OpenDlign: Enhancing Open-World 3D Learning with Depth-Aligned Images

Ye Mao **Junpeng Jing** **Krystian Mikolajczyk**
Imperial College London
https://yebulabula.github.io/OpenDlign/
{ye.mao21, j.jing23, k.mikolajczyk}@imperial.ac.uk

Abstract

Recent advances in Vision and Language Models (VLMs) have improved open-world 3D representation, facilitating 3D zero-shot capability in unseen categories. Existing open-world methods pre-train an extra 3D encoder to align features from 3D data (e.g., depth maps or point clouds) with CAD-rendered images and corresponding texts. However, the limited color and texture variations in CAD images can compromise the alignment robustness. Furthermore, the volume discrepancy between pre-training datasets of the 3D encoder and VLM leads to sub-optimal 2D to 3D knowledge transfer. To overcome these issues, we propose OpenDlign, a novel framework for learning open-world 3D representations, that leverages depth-aligned images generated from point cloud-projected depth maps. Unlike CAD-rendered images, our generated images provide rich, realistic color and texture diversity while preserving geometric and semantic consistency with the depth maps. OpenDlign also optimizes depth map projection and integrates depth-specific text prompts, improving 2D VLM knowledge adaptation for 3D learning efficient fine-tuning. Experimental results show that OpenDlign significantly outperforms existing benchmarks in zero-shot and few-shot 3D tasks, exceeding prior scores by 8.0% on ModelNet40 and 16.4% on OmniObject3D with just 6 million tuned parameters. Moreover, integrating generated depth-aligned images into existing 3D learning pipelines consistently improves their performance.

1 Introduction

3D understanding, which involves tasks such as point cloud classification and 3D object detection, is pivotal for advancing augmented/virtual reality [1; 2], autonomous vehicles [3; 4], and robotics [5; 6]. Traditional 3D models [7; 8; 9; 10; 11; 12; 13] are closed-world, which can only recognize pre-defined categories and struggle with 'unseen' ones. The emergence of Vision-Language Models (VLMs) like CLIP [14], renowned for their success in identifying 'unseen' categories in 2D images through open-world representation learning [15; 16; 17; 18], has sparked interest in applying these models to develop robust open-world 3D representations for 3D vision tasks.

Existing open-world 3D learning methods can be categorized into depth-based and point-based methods. Depth-based methods [19; 20; 21] project point clouds into multi-view depth maps and employ the pre-trained CLIP image encoder for 3D representations. However, this process encounters a domain gap because CLIP is primarily trained with RGB images rather than depth maps. To bridge this gap, methods like [21] incorporate an additional depth encoder and utilize contrastive learning to align depth features from this encoder with image and text features from pre-trained CLIP encoders, as illustrated in Fig. 1(a). The images used here, specifically rendered from CAD models for feature alignment, are not employed in the zero-shot inference phase. Point-based methods [22; 23; 24; 25; 26; 27] directly learn 3D representations from point clouds, avoiding the latency of

Preprint. Under review.

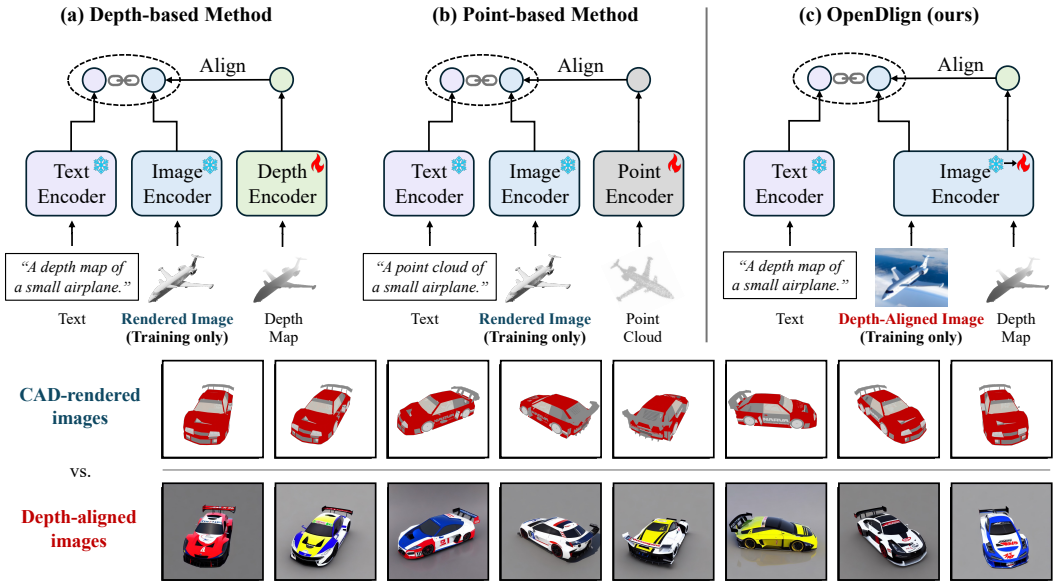

Figure 1: **Top:** OpenDlign vs. Conventional Open-World 3D Learning Frameworks: OpenDlign enhances multimodal alignment using depth-aligned images, providing more detailed geometric and semantic information along with enhanced color and texture compared to previously used rendered images. It refines 3D representation by fine-tuning the CLIP image encoder directly, eliminating the extra encoder pre-training required by other methods. Note that both rendered and depth-aligned images are used exclusively for learning alignment. **Bottom:** Visual comparison between CAD-rendered and corresponding depth-aligned multi-view images.

depth map projection. However, due to the inherent data format differences between images and point clouds, these methods also need an additional point encoder for extracting 3D features, akin to depth-based methods (See Fig. 1(b)). Thus, aligning 3D data (e.g., depth maps or point clouds) with the image-text modalities pre-aligned by CLIP is a standard step in current 3D open-world methods.

Depth-based and point-based methods encounter two primary challenges in the alignment process. First, the CAD-rendered images used for aligning 3D data typically display consistent color and texture styles across various views. Over-aligning with these low-diversity images compromises the generalizability of learned 3D representations. Secondly, the 3D datasets used for encoder pre-training, like ShapeNet [28] and Objaverse [29], contain less than **1 million** synthetic 3D objects, significantly smaller than the DFN5B [30] and LAION-5B [31] datasets with **5 billion** images used to train the cutting-edge CLIPs. This data volume disparity, which is due to the high cost of 3D data acquisition, results in the sub-optimal transfer of CLIP's knowledge to 3D representations. While fine-tuning CLIP's encoders yields more direct knowledge transfer, it restricts the input to depth maps. Unfortunately, 3D representations from depth maps still underperform in downstream 3D tasks compared to those from point clouds, due to two factors: (1) The absence of a robust projection method for creating dense depth maps with smooth contours from point clouds. (2) The current widely used CLIP text prompt templates are tailored for matching with RGB images, not depth maps.

To address these challenges, this paper proposes OpenDlign, a novel framework that learns **Open**-world 3D representations via aligning multi-view depth maps projected from point clouds with **D**epth-a**lign**ed images produced by a generative model [32]. These images offer enhanced color and texture diversity compared to CAD-rendered images while maintaining geometric and semantic consistency with the depth maps (See Fig. 1). Additionally, as shown in Fig. 1(c), OpenDlign fine-tunes the CLIP image encoder rather than pre-training a separate depth encoder, thus maximally adapting CLIP's existing knowledge for effective 3D learning, even with a limited 3D dataset. Specifically, fine-tuning is limited to the attention layers of the last transformer block, comprising just **6 million** parameters. Moreover, OpenDlign employs a new projection pipeline to generate dense depth maps with clear contours. For zero-shot inference, OpenDlign employs depth-specific text prompts and a logit aggregation method, emphasizing depth-related features and combining results from various viewpoint depth maps. Experimental results show that OpenDlign greatly surpasses the prior state-of-the-art, pre-trained on ShapeNet [28], with accuracy gains of 8.0% on ModelNet40

and 16.4% on OmniObject3D, the largest real-world 3D shape dataset. Notably, using realistic depth-aligned images significantly boosts the performance of existing SOTA models, like those pretrained on ShapeNet or 3D Ensemble datasets [24]. This consistent improvement across all benchmarks highlights the versatility of depth-aligned images in any 3D open-world learning pipeline.

The main contributions of this paper are outlined as follows:

- We propose a multimodal alignment framework that aligns features from depth maps and depth-aligned images to learn a unified depth map, image, and text representation.
- We develop a contour-aware projection pipeline to produce dense and contour-preserving multi-view depth maps from point clouds.
- We introduce depth-specific text prompt templates for zero-shot inference to accurately capture both the semantic and visual traits in depth maps.
- We design a logit aggregation strategy that derives final 3D representations from both CLIP and OpenDlign visual encoders, reducing catastrophic forgetting in alignment.

2 Related Work

2.1 Open-World 3D Representation Learning

Vision and Language models such as CLIP [14] have revolutionized 2D representation learning in open-world settings through contrastive learning with large-scale image-text pairs [33; 34; 35; 36]. Building on this, recent studies have adapted CLIP for 3D representation learning, achiving impressive performance in diverse 3D zero-shot tasks [24; 25].

PointCLIP [20], as a pioneering study, utilizes the CLIP image encoder for extracting 3D representations from depth maps of point clouds, achieving zero-shot recognition by aligning with text embeddings of semantic categories. To address CLIP's training bias towards RGB images, Zhu *et al.* [19] introduced GPT-generated 3D-specific prompts and a denser depth map projection, while CLIP2Point [21] pre-trains a depth encoder for closer alignment with CLIP's encoders. These methods derive representations from depth maps with noisy contours, causing a loss of key shape features needed for precise recognition. Moreover, their reliance on either natural image text prompts or depth-specific prompts generated by GPT-3 [37] for certain categories highlights a lack of versatility in handling diverse 3D contexts. Alternative methods [23; 23; 24; 25; 27] avoid depth map projection by directly aligning point clouds, images, and text using specialized 3D encoders. By scaling up the dataset and encoder sizes, these methods show promise in diverse 3D tasks. However, these methods are limited by their reliance on CAD-rendered images, which have limited texture diversity across views, leading to less generalizable representations. Additionally, the smaller volume of 3D datasets compared to CLIP's training data hinders effective knowledge transfer to point cloud encoders.

In this paper, we substitute rendered images with AI-generated, depth-aligned images to enhance texture diversity. We also fine-tune the CLIP image encoder for 3D representation learning instead of training a new 3D encoder from scratch, reducing the reliance on large 3D datasets.

2.2 Continual Learning in CLIP Fine-Tuning

Continual Learning (CL) in CLIP aims to mitigate catastrophic forgetting [38], ensuring retention of zero-shot capabilities across varied data distributions while fine-tuning to new tasks. CL methods fall into three categories: adaptive-plasticity methods [39; 40; 41; 42; 43; 44], replay methods [45; 46; 47], and architecture-based methods [48; 49]. Adaptive-plasticity methods limit the plasticity of the essential model parameters for past tasks during fine-tuning. For instance, the IMM-Mean [44] method achieves CL by simply averaging parameters of pre-trained and fine-tuned models for inference, although its efficacy might be limited for complex tasks [50]. Replay methods leverage stored exemplars to enable CLIP to recall previously learned knowledge, while they encounter scalability challenges. Without relying on exemplars, architecture-based CL methods dynamically adjust the model's architecture to accommodate new information without losing existing knowledge [50]. In this study, we align the depth map with the RGB image by freezing the pre-trained CLIP encoder weights and incorporating a trainable transformer-based branch for encoding depth maps, adhering to architecture-based principles. Inspired by IMM-Mean [44], we use pre-trained and fine-tuned model weights to compute classification logits for multi-view depth maps.

3 Methodology

Fig. 2 illustrates the OpenDlign framework, which learns effective open-world 3D representations by aligning embeddings from projected depth maps and depth-aligned images. Initially, a contour-aware projection method is employed to create shape-preserved, dense depth maps from point clouds. These maps then guide a generative model to produce depth-aligned images with rich color and texture diversity. OpenDlign then uses contrastive learning to align features between depth maps and generated images by fine-tuning a transformer block linked to the CLIP image encoder. This step enables the extraction of robust embeddings from 'unseen' multi-view depth maps at test time, using both fine-tuned and pre-trained states of the image encoder. These embeddings are matched with depth-specific text embeddings, which encode the depth maps' semantic and visual traits, to compute logits for each viewpoint and aggregate these logits to enable zero-shot classification. Alternatively, these embeddings can be refined using a logistic regressor for few-shot classification.

3.1 Contour-Aware Depth Map Projection

The contour-aware projection method transforms the input point cloud into multi-view depth maps with clear contours. Inspired by the pipeline in [19], this method involves four main steps: Quantize, Densify, Smooth, and Squeeze.

In the **Quantize** step, for the i^{th} view of point cloud P_i, the 3D coordinates $(x, y, z) \in P_i$ are normalized to $[0, 1]$ and mapped onto a discrete grid $G \in \mathbb{R}^{H \times W \times B}$, where H and W correspond to the dimensions required by the CLIP image encoder, and B is a pre-defined depth dimension. Next, the **Densify** step enhances G by updating each voxel to the maximum value within its $7 \times 7 \times 7$ neighborhood, yielding a denser map G'. Subsequently, the **Smooth** step applies bilateral filtering to each voxel v_i in G', adjusting its intensity I_{v_i} to I'_{v_i} using:

$$I'_{v_i} = \frac{1}{W_v} \sum_{v_j \in S} G_{\sigma_1}(\|v_i - v_j\|) G_{\sigma_2}(|I_{v_i} - I_{v_j}|) I_{v_j} \tag{1}$$

where $W_v = \sum_{v_j \in S} G_{\sigma_1}(\|v_i - v_j\|) G_{\sigma_2}(|I_{v_i} - I_{v_j}|)$ is the normalization factor that ensures voxel weights sum to 1.0. The Gaussian functions G_{σ_1} and G_{σ_2} adjust the influence of each neighboring voxel v_j within the $5 \times 5 \times 5$ kernel from set S around v_i, based on spatial and intensity differences, enhancing contour sharpness and reducing jagged edges in G'. Finally, the **Squeeze** step applies the minimal pooling on the depth channel of the smoothed G', then triples the output to mimic RGB intensity, producing the final depth map $D \in \mathbb{R}^{H \times W \times 3}$.

3.2 Depth-Aligned Image Generation

We generated **524,700** depth-aligned images from ShapeNet [28], one of the leading public 3D CAD datasets containing around 52,470 models, each annotated with semantic metadata. To align with prior experimental protocols [24; 23], we sampled a point cloud of 10,000 points from each model, projecting these onto 10 contour-aware depth maps. A conditional image generative model (ControlNet v1.1 [32]) then produced depth-aligned images for each map (D), using $1 - D$ and the model's metadata as conditions. This approach ensures that the images remain consistent with the depth maps both geometrically and semantically, while also adding texture diversity across different views. The conditioning of ControlNet utilizes $1 - D$ instead of D because it is predominantly pre-trained on depth images, in which brighter regions indicate closer proximity. The supplemental material details the positive and negative prompts used in ControlNet to achieve high-fidelity and noise-free depth-aligned image generation.

3.3 Multimodal Representation Alignment

OpenDlign aligns representations from multi-view depth maps and depth-aligned images by fine-tuning a transformer block that is residually connected to the final block of the pre-trained CLIP image encoder, using contrastive learning. As CLIP pre-training already aligns image and text modalities, OpenDlign implicitly aligns depth maps with the shared image and text space.

Multimodal Feature Extraction. Given a 3D point cloud input, let $D = \{D_i\}_{i=1}^{N}$ represent the set of its N projected depth map views, and $R = \{R_i\}_{i=1}^{N}$ the corresponding set of depth-aligned

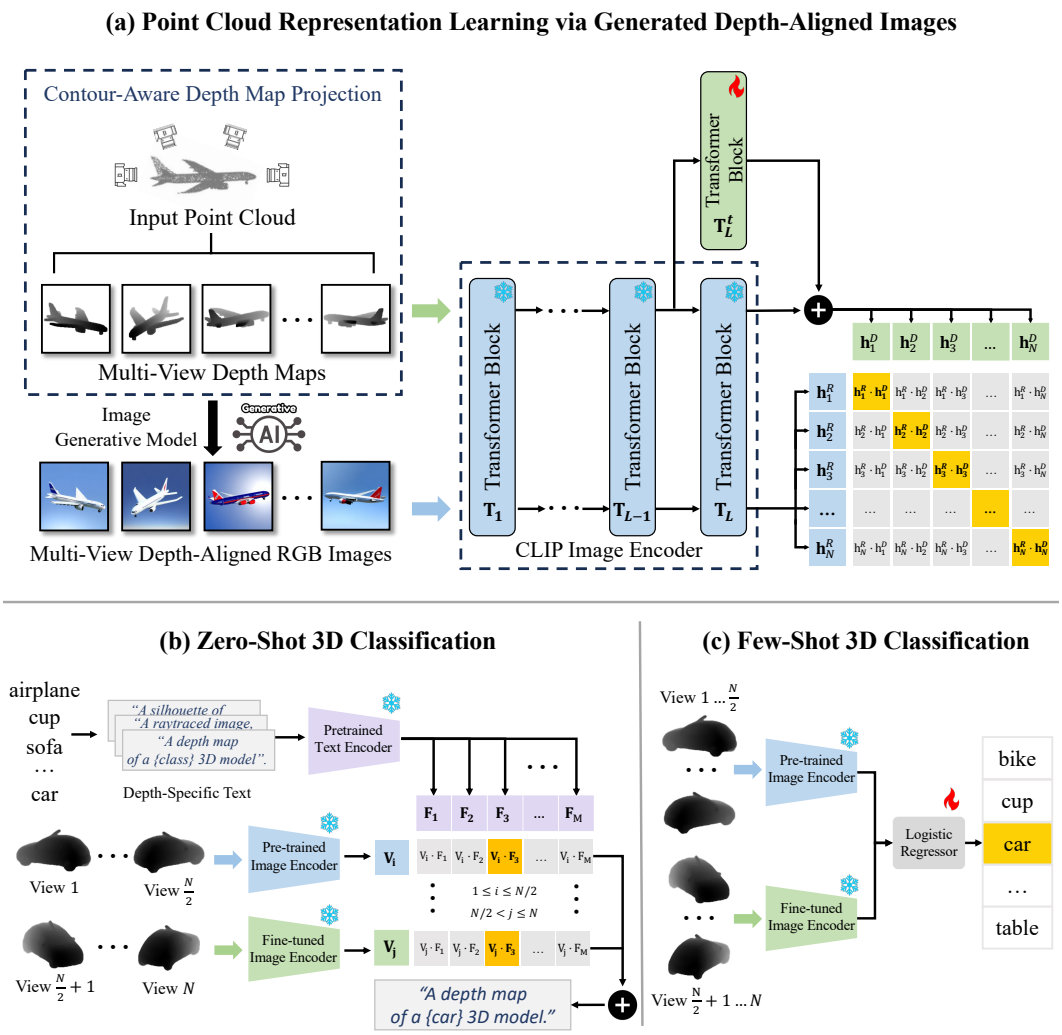

Figure 2: **Overview of OpenDlign.** In **(a)**, OpenDlign converts point clouds into multi-view depth maps using a contour-aware projection, which then helps generate depth-aligned RGB images with diverse textures, geometrically and semantically aligned with the maps. A transformer block, residually connected to the CLIP image encoder, is fine-tuned to align depth maps with depth-aligned images for robust 3D representation. For zero-shot classification **(b)**, OpenDlign aggregates multi-view logits from both pre-trained and fine-tuned encoders for label prediction and for few-shot classification **(c)**, it employs a logistic regressor trained on multi-view features from the encoders.

images. Each image R_i is encoded through L layers of a pre-trained CLIP image encoder, $\{\mathrm{T}_l(\cdot)\}_{l=1}^{L}$, to obtain feature representations $I_i^R = \mathrm{T}_{1\ldots L}(R_i)$. Each depth map D_i is processed up to layer T_{L-1}, obtaining preliminary features $\mathrm{T}_{1\ldots L-1}(D_i)$. Subsequently, these features are passed through the frozen layer T_L and its trainable counterpart T_L^t, yielding the feature for the i_{th} depth map view $I_i^D = \mathrm{T}_{1\ldots L}(D_i) + \mathrm{T}_L^t(\mathrm{T}_{1\ldots L-1}(D_i))$. Inspired by [17], only the layers for spatial interaction in T_L^t (i.e., attention layers) are trainable. The final feature vectors for multi-view depth maps D and depth-aligned images R are $\mathbf{h}^D = \frac{1}{N}\sum_{i=1}^{N}\|I_i^D\|$ and $\mathbf{h}^R = \frac{1}{N}\sum_{i=1}^{N}\|I_i^R\|$, respectively.

Loss Functions. The alignment of $\mathbf{h}^D$ and $\mathbf{h}^R$ is achieved by minimizing a composite loss function, comprising the contrastive loss $\mathcal{L}_{\text{cont}}$ and the feature distance loss $\mathcal{L}_{\text{dist}}$, defined as:

$$\mathcal{L}_{\text{total}} = \underbrace{\sum_{(i,j)} -\frac{1}{2}\log\frac{\exp\left(\mathbf{h}_i^D\mathbf{h}_j^R/\tau\right)}{\sum_k \exp\left(\mathbf{h}_i^D\mathbf{h}_k^R/\tau\right)} - \frac{1}{2}\log\frac{\exp\left(\mathbf{h}_i^D\mathbf{h}_j^R/\tau\right)}{\sum_k \exp\left(\mathbf{h}_k^D\mathbf{h}_j^R/\tau\right)}}_{\mathcal{L}_{\text{cont}}} + \underbrace{\sum_{(i,j)} \|\mathbf{h}_i^D - \mathbf{h}_j^R\|_2}_{\mathcal{L}_{\text{dist}}} \quad (2)$$

In each training batch, $(\mathbf{h}_i^D, \mathbf{h}_j^R)$ represents a positive pair and $k \neq i, j$. Here, τ is a learnable temperature parameter, similar to CLIP [14].

3.4 3D Zero-Shot Transfer

The alignment between depth maps and depth-aligned RGB images facilitates 3D zero-shot classification by aggregating multi-view classification logits. Each logit represents the similarity between features of a single-view depth map and text features specific to category candidates.

Depth-Specific Text Generation. We generate 80 depth-specific text prompt templates based on 80 ImageNet zero-shot recognition prompts[1], integrating keywords such as "depth map", "white background image", "raytraced image", and "silhouette of [CLASS]". These keywords guide OpenDlign to target depth-related features, such as the distance of object surfaces from a viewpoint. To identify these keywords, we use the CLIP-Interrogator tool [51] to analyze depth maps from ShapeNet [28], seeking text prompts that best match their visual features. The 10 most recurring prompts from this analysis are chosen as our essential keywords. In zero-shot inference, we employ our depth-specific templates to generate 80 text descriptions for each label l. These descriptions $\{t_i\}_{i=1}^{80}$ are encoded by a texture encoder $F(\cdot)$, normalized, and then merged into a unified text feature $\bar{F}_l$ via average pooling, calculated as $\frac{1}{80}\sum_{i=1}^{80}\|F(t_i)\|$.

Multi-View Logits Aggregation. To calculate classification logits, we first gather visual features from multi-view depth maps $\{V_i\}_{i=1}^{N}$, aiming to align with depth-specific text features of M candidate labels $\mathbf{F} = \{F_i\}_{i=1}^{M}$. The feature extraction utilizes a dual-encoder strategy: the first half of the views $\{V_i\}_{i=1}^{N/2}$utilize a pre-trained CLIP image encoder, while the second half of views $\{V_i\}_{i=N/2+1}^{N}$ employs a fine-tuned encoder. The strategy ensures that OpenDlign maintains its capability to recognize previously identifiable depth maps after learning multimodal alignment via fine-tuning. As shown in Fig. 2(b), the logit for a single depth map view is the product of V_i and $\mathbf{F}$, with the overall classification logit being the sum of logits across all views, calculated as $\sum_{i=1}^{N} V_i\mathbf{F}^T$.

4 Experiments

4.1 Zero-Shot 3D Classification

We first evaluated OpenDlign under the zero-shot shape classification task on three benchmark datasets: ModelNet40 [52], ScanObjectNN [53], and OmniObject3D [54]. ModelNet40 offers synthetic 3D CAD models in 40 categories. ScanObjectNN provides real-scanned objects in 15 categories from OBJ_ONLY version. OmniObject3D, the largest, includes 5,911 real-scanned objects in 216 categories, well-suited for fine-grained, real-world classification evaluation. Point cloud sizes are 10,000 points for ModelNet40, 2,048 for ScanObjectNN, and 4,096 for OmniObject3D. OpenDlign was compared against existing methods, including three depth-based methods: PointCLIP [20], PointCLIP V2 [19], and CLIP2Point [21], and three point-based methods: ULIP [23], OpenShape [24], and TAMM [27]. Additionally, we improved the OpenShape and TAMM models by retraining them with depth-aligned and CAD-rendered images from an integrated dataset provided by OpenShape, which combines four distinct collections: Objaverse [29], ShapeNet [24], 3D-Future [55], and ABO [56]. Our aim was to investigate if depth-aligned images consistently enhance the performance of existing 3D open-world methods. Moreover, we evaluated OpenDlign's scalability by training it with various CLIP variants to adapt to the complexity of pre-trained image-text encoders.

Table 1 shows OpenDlign substantially outperforms existing methods trained on ShapeNet on three benchmarks, exceeding the previous best, TAMM-SparseConv trained on ShapeNet, by margins of 8.0% on ModelNet40, 1.6% on ScanObjectNN, and 16.4% on OmniObject3D in top-1 accuracy. OpenDlign also greatly exceeds the leading depth-based method, PointCLIP V2—by 19% on ModelNet40 and 27.4% on OmniObject3D. Significantly, OpenDlign outshines all methods pre-trained on the ensemble dataset in the ScanObject3D benchmark. Moreover, OpenDlign's performance scales linearly with the complexity of CLIP variants, surpassing most of the baseline models on ModelNet40 and OmniObject3D benchmarks, even when employing the light ViT-B-16 CLIP model. Moreover, the use of depth-aligned images consistently boosts the performance of OpenShape and

[1]Text Prompts for ImageNet: ImageNet Prompt Engineering.

Table 1: **Zero-shot classification results on ModelNet40 [52], ScanObjectNN [53] and OmniObject3D[54].** Best: bolded. Second-best: underlined.

Training Source	3D Open-World Methods	CLIP Variant	ModelNet40 [52]			ScanObjectNN [53]			OmniObject3D[54]		
			Top1	Top3	Top5	Top1	Top3	Top5	Top1	Top3	Top5
2D inferences No Training	PointCLIP [20]	ResNet-50	19.3	28.6	34.8	10.5	20.8	30.6	0.3	1.0	1.8
	PointCLIP V2 [19]	ViT-B-16	63.6	77.9	85.0	42.2	63.3	74.5	3.9	9.6	14.4
ShapeNet	CLIP2Point [21]	ViT-B-32	49.5	71.3	81.2	25.5	44.6	59.4	1.4	3.7	7.1
	ULIP-PointBERT [23]	SLIP [57]	60.4	79.0	84.4	51.5	71.1	80.2	8.4	15.2	19.7
	OpenShape-PointBERT [24]	ViT-bigG-14	70.3	86.9	91.3	51.3	69.4	78.4	13.0	23.3	29.4
	OpenShape-SparseConv [24]	ViT-bigG-14	72.9	87.2	93.0	52.7	72.7	83.6	13.7	24.2	30.0
	TAMM-PointBERT [27]	ViT-bigG-14	73.1	88.5	91.9	54.8	74.5	83.3	14.9	26.2	33.4
	TAMM-SparseConv [27]	ViT-bigG-14	74.6	88.2	94.0	57.9	75.3	83.1	-	-	-
	OpenShape-PointBERT(+Dlign)	ViT-bigG-14	73.7	87.1	91.3	52.7	72.4	82.6	13.4	23.7	29.9
	OpenShape-SparseConv (+Dlign)	ViT-bigG-14	74.9	89.5	94.1	56.3	75.2	**85.4**	15.0	26.1	32.8
	TAMM-PointBERT(+Dlign)	ViT-bigG-14	73.7	89.1	92.2	57.3	73.6	82.3	15.8	27.4	33.0
	OpenDlign-B32	ViT-B-32	68.4	86.4	92.6	46.7	72.0	83.0	17.3	29.2	36.3
	OpenDlign-B16	ViT-B-16	74.2	90.5	95.4	49.3	74.0	84.4	23.2	37.5	44.3
	OpenDlign-L	ViT-L-14	77.8	93.1	96.4	52.1	74.6	82.8	27.5	41.3	47.8
	OpenDlign-H	**ViT-H-14**	**82.6**	**96.2**	**98.4**	**59.5**	**76.8**	83.7	**31.3**	**46.7**	**53.2**
Ensemble	OpenShape-SparseConv [24]	ViT-bigG-14	83.4	95.6	97.8	56.7	78.9	88.6	33.7	49.3	57.4
	OpenShape-PointBERT [24]	ViT-bigG-14	84.4	96.5	98.0	52.2	79.7	88.7	34.0	49.7	57.9
	TAMM-PointBERT [27]	ViT-bigG-14	85.0	96.6	98.1	55.7	80.7	88.9	37.1	53.5	61.8
	TAMM-SparseConv [27]	ViT-bigG-14	85.4	96.4	98.1	58.5	81.3	89.5	-	-	-
	OpenShape-SparseConv (+Dlign)	ViT-bigG-14	85.0	96.1	97.9	56.2	78.5	87.8	34.1	50.5	58.5
	OpenShape-PointBERT (+Dlign)	ViT-bigG-14	85.4	96.5	**98.2**	51.1	77.4	88.2	35.6	50.4	57.9
	TAMM-PointBERT(+Dlign)	**ViT-bigG-14**	**86.2**	**96.6**	97.5	**60.5**	**82.5**	**90.4**	**37.5**	**54.9**	**62.1**

TAMM variants pre-trained on the ShapeNet dataset across all benchmarks. It also improves the performance of variants pre-trained on the ensemble dataset in at least two benchmarks, despite depth-aligned images being available only for the 3D data from ShapeNet, which represents no more than 10% of the ensemble dataset. Significantly, TAMM-PointBERT (+Dlign) achieves a 4.8% top-1 accuracy improvement on the ScanObjectNN dataset, and OpenShape-PointBERT (+Dlign) gains a 1.6% increase on the most challenging OmniObject3D benchmark. These results validate that using depth-aligned images is a universally effective strategy to enhance any 3D open-world pipeline.

4.2 Few-Shot 3D Classification

We then assessed OpenDlign's few-shot classification capability by training a logistic regressor with linear probing on features from N-shot, 10-view depth maps. Similar to the zero-shot scenario, we extracted multi-view features using both fine-tuned and pre-trained OpenDlign encoders (see Fig. 2). At inference, the regressor aggregates logits from 10 views to predict the final label. We compared OpenDlign's few-shot performance with variants of ULIP [23], OpenShape [24], and TAMM [27], which extract features for training regressor from point clouds using their pre-trained point encoders. Table 2 shows OpenDlign outperforms all baselines across varied few-shot scenarios with 1 to 16 training samples per class. OpenDlign significantly outperforms the leading baseline on the OmniObject3D dataset, exceeding it by 8.8% and 11.8% in 4-shot and 8-shot classification, respectively. This underscores the robustness and transferability of its 3D representations.

Table 2: **Few-shot classification results on ModelNet40 [52], ScanObjectNN [53] and OmniObject3D [54].** Our results are averaged over 10 random seeds.

Model	ModelNet40 [52]					ScanObjectNN [53]					OmniObject3D [54]				
	1-Shot	2-Shot	4-Shot	8-Shot	16-Shot	1-Shot	2-Shot	4-Shot	8-Shot	16-Shot	1-Shot	2-Shot	4-Shot	8-Shot	16-Shot
ULIP-PointBERT [23]	54.4	64.3	74.1	79.3	81.3	46.7	55.1	62.5	70.7	73.9	37.5	41.2	44.1	49.7	53.4
OpenShape-PointBERT [24]	57.5	70.1	76.5	80.4	82.1	47.9	55.6	62.7	67.0	72.0	34.5	34.1	37.8	41.9	45.6
OpenShape-SparseConv [24]	62.8	72.0	78.9	82.9	85.7	47.3	56.3	64.5	68.2	74.0	36.0	37.0	41.5	44.7	48.6
TAMM-PointBERT [27]	62.4	73.3	81.7	83.8	85.9	48.2	57.1	63.6	72.1	76.5	38.9	41.6	46.3	50.1	54.2
OpenDlign (ours)	**65.6**	**73.9**	**82.9**	**85.5**	**87.6**	**48.9**	**58.5**	**67.9**	**74.2**	**79.0**	**42.1**	**46.9**	**55.1**	**61.9**	**65.8**

4.3 Zero-Shot 3D Object Detection

We evaluated OpenDlign's capabilities in Zero-Shot 3D Object Detection using the ScanNet V2 dataset [58], which contains richly annotated 3D indoor scenes in 18 object categories. Following the PointCLIP V2 methodology [19], we began with the pre-trained 3DETR-m model to pinpoint 3D regions of interest, successfully delineating 3D bounding boxes and extracting the points inside each box. Finally, we applied OpenDlign to these points to generate our predictions. Table 3 illustrates OpenDlign's zero-shot detection prowess using mean Average Precision (mAP) at IoU thresholds

Table 3: **Zero-shot 3D object detection results on ScanNet V2 [58].**

	Method	Mean	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Counter	Desk	Sink	Bathtub
AP_{25}	PointCLIP [20]	6.00	3.99	4.82	45.16	4.82	7.36	4.62	2.19	1.02	4.00	13.40	6.46
	PointCLIP V2 [19]	18.97	19.32	20.98	61.89	15.55	23.78	13.22	17.42	12.43	21.43	14.54	16.77
	OpenDlign (ours)	**50.72**	**38.91**	**67.27**	**86.33**	**72.01**	**58.72**	**44.58**	**32.07**	**50.49**	**62.04**	**51.98**	**64.29**
AP_{50}	PointCLIP [20]	4.76	1.67	4.33	39.53	3.65	5.97	2.61	0.52	0.42	2.45	5.27	1.31
	PointCLIP V2 [19]	11.53	10.43	13.54	41.23	6.60	15.21	6.23	11.35	6.23	10.84	11.43	10.14
	OpenDlign (ours)	**37.97**	**17.04**	**66.68**	**73.92**	**54.96**	**50.03**	**24.73**	**12.84**	**20.44**	**41.64**	**34.17**	**64.29**

of 0.25 and 0.5, achieving scores of 50.72% and 37.97%, respectively. It significantly outperforms PointCLIP V2 by more than 31.75% and 26.44%. Remarkably, OpenDlign can detect the 'Sofa' shape with an AP_{50} of 54.96%, whereas PointCLIP and V2 score below 10, demonstrating OpenDlign's superior capability in extracting robust 3D representations from sparse and noisy point clouds in real-world indoor scenes.

4.4 Cross-Modal Retrieval

3D shapes were retrieved by computing the cosine similarity between the embeddings of a query and those generated by OpenDlign, followed by a k-nearest neighbors (kNN) analysis to find the most similar shapes. Fig. 3 illustrates OpenDlign's capability in matching 3D shapes to image and text queries. Column (a) illustrates its precision in distinguishing sub-categories like grand versus upright pianos from image queries. Column (b) demonstrates successful shape retrieval using distinct text descriptions, such as "Batmobile armored". Notably, averaging image and text query embeddings allows OpenDlign to find shapes that combine elements of both inputs. For example, merging a running horse image with the text "man" results in the retrieval of both a centaur and a running man, as shown in Fig. 3 (c). A house image combined with "tree" retrieves a treehouse.

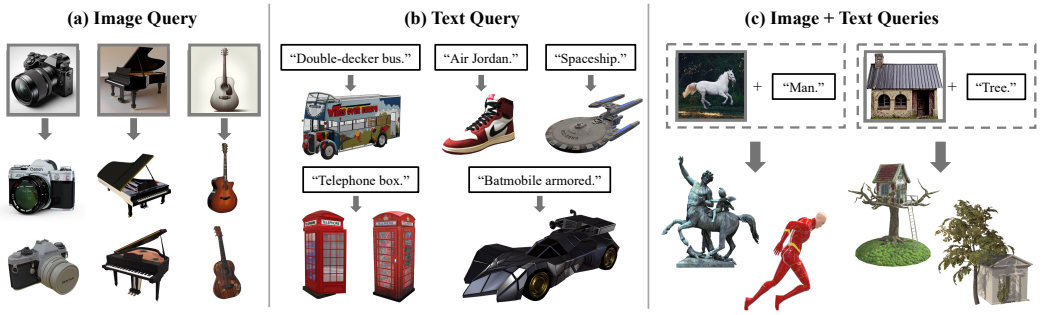

Figure 3: **3D shape retrieval results. (a)** Two most similar shapes for each query image. **(b)** Most similar shapes for each query text. **(c)** Two most similar shapes for combined image and text queries.

4.5 Ablation Study

Ablation studies were conducted on zero-shot classification benchmarks to assess the contribution of each component in OpenDlign. Consistently, all OpenDlign variants used in these studies employed OpenCLIP-ViT-H-14 as their backbone. ShapeNet was the default training dataset for all models.

Contour-Aware Projection. Replacing PointCLIP V2's projection pipeline [19] with our contour-aware version, as shown in Table 4, enables a pre-trained CLIP to reach 68.8% zero-shot accuracy on ModelNet40, even outperforming several baselines that need extra training. This suggests that through large-scale contrastive learning, CLIP can understand RGB images as well as depth maps, as long as key shape features are maintained during projection.

Multimodal Alignment. Table 4 shows that alignment between depth maps and depth-aligned images (depth-daRGB) substantially boosts performance. It improves top-1 accuracy by over 10% across datasets, indicating that depth-daRGB alignment effectively generalizes CLIP to depth maps, with consistent gains in zero-shot inference, regardless of depth-specific text prompts.

Further analysis compared depth-daRGB alignment against three alternatives: depth-rendRGB (aligning depth maps with CAD-rendered RGB images), daRGB-text & depth (aligning depth-aligned images with text before depth-daRGB alignment), and depth-text & daRGB (simultaneous alignment

Table 4: **Ablation study for OpenDlign on ModelNet40 [52] and ScanObjectNN [53].** Acc. improvements over the baseline (first-row) are highlighted in green.

Contour-Aware Projection	Multimodal Alignment	Depth-Specific Texts	Logits Aggregation	ModelNet40 [52] Top 1	ModelNet40 [52] Top 3	ModelNet40 [52] Top 5	ScanObjectNN [53] Top 1	ScanObjectNN [53] Top 3	ScanObjectNN [53] Top 5
✗	✗	✗	✗	59.7	79.6	86.3	42.8	66.7	78.4
✓	✗	✗	✗	68.8 (+9.1)	85.8 (+6.2)	91.6 (+5.3)	44.6 (+1.8)	68.3 (+1.6)	78.9 (+0.5)
✓	✓	✗	✗	79.2 (+19.5)	94.4 (+14.8)	97.6 (+11.3)	56.9 (+14.1)	75.5 (+8.8)	83.8 (+5.4)
✓	✗	✓	✗	75.9 (+16.2)	91.0 (+11.4)	95.4 (+9.1)	49.3 (+6.5)	69.8 (+3.1)	79.2 (+0.8)
✓	✓	✓	✗	80.2 (+20.5)	95.3 (+15.7)	97.7 (+11.4)	58.1 (+15.3)	75.2 (+8.5)	**84.2** (+5.8)
✓	✓	✗	✓	81.0 (+21.3)	95.2 (+15.6)	97.6 (+11.3)	56.8 (+14.0)	74.6 (+7.9)	81.6 (+3.2)
✓	✓	✓	✓	**82.6** (+22.9)	**96.2** (+16.6)	**98.4** (+12.1)	**59.5** (+16.7)	**76.8** (+10.1)	83.7 (+5.3)

of depth maps with text and depth-aligned images). Table 5 shows depth-daRGB outperforming depth-rendRGB by 6.8% on the ScanObjectNN dataset, confirming concerns that alignment with rendered images may lead to overfitting on specific 3D shapes. Moreover, daRGB-text & depth performs worst, suggesting that pre-aligning depth-aligned images with text compromises CLIP's ability to generate robust image representations, thus affecting subsequent depth-daRGB alignment efficacy. Depth-daRGB's superior performance on ModelNet40 and OmniObject3D compared to depth-text & daRGB shows that aligning depth maps with depth-aligned images indirectly aligns with text, making additional text alignment unnecessary and potentially limiting OpenDlign's generalization.

Depth-Specific Texts. Table 4 indicates that OpenDlign outperforms others in zero-shot classification tasks using depth-specific prompts, whether it incorporates multimodal alignment or logit aggregation. This implies that the inaccuracies in recognition partly result from processing input data as typical RGB images, rather than as depth maps.

Logits Aggregation. Results in Table 4 show that multi-view logit aggregation improves zero-shot classification on all datasets by combining logits from pre-trained and fine-tuned encoders. This approach effectively mitigates the catastrophic forgetting problem in OpenDlign's multimodal alignment, enabling it to recognize 3D objects identifiable by both pre-trained CLIP and OpenDlign.

Varying Number of Depth Views. OpenDlign, like other depth-based methods, necessitates extracting multiple embeddings from multi-view depth maps for zero-shot inference. Figure 4 illustrates that OpenDlign's zero-shot accuracy on both ModelNet40 and OmniObject3D increases as the number of depth map views rises. Notably, OpenDlign achieves top benchmark performance, comparable to TAMM-PointBERT, with no more than two views, indicating a good balance between latency in embedding extraction and effective zero-shot classification. Furthermore, we observed a slower performance improvement on OmniObject3D, reflecting its finer-grained classification requirements.

Table 5: **Ablation study on various alignment strategies.** Aligning with text modality was achieved by fine-tuning the image encoder.

Alignment Strategy	MNet40 Top 1	MNet40 Top 5	ScanNN Top 1	ScanNN Top 5	Omni3D Top 1	Omni3D Top 5
depth-rendRGB	78.8	96.8	52.7	82.5	29.4	51.8
daRGB-text & depth	78.6	96.4	51.1	79.6	29.1	51.6
depth-text & daRGB	79.4	98.0	**60.7**	**86.0**	29.5	52.7
depth-daRGB (ours)	**82.6**	**98.4**	59.5	83.7	**31.3**	**53.2**

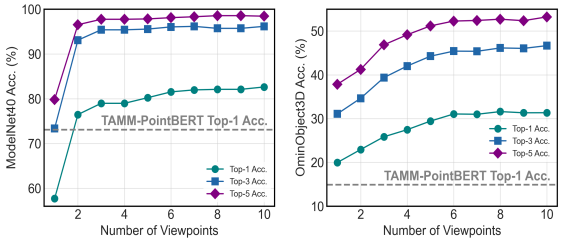

Figure 4: Impact of the number of views on OpenDlign's zero-shot performance.

5 Conclusion and Future Work

In this study, we introduce OpenDlign, an open-world framework that enhances 3D representation by efficiently fine-tuning the CLIP with depth-aligned images, which exhibit more diverse textures and colors than CAD-rendered images. Our experiments demonstrate OpenDlign's superior performance in various 3D zero-shot and few-shot tasks, especially with real-scanned objects. However, generating depth-aligned images with the ControlNet model is slower than direct CAD rendering, which extends training dataset preparation time. Moreover, depth-aligned images can be created from both CAD objects and real 3D scenes, likely highlighting a greater texture diversity gap between depth-aligned and CAD-rendered scenes and further highlighting OpenDlign's 3D scene understanding capabilities.

References

[1] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1534–1543, 2016.

[2] T. Vu, K. Kim, T. M. Luu, T. Nguyen, and C. D. Yoo, "Softgroup for 3d instance segmentation on point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2708–2717, 2022.

[3] Y. Zeng, Y. Hu, S. Liu, J. Ye, Y. Han, X. Li, and N. Sun, "Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3434–3440, 2018.

[4] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1513–1518, IEEE, 2017.

[5] A. Ten Pas and R. Platt, "Using geometry to detect grasp poses in 3d point clouds," *Robotics Research: Volume 1*, pp. 307–324, 2018.

[6] X. Li, S. Du, G. Li, and H. Li, "Integrate point-cloud segmentation with 3d lidar scan-matching for mobile robot localization and mapping," *Sensors*, vol. 20, no. 1, p. 237, 2019.

[7] 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*, pp. 652–660, 2017.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[9] 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*, pp. 16259–16268, 2021.

[10] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *arXiv preprint arXiv:1608.07916*, 2016.

[11] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2906–2917, 2021.

[12] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," in *Conference on Robot Learning*, pp. 923–932, PMLR, 2020.

[13] L. Yi, H. Su, X. Guo, and L. J. Guibas, "Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2282–2290, 2017.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[15] Y. Zang, W. Li, K. Zhou, C. Huang, and C. C. Loy, "Open-vocabulary detr with conditional matching," in *European Conference on Computer Vision*, pp. 106–122, Springer, 2022.

[16] H. Luo, J. Bao, Y. Wu, X. He, and T. Li, "Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation," in *International Conference on Machine Learning*, pp. 23033–23044, PMLR, 2023.

[17] S. Cho, H. Shin, S. Hong, S. An, S. Lee, A. Arnab, P. H. Seo, and S. Kim, "Cat-seg: Cost aggregation for open-vocabulary semantic segmentation," *arXiv preprint arXiv:2303.11797*, 2023.

[18] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, *et al.*, "Grounded language-image pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10965–10975, 2022.

[19] 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," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2639–2650, 2022.

[20] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. J. Qiao, P. Gao, and H. Li, "Pointclip: Point cloud understanding by clip," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8542–8552, 2021.

[21] T. Huang, B. Dong, Y. Yang, X. Huang, R. W. H. Lau, W. Ouyang, and W. Zuo, "Clip2point: Transfer clip to point cloud classification with image-depth pre-training," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22100–22110, 2022.

[22] D. Hegde, J. M. J. Valanarasu, and V. M. Patel, "Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition," *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2020–2030, 2023.

[23] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese, "Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1179–1189, 2023.

[24] M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han, H. Cai, F. M. Porikli, and H. Su, "Openshape: Scaling up 3d shape representation towards open-world understanding," *ArXiv*, vol. abs/2305.10764, 2023.

[25] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang, "Uni3d: Exploring unified 3d representation at scale," *ArXiv*, vol. abs/2310.06773, 2023.

[26] Z. Qi, R. Dong, G. Fan, Z. Ge, X. Zhang, K. Ma, and L. Yi, "Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining," *ArXiv*, vol. abs/2302.02318, 2023.

[27] Z. Zhang, S. Cao, and Y.-X. Wang, "Tamm: Triadapter multi-modal learning for 3d shape understanding," *arXiv preprint arXiv:2402.18490*, 2024.

[28] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q.-X. 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," *ArXiv*, vol. abs/1512.03012, 2015.

[29] 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," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13142–13153, 2022.

[30] A. Fang, A. M. Jose, A. Jain, L. Schmidt, A. Toshev, and V. Shankar, "Data filtering networks," *ArXiv*, vol. abs/2309.17425, 2023.

[31] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25278–25294, 2022.

[32] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824, 2023.

[33] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," in *International Conference on Learning Representations*, 2021.

[34] X. Zhou, R. Girdhar, A. Joulin, P. Krahenbuhl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," *ArXiv*, vol. abs/2201.02605, 2022.

[35] X. Dong, Y. Zheng, J. Bao, T. Zhang, D. Chen, H. Yang, M. Zeng, W. Zhang, L. Yuan, D. Chen, F. Wen, and N. Yu, "Maskclip: Masked self-distillation advances contrastive language-image pretraining," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10995–11005, 2022.

[36] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu, "Open-vocabulary semantic segmentation with mask-adapted clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7061–7070, 2023.

[37] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. J. Henighan, R. Child, A. Ramesh, D. M. 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," *ArXiv*, vol. abs/2005.14165, 2020.

[38] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of Learning and Motivation*, vol. 24, pp. 109–165, 1989.

[39] J. Serrà, D. Surís, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International Conference on Machine Learning*, 2018.

[40] J. Schwarz, W. M. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," *ArXiv*, vol. abs/1805.06370, 2018.

[41] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," *ArXiv*, vol. abs/1711.09601, 2017.

[42] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," *Proceedings of machine learning research*, vol. 70, pp. 3987–3995, 2017.

[43] I. Paik, S. Oh, T. Kwak, and I. Kim, "Overcoming catastrophic forgetting by neuron-level plasticity control," in *AAAI Conference on Artificial Intelligence*, 2019.

[44] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," *ArXiv*, vol. abs/1703.08475, 2017.

[45] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," *ArXiv*, vol. abs/1802.10269, 2018.

[46] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2016.

[47] A. Prabhu, P. H. S. Torr, and P. K. Dokania, "Gdumb: A simple approach that questions our progress in continual learning," in *European Conference on Computer Vision*, 2020.

[48] S. Yan, J. Xie, and X. He, "Der: Dynamically expandable representation for class incremental learning," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3013–3022, 2021.

[49] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," *ArXiv*, vol. abs/1708.01547, 2017.

[50] Y. Ding, L. Liu, C. Tian, J. Yang, and H. Ding, "Don't stop learning: Towards continual learning for the clip model," *ArXiv*, vol. abs/2207.09248, 2022.

[51] pharmapsychotic, "Clip interrogator." `https://github.com/pharmapsychotic/clip-interrogator`, 2022.

[52] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, 2014.

[53] 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*, pp. 1588–1597, 2019.

[54] 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," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 803–814, 2023.

[55] H. Fu, R. Jia, L. Gao, M. Gong, B. Zhao, S. Maybank, and D. Tao, "3d-future: 3d furniture shape with texture," *International Journal of Computer Vision*, vol. 129, pp. 3313–3337, 2021.

[56] J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, X. Zhang, T. F. Y. Vicente, T. Dideriksen, H. Arora, *et al.*, "Abo: Dataset and benchmarks for real-world 3d object understanding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21126–21136, 2022.

[57] N. Mu, A. Kirillov, D. Wagner, and S. Xie, "Slip: Self-supervision meets language-image pre-training," in *European Conference on Computer Vision*, pp. 529–544, Springer, 2022.

[58] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2432–2443, 2017.