Supplementary Materials

A. Additional Implementation Details

Our default encoder, PointBERT [42], is only capable of accepting fixed input sizes, which ranges from 8192 to ∼10k points. As the number of points in on real-world objects are much lesser, a small architectural modification is needed to accommodate them simultaneously. Point-BERT divides the point cloud into overlapping patches. To keep the in-patch (number of points and density) and between-patch (overlap amount, coverage) distribution of the divided point cloud, the number of patches are calculated individually, and a special padding token is used for the remaining patch embeddings, which are masked out in the attention mechanism of the transformer encoder part. This provides a fair baseline, as only true, meaningful points contribute to the final result. These masked patch tokens are also left out of the final global feature pooling to avoid influencing results.

Model	Optimizer	Peak LR	LR Schedule	Weight Decay	Batch Size	Epochs
Point-BERT	AdamW	1e-3	Cosine Decay	0.1	512	250

Table S1. Hyperparameters used for training.

All models are bootstrapped from pre-trained weights rather than trained from scratch. This follows the proceedings of the state-of-the-art method ULIP-2 [41]. Point-BERT [42] is loaded with the checkpoint obtained after its masked-reconstruction pre-training phase. The hyperparameters used for training are described in Table S1

B. Class-Wise Comparison:

To better understand how our method performs across different object categories, we analyze the class-wise top-1 accuracy on the nuScenes-triplets validation split. Figure S1 compares the per-class classification accuracy with the corresponding class frequencies in the training set. As is typical for autonomous driving data, the class distribution is heavily long-tailed, with `car` making up over 40% of all instances, while classes like `bicycle` or `construction vehicle` appear in fewer than 1% of examples.

Interestingly, both ULIP-2 variants exhibit only a weak dependence on class frequency, suggesting that point cloud–language alignment in general mitigates some of the long-tail effects common in supervised training. However, important differences remain. ULIP-2 trained only on Objaverse achieves good performance on semantically rich categories but collapses on outdoor-specific classes such as `construction vehicle` and `trailer`. Conversely, ULIP-2 trained on nuScenes adapts to the outdoor domain but lags behind on less frequent categories. In contrast, BlendCLIP combines the strengths of both: it maintains competitive accuracy on common classes while significantly boosting performance on rare ones (e.g., `bicycle`, `construction vehicle`). This balanced behavior indicates that our curriculum-based mixing strategy effectively transfers semantic richness from synthetic data while adapting to LiDAR-specific sensor characteristics, without requiring explicit frequency reweighting.

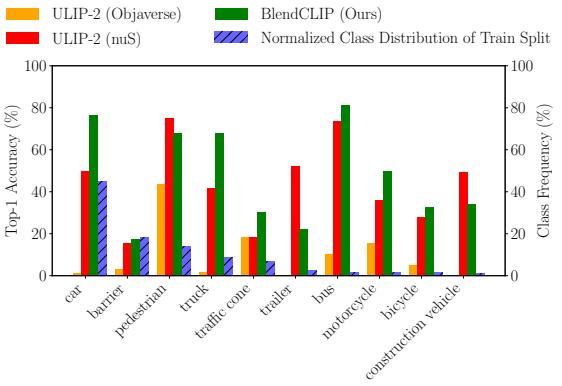

Figure S1. Per-class top-1 zero-shot accuracy on the nuScenes-triplets validation set, with class frequencies shown in blue. Compared are ULIP-2 (Objaverse), ULIP-2 (nuScenes), and BlendCLIP (ours).

Table S2 extends this comparison to a broader set of baselines. BlendCLIP achieves the best or second-best accuracy in nearly all categories, confirming the balanced gains suggested by the histogram. Notably, it delivers large improvements on common classes such as `car` and `truck`, while also setting new best results on rarer categories like `motorcycle` and `bicycle`. Other baselines occasionally peak on individual classes (e.g., PointCLIP on `trailer`, ULIP-2 on `pedestrian`), but none match BlendCLIP's overall consistency.

C. Additional Datasets

Our method is additionally evaluated on ScanObjectNN [33], an object-centric real-world dataset collected with RGB-D sensors across 15 object categories, and ModelNet40 [38], which is a well-established synthetic CAD object model dataset over 40 categories.

Table S3 shows our method performs comparably on zero-shot classification benchmarks to current state-of-the-art methods. The performance on ScanObjectNN highlights that our method is not restricted to only outdoors LiDAR scenarios: it can be used for indoor use cases, where object categories and capture quality differ.

Method	Pre-train Dataset	Avg.	Car	Barrier	Ped.	Truck	T.C.	Trailer	Bus	Motor.	Bicycle	C.V.
Frequency (%)		–	44.9	18.1	13.7	8.6	7.0	2.3	1.7	1.4	1.3	1.1
PointClip* [44]	nuS	21.1	18.8	2.1	74.0	0.0	29.7	**57.0**	5.5	4.5	17.9	1.9
Clip2Point* [16]	nuS	24.0	26.7	10.5	45.2	16.8	34.2	13.9	51.2	5.7	15.8	20.0
CLIP^{2}* [43]	nuS	28.7	41.9	17.3	40.3	41.3	**35.3**	20.6	22.5	22.4	21.1	24.8
LidarCLIP† [14]	nuS	10.1	13.7	**20.8**	15.7	5.5	0.1	0.0	0.5	43.0	0.0	1.8
ULIP-2 [41]	nuS	43.8	49.7	15.4	**74.9**	41.6	18.2	52.0	73.6	36.1	28.0	**49.1**
ULIP-2† [41]	Objaverse [7]	9.9	1.1	3.1	43.6	1.8	18.4	0.2	10.0	15.7	5.1	0.0
BlendCLIP (Ours)	Objaverse [7]+nuS	**47.9**	**76.2**	17.3	67.9	**68.0**	30.1	22.1	**81.2**	**49.7**	**32.8**	34.2

Table S2. Per-class Top-1 zero-shot classification accuracy (%) on nuScenes. Classes are ordered by frequency of occurrence (shown in the first row). Ped: pedestrian, Motor.: motorcycle, T.C.: traffic cone, C.V.: construction vehicle. ∗: reported by authors of CLIP^{2} [43]. †: reproduced by us. Best results in bold, second-best are underlined.

Method	Pre-train Dataset	ScanObjectNN [33]	ModelNet40 [38]
PointCLIP* [44]	2D inference	10.5 / 30.6	19.3 / 34.8
PointCLIP v2* [48]		42.2 / 74.5	63.6 / 85.0
CLIP2Point* [16]	ShapeNet	25.5 / 59.4	49.5 / 81.2
OpenShape-PointBERT* [21]	Ensembled†	56.7 / **88.6**	**84.4** / 98.0
ULIP-2 [41]	Objaverse [7]	62.3 / 85.9	82.4 / 97.6
BlendCLIP (Ours)	Objaverse [7] + nuScenes-triplets	**63.2** / 86.1	82.5 / **98.3**

Table S3. Object-wise zero-shot classification accuracies (top-1/top-5, %) for different methods on ScanObjectNN and ModelNet40. Higher is better. Best scores are highlighted in bold, second-best are underlined. ∗: Reported by authors of OpenShape. †: Generated by authors of OpenShape.

D. Effect of Data Mixing Ratio

In Table S4, we investigate how the maximum outdoor mixing ratio $r_{\max}$ affects zero-shot performance across domains. As expected, increasing the proportion of outdoor LiDAR crops leads to a consistent decrease in accuracy on Objaverse-LVIS. This is likely due to the reduced presence of high-resolution CAD objects in the training batches, which limits the model's exposure to long-tailed fine-grained categories.

r_{max}	Objaverse-LVIS [7] (%) ↑	nuS (%) ↑	TruckS (%) ↑
10	**47.9**	47.7	24.9
20	47.3	47.8	23.3
30	46.7	**48.0**	**25.5**
40	45.7	46.4	23
50	45	45.2	22.2

Table S4. Effect of dataset mixing ratio r_{max} (%) on average class-wise Top-1 zero-shot classification accuracy (%). Higher indicates better performance.

On nuScenes and TruckScenes triplets, accuracy increases as $r_{\max}$ grows, peaking at 30% before declining. This indicates that while some real-world supervision is essential for bridging the domain gap, too much can erode the semantic generalization learned from CAD. Overall, $r_{\max} = 30\%$ provides the best trade-off, yielding the strongest outdoor performance with only a moderate drop on Objaverse-LVIS. These results validate our curriculum-style mixing strategy and confirm that only a modest proportion of real-world supervision is sufficient to drive effective domain transfer without sacrificing generalization to unseen or synthetic classes.

E. Effect of VLM for Caption Generation

For caption generation of outdoor triplets, prior works such as ULIP-2 [41] rely on BLIP-2 with OPT [45] as the language model head. In this experiment, we investigate replacing the VLM with the recently released Gemma 3 4B [31], an open-weight, instruction-tuned language model that runs efficiently on local hardware. Its explicit support for system prompts allows controlled caption style and content. Specifically, the following system prompt is used to guide Gemma's output during caption generation:

> "You are to caption images. Capture as much detail and semantic information as possible. Only describe one object, which is the largest one in the image. Ignore the background. Leave out image quality description from the caption."

This prompt ensures object-centric captions, avoids visual noise from background clutter, and discourages style-based hallucinations such as "a blurry photo." In Table S5, we compare captions generated by BLIP-2-OPT [20] and Gemma 3 on the same input crop, shown in Figure S2. While BLIP-2 fails to correctly identify the object and offers vague descriptions, Gemma 3 provides a detailed and semantically rich caption consistent with the vehicle in the image.

Table S6 reports quantitative comparison between the BLIP-2 versus Gemma 3. Across all benchmarks, the differences are within $\pm 0.5\%$ for LVIS and nuScenes, and within

Figure S2. Example image crop used for captioning evaluation.

Model	Caption
BLIP-2-OPT 6.7B [20]	a photo of traffic cones in front of a **white** car
Google Gemma 3 4B [31]	The image shows a **beige** sedan with a **rounded** roofline, a **black** trunk, and **dark tinted** windows. It has a **chrome** strip along the bottom and **black** wheels with **silver** rims.

Table S5. Comparison of generated captions. Underlined = subject; **bold** = fine-grained descriptors.

$\pm 0.7\%$ for TruckScenes. This consistency indicates that the choice of captioning VLM has only marginal influence on downstream performance. We therefore attribute the majority of the observed gains to our curriculum-based data mixing strategy rather than to the specific caption generator employed.

VLM	Objaverse-LVIS [7] (%) ↑	nuS (%) ↑	TruckS (%) ↑
BLIP-2 [20]	46.7 / 75.2	48.0 / 79.6	25.5 / 63.5
Gemma 3 4B [31]	46.4 / 75.1	48.5 / 83.1	24.8 / 63.8

Table S6. Zero-shot top-1 / top-5 classification accuracy (%) with captions generated by BLIP-2 or Gemma 3. Performance remains stable across VLMs.

F. Viewpoint-Aware Occlusion Synthesis

To simulate the geometric gap between full CAD models and partial real-world LiDAR scans at training time, we apply a lightweight visibility filter that generates partial CAD views mimicking realistic occlusions. Referred to as occlusion augmentation, we hypothesize this encourages the encoder to learn from incomplete observations, as in real-world sensing.

Our approach uses the Hidden Point Removal (HPR) operator [19] to generate viewpoint-conditioned partial point clouds from synthetic objects. HPR approximates visibility from a given virtual viewpoint without requiring expensive ray tracing. To ensure the model learns a diverse range of visibility patterns, we randomize the virtual viewpoint for each instance, sampling its position on a spherical shell around the object. This placement simulates plausible LiDAR sensor viewpoints from varying angles and distances, enhancing geometric diversity across batches. An example is shown in Figure S3.

(a) Before HPR (b) After HPR

Figure S3. Visualization of occlusion-aware augmentation using the HPR operator [19]. The raw synthetic point cloud (a) is filtered from a sampled viewpoint to retain only the visible surface (b), simulating realistic LiDAR-style occlusion.

While the HPR operator requires setting an inversion-sphere radius, we deliberately use a generously scaled radius to preserve fine-grained but partially concave features such as undercarriage cavities and wheel wells. This ensures the encoder retains access to informative geometry while still learning realistic visibility cues. The result is an occlusion-aware sampling of CAD points that attempts to better match the sparsity and perspective-induced occlusions found in real LiDAR data.

Dataset Mixing	Occlusion Augmentation	Objaverse-LVIS [7]	nuS	TruckS
✗	✗	**47.8**	9.9	8.5
✗	✓	45.8	8.9	10.9
✓	✗	46.4	**48.5**	**24.8**
✓	✓	43.4	48.3	24.1

Table S7. Effect of occlusion augmentation on top-1 zero-shot classification accuracy. Higher indicates better performance.

Table S7 compares the effectiveness of occlusion augmentation towards zero-shot classification. Occlusion augmentation alone provides small gains on TruckScenes but has limited effect elsewhere. When combined with mixing, performance slightly drops compared to mixing alone (–0.2% on nuScenes, –0.7% on TruckScenes), likely due to added geometric variability that does not fully match real-world occlusion patterns. These results suggest that while viewpoint-aware sparsity may help in principle, domain adaptation benefits more from capturing distributional diversity and semantic alignment than from geometric realism alone.

G. Qualitative Retrieval Results

We show that our method not only performs well on general outdoor classes but can pinpoint fine-grained object attributes. There are cases where knowing the exact type of the instance is essential for correct decision-making. One such case is recognizing police officers controlling traffic or pedestrians with mobility impairments whose moving trajectories could be wildly different compared to other pedestrians.

To show this capability, we compute the similarities between the prompt and the point clouds, as described in Section 4.2, and then select the top-5 most similar examples. From the top-5 retrieved samples, we present the images of three representative instances for visualization, in Figure S4. For illustration, our qualitative analysis focuses on categories where fine-grained recognition is particularly relevant. Our choices fell to `police officer`, `wheelchair`, `stroller`, `motorcyclist`, `bicyclist`, and `scooter`. Some retrieved samples (e.g., for `stroller`) are not exact matches but semantically similar, revealing the model's ability to surface rare, functionally relevant instances beyond the labeled taxonomy. We believe that by recognizing these rare cases, a perception system could create better educated decisions, making autonomous driving potentially safer.

In some cases, the same instance appears multiple times, as the retrieval process may independently select different point clouds of that instance. Since our method does not exploit temporal information, this recurrence demonstrates consistency: the model reliably categorizes the same instance across different times and viewpoints.

(a) `police officer`

(b) `wheelchair`

(c) `stroller`

(d) `motorcyclist`

(e) `bicyclist`

(f) `scooter`

Figure S4. Selected examples from top-5 retrieved samples for BlendCLIP. The images are shown only for visualization. The similarities were computed only with the embeddings of the point clouds. Every image has a different point cloud associated with it.