobe, India [3] IIT Bombay, India [4] Fondazione Bruno Kessler, Italy
{mainak.singha, paolo.casari, e.ricci}@unitn.it

1. Contents of the supplementary materials

In this supplementary document, we present detailed information and further experimental results, including:

1. **Dataset descriptions:** In Table 1, we provide the total number of point cloud samples in the training and test splits of each domain of each datasets, though we do few-shot training in our proposed method.
2. **LLM attributes generation:** In Fig. 1, we show the pipeline of generation of high-level knowledge attributes using an LLM.
3. **Pseudo-code of the CLIPoint3D algorithm:** In Algorithm 1, we provide the detailed procedure of our proposed method through a pseudo-code.
4. **Analysis of the LoRA rank:** In Fig. 3 we showcase the effect of the rank of LoRA metrices in CLIPoint3D on both datasets.
5. **Conventional plug-in UDA methods in CLIP baselines:** In Table 2 we provide an analysis of training our CLIP-based zero-shot baselines using traditional UDA methods e.g. DANN [5], CDAN [10] and SCDA [9].
6. **Effect of α hyperparameter:** In Fig. 4, we showcase the importance of α hyperparameter used in the total loss function of Eq. 13.
7. **Influence of the length of prompt q:** In Fig. 5, we show the effect of shared prompt length in CLIPoint3D.
8. **Impact of CLIP variants:** In Table 3 we analyze the effect of CLIP backbones i.e. ViT-B/16, ViT-B/32 and ViT-L/14 in our proposed CLIPoint3D method.
9. **Effect of various LLMs:** We also ablate how the attributes generated from different LLMs e.g. GPT-5 [12], Llama-3.2-3B [8], Qwen2.5-14B [19], Phi-4 [1] etc can effect the performances of our method.

2. Dataset descriptions

The PointDA-10 benchmark collects object point clouds from ModelNet40 [18], ShapeNet [2], and ScanNet [3], covering ten shared object categories. ModelNet-10 (**M**) dataset contains 4,183 training and 856 testing samples obtained by following online perspective projection [7] unlike post rendering [17], i.e., simply projecting each point onto a series of pre-defined image planes to generate scatter depth maps. ShapeNet-10 (**S**) dataset includes 17,378 training and 2,492 testing point clouds, exhibiting greater structural

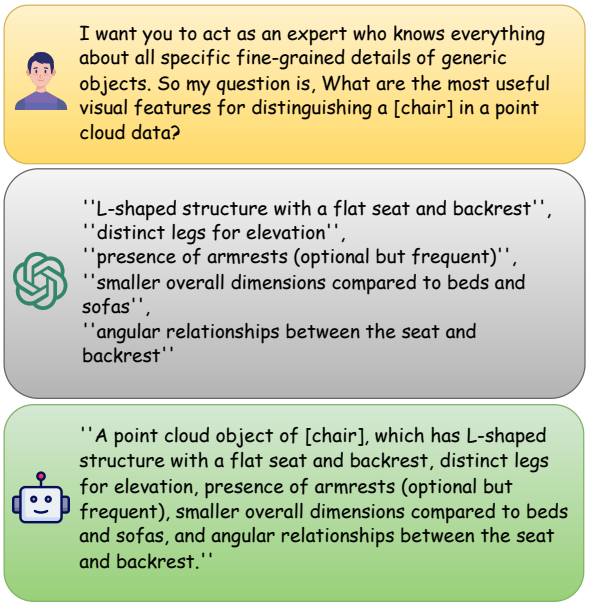
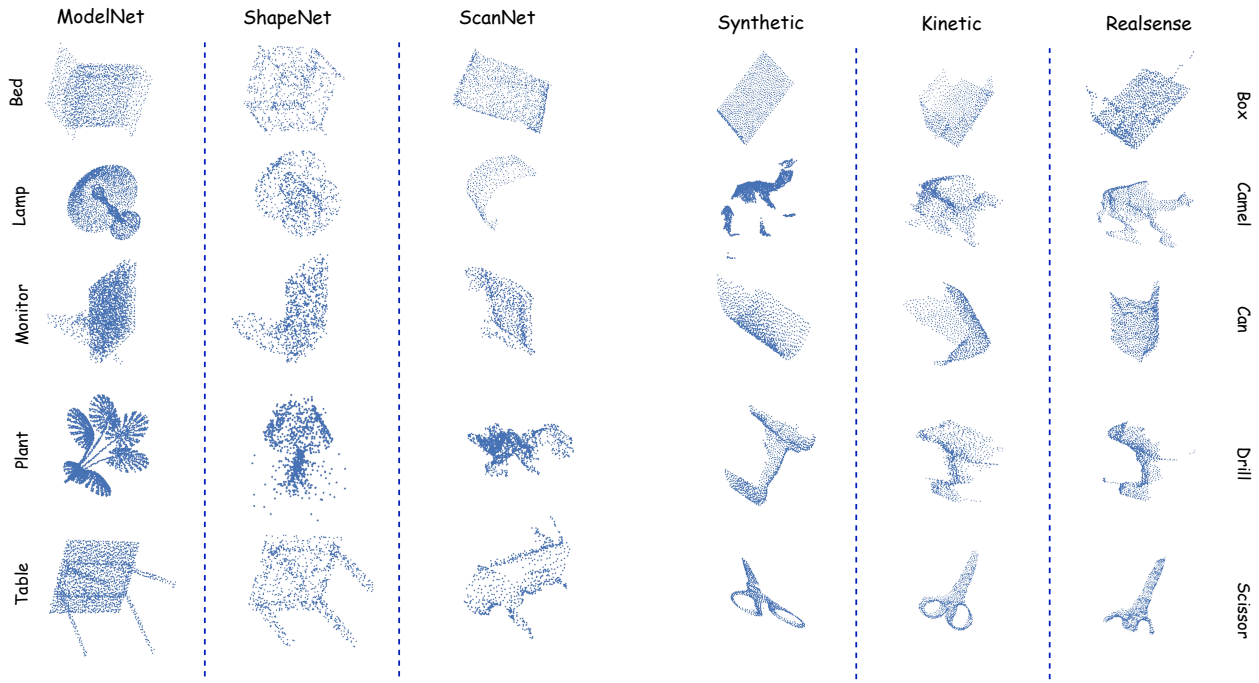


Figure 1. **LLM attributes generation.** To derive high-level 3D knowledge representations, we follow a three-stage pipeline. First (top box), we provide an instructional query prompt to a LLM (e.g. GPT-5 [12]). In response, the LLM produces detailed, geometry-aware visual descriptions (middle box). Finally (bottom box), we generate highly contextualized textual prompts (one caption per class) by combining a modality-specific prefix template with the LLM-generated attributes.

*equal contribution

Table 1. Domain Generation dataset statistical details on class, training and test splits, prefix template.

Dataset	Domains	Common Classes	# Samples	# Training / Test
PointDA-10 [13]	ModelNet [18]	Bathtub, Bed, Bookshelf,	5039	4183 / 856
	ShapeNet [2]	Cabinet, Chair, Lamp,	19870	17378 / 2492
	ScanNet [3]	Monitor, Plant, Sofa, Table	7879	6110 / 1769
GraspNetPC-10 [4, 15]	Synthetic	Banana, Box, Can,	12,000	12,000 / -
	Kinect	Camel, Dish, Drill, Mouse,	13,533	10,973 / 2560
	Realsense	Pear, Scissors, Shampoo	13,258	10,698 / 2560



(a) PointDA-10 (b) GraspNetPC-10

Figure 2. **Domain Visualization.** We show the diverse geometry variations across the domains of PointDA-10 and GraspNetPC-10 datasets.

diversity due to its larger number of object instances and wider geometric variation. ScanNet-10 (**S***) dataset consists of 6,110 training and 1,769 testing point clouds, containing sensor noise, occlusions, and missing surfaces inherent to reconstructed indoor scenes.

GraspNetPC-10 benchmark is constructed from GraspNet [4], a large-scale dataset designed for robotic grasping from raw depth scans and reconstructed CAD models. The point clouds are generated by re-projecting depth maps into 3D space and cropping objects using segmentation masks. Unlike PointDA-10, the point clouds in GraspNetPC-10 are not aligned. This benchmark includes three domains: Synthetic (**Syn.**), Kinect (**Kin.**), and Realsense (**RS.**), corresponding to CAD-rendered depth scans and raw sensor captures from two different depth cameras. The synthetic domain contains 12,000 training samples, while the Kinect and Realsense domains contain 10,973/2,560 and 10,698/2,560 training/testing samples, respectively. Real-world scans from two different depth cameras i.e. Kinect2 and Intel Realsense exhibit domain-specific artifacts, including varying noise patterns, geometric distortions, and missing regions.

In Figure 2, we show the diverse geometric variations of different point cloud class objects in synthetic (ModelNet, ShapeNet, Synthetic) and real-world (ScanNet, Kinect, Realsense) environments on both the PointDA-10

and GraspNetPC-10 benchmarks.

3. LLM attributes generation

To generate descriptive attributes for each class, we leverage an LLM i.e. GPT-5 [12]. Each class label is passed through a structured instructional prompt, adapted and expanded from the template proposed in [11], as shown in Figure 1. Specifically, We follow a three-stage pipeline similar to [16] to construct the attributes integrating two complementary components: a modality-specific prefix template and the attribute set produced by the LLM. First, we design a prefix that is tailored to the imaging modality of interest i.e. ``A point cloud object of [class]''. We then propose a way to enrich this prefix by appending the combination of the LLM-generated attributes on a single sentence using connective phrases such as which is a/an'' or which has'', i.e. a single descriptive attribute for each class, which is different from others. This yields a semantically detailed and context-aware prompt that captures both modality information and discriminative visual characteristics. A complete example for the class ``chair'' is shown in the third row of Figure 1.

4. Pseudo-code of CLIPoint3D

In Algorithm 1 we provide the detailed pseudo-codes of training and inference process of CLIPoint3D algorithm.

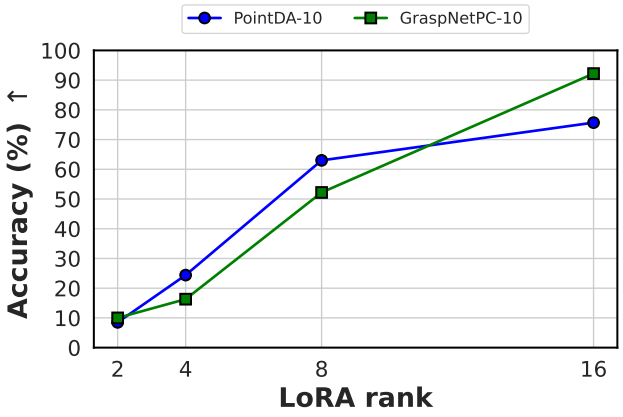


Figure 3. **Effect of varying LoRA rank.** We report the adaptation performances of CLIPoint3D-V and CLIPoint3D-B on PointDA-10 and GraspNetPC-10 datasets respectively.

5. Analysis of the LoRA rank

To understand the influence of the low-rank decomposition on adaptation quality, we conduct an ablation study on LoRA ranks $2, 4, 8$ & 16 using our CLIPoint3D framework. We choose the vision and both encoder variants for the PointDA-10 and GraspNetPC-10 benchmarks respectively. As shown in Figure 3, increasing the LoRA rank consistently improves performance, but the rate of improvement differs markedly between the two datasets.

On PointDA-10, which contains relatively clean synthetic CAD models alongside noisier real scans, accuracy improves steadily from rank 2 to rank 16. The sharp gain from rank 4 to rank 8 indicates that a moderate rank is essential to capture the geometric variability and structural inconsistencies across domains. Beyond rank 8, the improvement becomes more modest, suggesting diminishing returns as representational capacity saturates. Whereas on GraspNetPC-10, whose features are more complex real-world sensor noise and greater intra-class variation, the benefit of increasing the LoRA rank is even more pronounced. Performance rises from rank 2 to rank 16, with a substantial leap between rank 8 and rank 16. The results show that higher-rank LoRA modules in CLIPoint3D provide the expressive adaptability and generalizability on realistic depth distortions, object incompleteness, and viewpoint variability present in the point cloud domains.

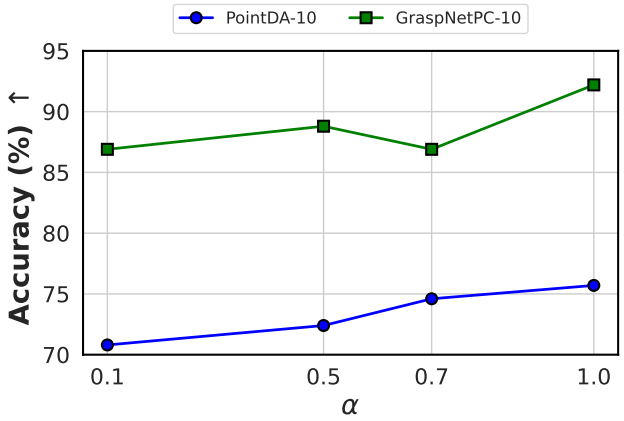


Figure 4. **Effect of varying α hyperparameter.** We report the adaptation performances of CLIPoint3D-V and CLIPoint3D-B on PointDA-10 and GraspNetPC-10 datasets respectively.

6. Conventional plug-in UDA methods in CLIP baselines

In Table 2, we evaluate the effect of integrating standard UDA techniques e.g. DANN [5], CDAN [10], and SCDA [9] into our CLIP-based baselines (Zs-CLIP, PointCLIP & PointCLIPv2) to train them for point-cloud UDA task. While these conventional UDA methods can bring modest improvements in certain cross-domain transfers, their gains are inconsistent and often fail to fully bridge the domain gap inherent in 3D point cloud data. To be noted, we have just added a learnable adapter on top of the frozen visual features of the vision encoder similar to [6], while keeping both

Table 2. **Comparison of plug-in UDA methods in CLIP baselines with CLIPoint3D**. We report the adaptation performances on the PointDA-10 benchmark. M: ModelNet, S: ShapeNet, S*: ScanNet; → indicates the adaptation direction. Best results and second-best results are reported in bold and underlined, respectively.

Methods	M→S	M→S*	S→M	S→S*	S* →M	S* →S	Avg
ZS-CLIP [14]	46.1	17.0	52.0	17.0	52.0	46.1	38.4
CLIP [14] + DANN [5]	62.0	8.6	77.3	10.5	56.7	50.4	44.3
CLIP + CDAN [10]	60.9	7.0	76.5	11.0	56.7	50.0	43.7
CLIP + SCDA [9]	46.5	16.2	51.2	17.0	51.8	46.2	38.2
PointCLIP [20]	50.8	20.9	50.1	20.9	50.1	50.8	40.6
PointCLIP [20] + DANN	55.3	9.8	74.2	14.3	50.4	49.7	42.3
PointCLIP + CDAN	55.8	9.2	72.1	13.7	50.9	49.3	41.8
PointCLIP + SCDA	39.2	17.6	70.9	19.8	67.6	37.6	42.1
PointCLIPv2 [21]	38.8	19.5	71.6	19.5	71.6	38.8	43.3
PointCLIP [20] + DANN	46.2	12.2	80.4	13.6	79.5	40.6	45.4
PointCLIP + CDAN	44.6	12.8	75.7	12.9	76.8	41.7	44.1
PointCLIP + SCDA	45.9	12.0	74.8	12.5	77.2	40.2	43.8
CLIPoint3D-T	74.4	9.5	86.0	24.1	50.5	59.8	50.7
CLIPoint3D-V	**84.6**	**53.5**	**91.6**	**55.3**	**87.9**	81.3	**75.7**
CLIPoint3D-B	81.5	51.9	90.3	46.6	85.2	**85.8**	73.6

Table 3. **Effect of using various CLIP variants on CLIPoint3D-V**. We report the adaptation performances on the PointDA-10 benchmark.

Methods	ViT-B/16	ViT-B/32	ViT-L/14
ModelNet→ShapeNet	84.6	82.4	85.7
ModelNet→ScanNet	53.5	42.7	54.4
ShapeNet→ModelNet	91.6	88.3	92.1
ShapeNet→ScanNet	55.3	36.9	59.5
ScanNet→ModelNet	87.9	88.2	88.5
ScanNet→ShapeNet	81.3	73.7	82.3
Average	75.7	68.7	77.1

the encoders entirely frozen. While the CLIP-based methods improve adaptation after pluggin on the UDA methods, but still underperform compared to CLIPoint3D variants. It highlights the limitations of applying traditional 2D-centric UDA strategies directly to CLIP-based 3D recognition. The results of Table 2 suggest that while conventional UDA provides some benefits, more specialized adaptation strategies are necessary to consistently leverage the cross-modal representations of CLIP and achieve robust performance across diverse 3D domains.

7. Effect of α hyperparameter

We investigate the influence of the α hyperparameter in the total loss function (*Eq. 13 of main paper*) in our proposed CLIPoint3D. As shown in Fig. 4, varying α affects the trade-off between different components of the loss and consequently the adaptation performance. For PointDA-10, increasing α leads to a steady improvement, indicating that a higher weight on the corresponding loss term better guides the model for cross-domain alignment. In contrast, GraspNetPC-10 exhibits a more varied trend, with performance peaking at intermediate and higher α values, suggesting that overly small or excessively large weighting can underutilize certain loss components.

8. Influence of the prompt length

We analyze the effect of the length of shared prompt $\mathbf{q}$ on CLIPoint3D. As shown in Fig. 5, varying $\mathbf{q}$ affects the UDA performances. We choose the vision and both encoder variants for the PointDA-10 and GraspNetPC-10 benchmarks respectively. For PointDA-10, the performance remains relatively stable across different $\mathbf{q}$ values, indicating that CLIPoint3D is robust to moderate changes in prompt length. In contrast, GraspNetPC-10 exhibits slight fluctuations, with intermediate values of $\mathbf{q}$ yielding the best results.

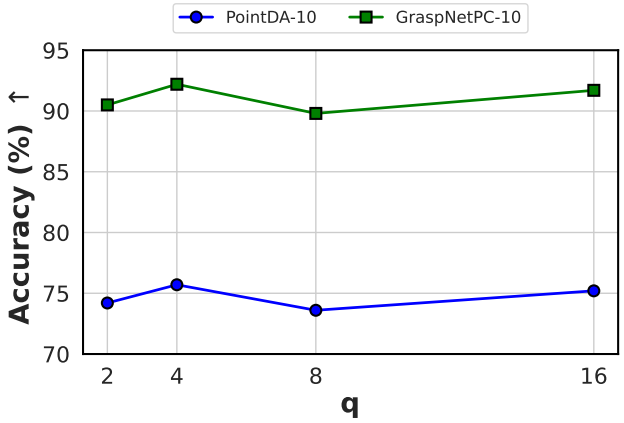


Figure 5. **Effect of varying length of q.** We report the adaptation performances of CLIPoint3D-V and CLIPoint3D-B on PointDA-10 and GraspNetPC-10 datasets respectively.

Table 4. **Effect of using different LLMs on CLIPoint3D-B for attributes generation**. We report the adaptation performances on the GraspNetPC-10 benchmark. and choose a snippet of attributes of class 'drill machine'.

Methods	Attributes Snippet	Syn.→Kin.	Syn→RS.	Kin.→RS.	RS.→Kin.	Avg
Handcrafted	"A point cloud object of"	93.5	81.6	77.9	94.8	87.0
Llama-3.2-3B [8]	"Compact, rectangular structure with a flat surface"	95.5	84.1	84.3	95.6	89.9
Qwen2.5-14B [19]	"Cylindrical main body with handle and trigger"	95.0	87.5	83.2	95.4	90.3
Phi-4 [1]	"Irregular mechanical shape"	95.8	83.2	79.6	92.3	87.7
GPT-5 [12]	"Elongated body with a cylindrical or pistol-like grip"	**96.5**	**89.3**	**86.8**	**96.2**	**92.2**

9. Impact of CLIP variants

We investigate how different CLIP backbone architectures affect the performance of our CLIPoint3D method and choose the vision variant to study on PointDA-10 benchmark. Table 3 compares results using ViT-B/16, ViT-B/32, and ViT-L/14. Overall, larger frozen ViT backbones tend to provide stronger feature representations, leading to improved domain adaptation performance. Specifically, ViT-L/14 achieves the highest average accuracy, benefiting from its larger model capacity and fine-grained patch representation. ViT-B/16 offers a strong trade-off between efficiency and performance, outperforming ViT-B/32 in most cases despite similar model sizes. It indicates that the choice of backbone has a significant impact on the effectiveness of cross-domain alignment, while showcasing CLIP has the extreme potential of capturing better the nuances of 3D point cloud distributions.

10. Effect of various LLMs

We examine the impact of using different LLMs to generate semantic attributes for our method CLIPoint3D and choose the both encoder variant to study on GraspNetPC-10 benchmark.. Table 4 summarizes the adaptation performance when attributes are derived from GPT-5 [12], Llama-3.2-3B [8], Qwen2.5-14B [19], and Phi-4 [1]. Across all adaptation scenarios, GPT-5 consistently produces the most informative attributes, leading to the highest average performance. While other LLMs such as Llama-3.2-3B and Qwen2.5-14B yield competitive results in certain domain pairs, their overall effectiveness is slightly lower and more variable. The results highlight that the quality and expressiveness of the generated attributes significantly influence cross-domain alignment, emphasizing the importance of selecting a capable LLM for robust 3D domain adaptation.

References

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024. 1, 5

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 2

[4] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020. 2

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35. 1, 3, 4

[6] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2): 581–595, 2024. 3

[7] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *International conference on machine learning*, pages 3809–3820. PMLR, 2021. 1

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1, 5

[9] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic concentration for domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9102–9111, 2021. 1, 3, 4

[10] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018. 1, 3, 4

[11] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *International Conference on Learning Representations*, 2023. 3

[12] OpenAI. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/, 2025. August 7, 2025. 1, 3, 5

Algorithm 1 CLIPoint3D algorithm

Require: Training data: source domain $\mathcal{D}^{\mathcal{S}_l} = \{(P_i^{\mathcal{S}_l}, y_i^{\mathcal{S}_l})\}_{i=1}^{N_{\mathcal{S}_l}} = \{(x_{i,m}^{\mathcal{S}_l}, y_i^{\mathcal{S}_l})\}$ and target domain $\mathcal{D}^{\mathcal{T}_u} = \{P_j^{\mathcal{T}_u}\}_{j=1}^{N_{\mathcal{T}_u}} = \{x_{j,m}^{\mathcal{T}_u}\}$, $\mathcal{E}_t$, $\mathcal{E}_v$ & $\mathcal{E}_{3D}$.

procedure TRAINING OBJECTIVE
  Generate the attributes of set of classes, $\mathcal{C} = \{c_k\}_{k=1}^{K}$, by a `LLM`, and extract $\mathbf{T}^{\text{llm}}$ using Eq.2.
  Generate M 2D projected depth maps for each 3D point cloud sample.
  Initialize a random prompt vector of length l from a Gaussian distribution.
  if $n = 1$ **then** ▷ Given total $\mathcal{N}$ epochs
    for $i \leftarrow 0$ **to** K **do** ▷ Given K iterations
      Generate textual prompts $\mathbf{P}_t(\mathbf{T}^{\text{llm}}, \mathbf{p})$ using Eq.3, and extract textual embeddings $\mathbf{T}$ from $\mathcal{E}_t$.
      Generate source and target visual prompts ($\mathbf{P}_v^{\mathcal{S}}$ & $\mathbf{P}_v^{\mathcal{T}}$) separately using Eq.4.
      Extract visual embeddings $\mathbf{I}_{\mathcal{S}_l}$ and $\mathbf{I}_{\mathcal{T}_u}$ from $\mathcal{E}_v^{\mathcal{S}}$ & $\mathcal{E}_v^{\mathcal{T}}$ respectively.
      Do PEFT adaptation of text / vision / both encoder(s).
      Select the views of minimum entropy using Eq.5 and calculate the final prediction probability of a point cloud sample using Eq.6.
      Calculate $\mathbf{L}_{\text{ce}}$, $\mathbf{L}_{\text{ortho}}$, $\mathbf{L}_{\text{OT}}$ & $\mathbf{L}_{\text{conf}}$ using Eq.13.
      Append the source batch probabilities $p_{\mathcal{S}_l}$ in a list.
    end for
    Save all $p_{\mathcal{S}_l}$ and calculate uncertainty-weighted source class prototypes using Eq.7.
  end if
  for $n \leftarrow 2$ **to** $\mathcal{N}$ **do**
    Repeat the procedure of steps 7-15.
    Calculate $\mathbf{L}_{\text{proto}}$ using the source prototypes from the $(n-1)$-th epoch and Eq.10.
    Calculate $\mathbf{L}_{\text{total}}$ using Eq.13.
  end for
end procedure
procedure INFERENCE
  Consider all test samples of target domain $\mathcal{D}^{\mathcal{T}_u}$ in the source dataloader and calculate `top-1` accuracy by selecting the class of maximum $p_{\mathcal{T}_u}$.
end procedure

[13] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 4

[15] Yuefan Shen, Yanchao Yang, Mi Yan, He Wang, Youyi Zheng, and Leonidas J Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7223–7232, 2022. 2

[16] Mainak Singha, Subhankar Roy, Sarthak Mehrotra, Ankit Jha, Moloud Abdar, Biplab Banerjee, and Elisa Ricci. Fedmvp: Federated multi-modal visual prompt tuning for vision-language models. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 3

[17] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 1

[18] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1, 2

[19] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 1, 5

[20] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8552–8562, 2022. 4

[21] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2639–2650, 2023. 4