*Cross-Mo*dal *S*elf-*T*raining: Aligning Images and Pointclouds to learn Classification without Labels

Amaya Dharmasiri
Princeton University

Muzammal Naseer
MBZUAI

Salman Khan
MBZUAI

Fahad Shahbaz Khan
MBZUAI

Abstract

Large-scale vision 2D vision language models, such as CLIP can be aligned with a 3D encoder to learn generalizable (open-vocabulary) 3D vision models. However, current methods require supervised pre-training for such alignment, and the performance of such 3D zero-shot models remains sub-optimal for real-world adaptation. In this work, we propose an optimization framework: **Cross-MoST: Cross-Mo**dal **S**elf-**T**raining*, to improve the label-free classification performance of a zero-shot 3D vision model by simply leveraging unlabeled 3D data and their accompanying 2D views. We propose a student-teacher framework to simultaneously process 2D views and 3D point clouds and generate joint pseudo labels to train a classifier and guide cross-model feature alignment. Thereby we demonstrate that 2D vision language models such as CLIP can be used to complement 3D representation learning to improve classification performance without the need for expensive class annotations. Using synthetic and real-world 3D datasets, we further demonstrate that* ***Cross-MoST*** *enables efficient cross-modal knowledge exchange resulting in both image and point cloud modalities learning from each other's rich representations. The code and pre-trained models are available here.*

1. Introduction

Recent developments of foundational models such as CLIP [34], contrastively pre-trained on large-scale 2D image text pairs, enable open-vocabulary zero-shot classification. On the other hand, the 3D visual domain which has an increasingly important role in real-world applications such as mixed reality, robotics, and autonomous driving, suffers from the limited amount of training data. Therefore, directly learning 3D foundational models lacks scalability. Recent works [16, 51] propose to train a 3D encoder by aligning its latent space with a pre-trained 2D image-text model, CLIP [34]. This allows zero-shot 3D classification whose performance, however, remains sub-optimal for real-world adaptation, especially when compared to supervised learning. Nevertheless, supervised learning requires expensive annotated datasets.[4, 7].

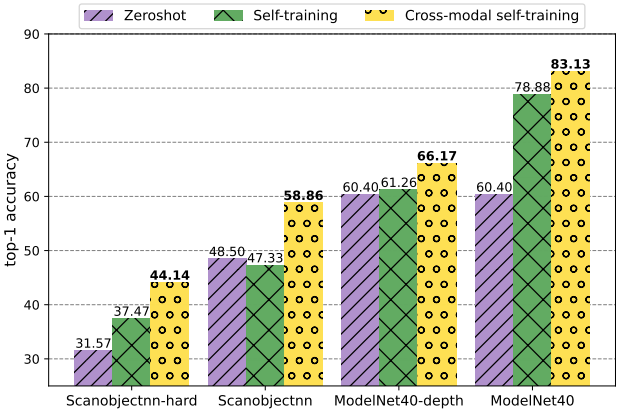

Figure 1. **Proposed cross-modal self-training** achieves significant performance gains over zero-shot [51] 3D classification, as well as recently proposed self-training[24] applied on point clouds.

Self-training is an interesting learning paradigm that belongs to semi-supervised learning and aims to adapt models for downstream tasks where pseudo-labels from unlabeled data are used as training targets. Especially for large foundational models such as CLIP pretrained on very general datasets, self-training acts as a useful training paradigm to adopt its general knowledge to specific downstream tasks without requiring any labels. Self-training on images capitalizing on the open-vocabulary zero-shot classification ability of CLIP to generate a pseudo-supervisory signal has been explored by works such as MUST[24]. However, adopting similar self-training methods in other modalities such as 3D point clouds has not been widely explored, and is accompanied by the challenge of noise in pseudo labels due to the lack of pretrained knowledge and limited open-vocabulary performance.

On the other hand, real-world data gathered by 3D scanners are often accompanied by their corresponding RGB

and/or RGBD images, whereas synthetic 3D data such as CAD models can easily be rendered into a set of defined 2D views. This provides an opportunity for two coexisting data modalities to learn from each other's unique representations to understand one reality; even without labels.

To this end, we propose "*Cross-MoST: Cross-Modal Self-Training:* Aligning Images and Pointclouds to learn Classification without Labels" aiming to 1) Harness the multimodality of data to mitigate the lack of expensive annotations, 2) Generate more robust pseudo-labels to enable self-training on 3D point clouds, and 3) Implement cross-modal learning and facilitate both image and point cloud modalities to learn from each other's unique and rich representations. Our contributions are as follows:

- We explore self-training as a setting to implement cross-modal learning. By formulating joint pseudo-labels by taking into account both 3D point clouds and their 2D views, we create more robust pseudo-labels for self-training while simultaneously aligning the two data modalities. Furthermore, we use instance-level feature alignment to complement this objective.
- We carefully engineer design elements from uni-modal self-training such as student-teacher networks for self-training, and masked-image-modeling for learning local features, to simultaneously accommodate multiple modalities.
- We demonstrate that the proposed joint pseudo-labels and feature alignment between images and point clouds enable each modality to benefit from one another's unique representations, leading to improved classification performance in each modality through label-free training.

As shown in Figure 1, *Cross-MoST* achieves respectively $+10.36\%$ and $+22.73\%$ improvement over zero-shot on the most widely used datasets; Scanobjectnn [43] and Modelnet40 [47] respectively. It also achieves respectively $+11.53\%$ and $+4.25\%$ improvement over Self-training on the single point cloud modality, highlighting the impact of cross-modal learning.

It is important to note that Cross-MoST is orthogonal to works such as ULIP[51] which addresses self-supervised pre-training in images or point clouds, as well as works such as MUST[24] which explores self-training on a single (image) modality. Advances in either or both domains will lead to even better Cross-modal Self-training paradigms. We present *Cross-MoST* as a simple and effective solution for 3D classification that unlocks the potential of CLIP-like foundational models in practical scenarios where 3D scans and their corresponding 2D views are abundant, but the labels are scarce. We further demonstrate the effectiveness of our *Cross-MoST* on 8 different versions of 4 popular 3D datasets, collectively representing synthetic and real-world 3D objects and 2D images, as well as real RGB, synthetic rendered, and depth-based 2D images.

2. Related work

Supervised Training: One approach to point cloud modeling involves projecting 3D point clouds into voxel or grid-based representation [27, 39], followed by 2D or 3D convolutions for feature extraction. On the other hand, PointNet [31] and PointNet++ [32] pioneered directly ingesting 3D point clouds and extracting permutation-invariant feature representations through shared MLPs. These networks have been widely used in various point cloud applications [9, 46, 50]. Similarly, PointNeXt [33] emerged as a lightweight version of PointNet++. Point Transformer [55] and PCT [13] show improvements in the supervised training paradigm using transformer-based networks.

Self-Supervised Training: self-supervised 3D pre-training methods [30, 38] utilize encoder-decoder architectures to first transform point clouds into latent representations, and then recover them into the original data form. Other works [10, 35, 48] conducts self-supervised pre-training via contrastive learning. Inspired by the BERT [8] in the language domain and masked image modeling [15, 49], several works have been proposed for masked point modeling for self-supervised learning [25, 29, 52, 54].

Multi-modal Pre-Training: Recent advancements in multi-modal contrastive learning have enabled CLIP [34] to perform robust and efficient multi-modal training with millions of image text pairs. CLIP has been extended to high-efficiency model training such as ALBEF [23], cycle consistency [11], and self-supervised learning [28]. PointClip [53, 57] and CLIP2Point [17] attempt to leverage the superior pre-trained knowledge of CLIP to improve 3D shape understanding by converting point clouds into multiple 2D views/depth maps. ULIP [51] and CG3D [16] enable direct processing of point clouds using a separate 3D encoder, and leverage point cloud, image, and language triplets to align these three modalities together. Recent methods such as Uni3D[56], Vit-Lens[22], and OpenShape[26] scale up multi-modal pre-training using larger datasets.

Self-training for Semi-supervised Training: Self-training improves the quality of features by propagating a small initial set of annotations to a large set of unlabeled instances, and has shown promising progress in domains including vision [37, 49, 58], NLP [14], and speech [18]. [2, 24, 40, 42]. Similar to [24], this work uses pseudo labels[21] and applies consistency regulation [20, 40] objective to encourage the model to output same predictions when perturbations are added to image/point cloud inputs, and guide the model to give sharp predictions with low entropy [12].

We solve the problem of confirmation bias in self-training by combining pseudo-labels from complementary modalities to eliminate label noise and improve robustness Furthermore, similar to [3, 42], we model the teacher as an exponentially moving average of the student, thus improving the tolerance to inaccurate pseudo-labels.

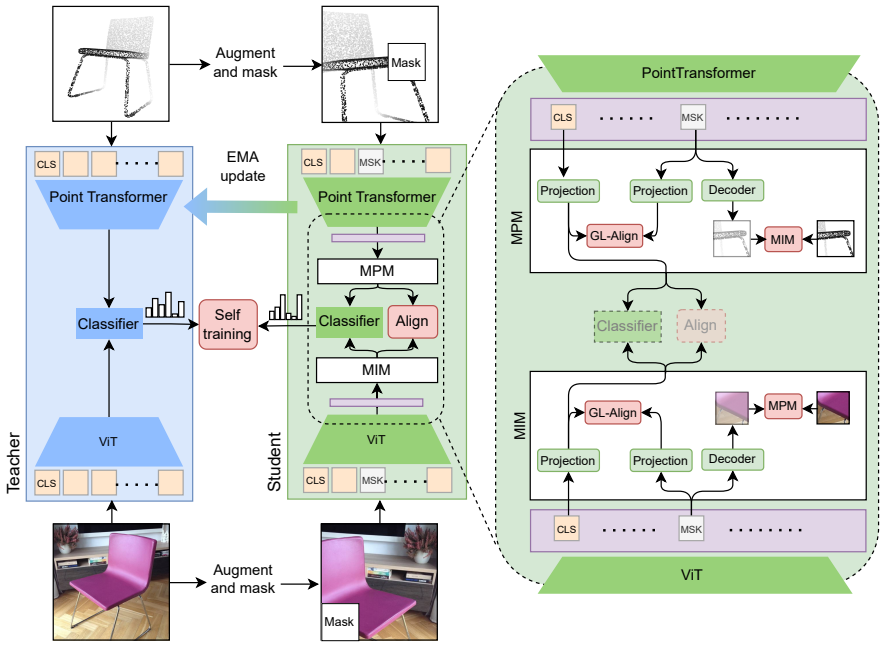

Figure 2. Cross-modal Self-training for 3D point clouds and their corresponding 2D views. The teacher (blue) weights are updated as an exponentially moving average of the student (green). The teacher generates joint pseudo-labels to allow cross-modal self-training. Our **MPM** and **MIM** modules inside the student model implement masked point and image modeling. **Align** represents the cross-modal feature alignment, whereas **GL-Align** within MIM and MPM modules represent global-local feature alignment to support masked modeling within each individual modality (image and pointcloud).

3. Cross-Modal Self-Training

As shown in Figure.2, we operate in a unified embedding space for both image and point cloud branches. We initialize the image encoder ViT with a CLIP pre-trained image encoder such that a common classifier could be initialized using CLIP text embeddings corresponding to the categories available in the training dataset. The point and image encoders, as well as the classifiers of student and teacher networks, are initialized identically. An input training pair consists of a 3D point cloud, and an image representing the corresponding 3D object from a random view. The teacher network classifier generates the joint pseudo-label for the input image-point cloud pair by combining the two corresponding sets of classification logits. The same input pair is sent to the student network with heavy augmentations and masking to generate two sets of predictions using the student network classifier. Cross entropy loss is calculated between the joint pseudo labels and the student network predictions, and the weights of the classifier as well as both image and point cloud encoders are updated accordingly. Finally, the teacher network weights are updated as an exponentially moving average of the student. Additionally, image and point mask modeling, as well as alignment losses complement our cross-modal self-training process as regularizers and additional supervision signals.

3.1. Preliminaries

CLIP [34] pre-trains an image encoder and a text encoder with a contrastive loss such that paired images and texts have high similarities in a shared embedding space. We denote the CLIP's image and text encoders as h^I and h^S. The input image X is divided into K patches followed by a projection to produce patch tokens. Then a learnable [CLS] token is preppended to the input patch embeddings to create $\mathbf{X} = \{x_{cls}, x_1, x_2, ...x_K\}$ which is used as input to h^I, $h^I(\mathbf{X}) = \tilde{\mathbf{X}} = \{\tilde{x}_{cls}, \tilde{x}_1, \tilde{x}_2, ...\tilde{x}_K\}$. The output embedding of the [CLS] token is then normalized and projected to obtain the feature embedding of the image, $f^I(\tilde{x}_{cls}) = \tilde{x}$.

For zero-shot classification, each category c's name is wrapped in several templates such as `"a photo of a {category}"`, `"a picture a {category}"` to produce s_c. These text descriptions are passed to the text encoder to yield the category-level normalized text embedding, $\tilde{s}_c = avg\left(h^S(s_c)\right)$. During inference, the dot product between the text embeddings $\tilde{\mathbf{S}} = \{\tilde{s}_i\}_{c=1}^{C}$ and the image embedding yield prediction logits; $\tilde{x}.\tilde{\mathbf{S}} = p_{img}$

Uni-modal self-training for images: MUST [24] proposes an EMA teacher-student setting to implement self-training on images using CLIP's zero shot prediction ability to generate pseudo-labels. It converts CLIP's non-parametric text embedding $\tilde{\mathbf{S}}$ into weights of a linear classifier Q that takes as input the feature embedding of images; $Q(\tilde{x}) = p$. Both the EMA teacher and the student models are initialized with the same pre-trained weights, $\theta = \{\theta^I, \theta^Q\}$, where θ^I is the weights of CLIP visual encoder. The model teacher weights are updated at each iteration as: $\Delta = \mu\Delta + (1-\mu)\theta$. A batch B of weakly augmented images is passed to the teacher model to yield a set of soft prediction logits q_b, which is converted to hard pseudo-labels as $\hat{q}_b = argmax_c(q_b)$. The same batch of images is sent

to the student model with strong augmentations to yield prediction logits p_b. Self-training loss is calculated as the cross-entropy H between the predictions and the pseudo-labels that exceed a confidence threshold, T.

$$\mathcal{L}_{cls} = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}(\max(q_b) \geq T) H(\hat{q}_b, p_b) \tag{1}$$

This encourages the model to return the same predictions for perturbed inputs, while the student learns stronger representations as augmentations are applied to its input.

3.2. Cross-Modal Self-training for images and pointclouds

In this subsection, we elaborate the architectural elements and loss functions in Cross-MoST as shown in Figure.2.

Pointcloud encoder: We denote an input point cloud as Y. Following the work of PointBert [52], we first cluster the point cloud object into K local patches or sub-clouds by applying the k-nearest-neighbor algorithm on a set of selected sub-cloud centers. Similar to patchifying in images, these sub-clouds contain only local geometric information, regardless of their original location. Next, they are passed through a pre-trained Point tokenizer from [52] to convert each sub-cloud into a point embedding, and a learnable [CLS] token is appended as follows; $\mathbf{Y} = \{y_{cls}, y_1, y_2, ...y_K\}$. This enables each point cloud object to be represented as a sequence of tokens and sent as input to a standard transformer encoder, h^P as $\tilde{\mathbf{Y}} = h^P(\mathbf{Y}) = \{\tilde{y}_{cls}, \tilde{y}_1, \tilde{y}_2, ...\tilde{y}_K\}$. The output embedding of the [CLS] token $\tilde{y}_{cls}$ is normalized and projected to yield the point cloud feature embedding, $f^P(\tilde{y}_{cls}) = \tilde{y}$.

Following the work of ULIP[51] we pre-train the point cloud encoder to learn a 3D representation space aligned with the image-text embedding space of CLIP. The pre-training is done on synthesized image-text-point cloud triplets from the CAD models of Shapenet[4] dataset.

EMA teacher-student setting: We use the non-parametric text embeddings $\tilde{\mathbf{S}}$ to initialize the classifier Q. Since both image and point cloud features are now projected to the same embedding space, *the same classifier Q is used to obtain prediction logits for both modalities.* $Q(\tilde{x}) = p_{img}$, $Q(\tilde{y}) = p_{pcl}$. We use an ensemble of text prompts such as `"a photo of a {category}"`, `"a 3D model of a {category}"` and average them to initialize the classifier, to ensure sufficient zero-shot accuracy for both image and point cloud inputs.

Now, we characterize our model parameters as $\theta = \{\theta^I, \theta^P, \theta^Q\}$, while the teacher is the EMA of the student model similar to MUST i.e., $\Delta = \mu\Delta + (1-\mu)\theta$. Our proposed self-training leverages cross-modal losses derived by joint pseudo-labels and feature alignment as well as regularization based on masked modeling and fairness.

Cross-Modal cross-entropy via joint pseudo-labels: We create a training sample by pairing a point cloud object with an image showing the object from a random viewpoint. A set of weak augmentations are applied to the batch B of image and point cloud pairs which is then passed through the teacher model to obtain two sets of soft prediction logits from each modality; $q_{b,img}, q_{b,pcl}$ which are used to derive modality-specific pseudo-labels $\hat{q}_{b,img} = \text{argmax}_c(q_{b,img})$ and $\hat{q}_{b,pcl} = \text{argmax}_c(q_{b,pcl})$. Then, the teacher prediction for each sample b that corresponds to the highest confidence between the two modalities in $\hat{q}_{b,img}$ and $\hat{q}_{b,pcl}$ is selected to assemble a set of joint pseudo-labels, $\hat{r}_b$. The confidence score from the selected modality for each sample is defined as r_b; confidence score for the combined pseudo-label. Following the work of FixMatch [40] and MUST [24], we select the samples whose confidence exceeds a threshold T to act as pseudo-labels. (further details in Appendix **??**).

The student model receives strongly augmented versions of the same image-point cloud pairs, yielding predictions $p_{b,img}, p_{b,pcl}$. We apply cross-entropy loss H between the student predictions and the selected pseudo-labels.

$$\mathcal{L}_{cls} = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}(r_b > T)\{H(\hat{r}_b, p_{b,\ img}) + H(\hat{r}_b, p_{b,\ pcl})\} \tag{2}$$

The pseudo-label selection through confidence thresholding indirectly encourages the model to yield sharp predictions. Moreover, the combination facilitates cross-modal exchange of class-level feature knowledge while encouraging pseudo-label agreement between modalities.

Cross-modal feature alignment: Extending the pertaining objective of CLIP and ULIP, we continue to enforce Feature alignment between image and point cloud pairs in the multimodal embedding space. Thereby we encourage images and point clouds representing one instance of reality to be embedded close to each other. This unsupervised objective complements the pseudo-supervised class-level discrimination implemented by self-training.

For a batch B of training image-point cloud pairs, we calculate the cosine similarity between the feature embeddings of each image and point cloud $\tilde{x}_b, \tilde{y}_b$, and maximize the similarity of image-point cloud embeddings of B positive pairs, while minimizing the similarity of $B^2 - B$ negative pairs. We implement this objective by optimizing a symmetric cross-entropy over the similarity scores.

$$\mathcal{L}_{align} = \sum_{(i,j)} -\frac{1}{2}\log\frac{\exp\left(\frac{\tilde{x}_i\tilde{y}_j}{\tau}\right)}{\sum_b \exp\left(\frac{\tilde{x}_i\tilde{y}_b}{\tau}\right)} - \frac{1}{2}\log\frac{\exp\left(\frac{\tilde{x}_i\tilde{y}_j}{\tau}\right)}{\sum_b \exp\left(\frac{\tilde{x}_b\tilde{y}_j}{\tau}\right)} \tag{3}$$

Fairness Regularization: As shown by prior research [45], CLIP-like models are often biased towards certain classes,

causing the pseudo-labels to further magnify such biases in the self-training settings. Therefore, we apply fairness regularization on both image and point cloud predictions separately during self-training by enforcing the following loss:

$$\mathcal{L}_{fair} = -\frac{1}{C}\sum_{c=1}^{C}\left(log(\overline{p}_{c,img}) + log(\overline{p}_{c,pcl})\right) \quad (4)$$

where $\overline{p}$ denotes the batch-average prediction and C is the number of class categories.

Masked-Modeling MM: Masked image and point cloud modeling aim to learn specific local features at image-patch and point-subcloud levels respectively. It not only acts as a regularization to further reduce the noise in the pseudo labels but also as a complementary supervision signal from unlabeled data.

Masked-image reconstruction - ***MIM:*** We formulate the masked image modeling (MIM) objective as predicting the missing RGB values of masked-out patches using contextual information learned through attention. We train a simple linear decoder g^I that takes output embedding $\tilde{x}_m$ of mth [MSK] token as input and reconstructs the masked-out RGB pixel values; $z_m = g^I(\tilde{x}_m)$ which are then compared with the ground truth patch to formulate the MIM loss.

$$\mathcal{L}_{mim} = \frac{1}{MN}\sum_{m=1}^{M}\|z_m - \sigma_m\|_1$$

$z_m,\ \sigma_m \in R^N$, N is the number of RGB pixels in a patch, σ_m is the RGB values of the originally masked out patch.

Masked-point reconstruction - ***MPM:*** Representing each sub-cloud of a 3D spatial locality by a discrete token enables us to implement a masked point reconstruction objective. A selected set of tokens from $\mathbf{Y}$ is replaced by a learnable [MSK] token and passed as input to the point cloud encoder h^P. Random masking of patches from multiple locations makes the learning objective too easy given the richness of contextual information from neighboring patches in 3D point clouds. Therefore, we mask out a selected token, and m other tokens corresponding to sub-clouds in its spatial neighborhood resulting in a block-wise masking strategy [52]. We pass the output token through a linear decoder g^P; $w_m = g^P(\tilde{y}_m)$, and apply an L_1 loss between w_m and corresponding the masked-out input token y_m.

$$\mathcal{L}_{mpm} = \frac{1}{MN}\sum_{m=1}^{M}\|w_m - y_m\|_1$$

where $w_m,\ y_m \in R^N$, and N is the dimensionality of the point token. The model learns to predict missing geometric structures of a point cloud given its neighboring geometries, encouraging local feature representation learning.

Global-local alignment: Both MIM and MPM encourage each branch to learn rich local semantics. Then we transfer this local feature information to the object-level embedding space by aligning local and global features within each modality. Specifically, we project the output embeddings of all [MSK] tokens to the image/point cloud feature embedding space using f^I and f^P, and calculate a global-local feature alignment loss *within* each modality.

$$\mathcal{L}_{lg-align} = \frac{1}{M_{img}}\sum_{m=1}^{M_{img}}\|\tilde{x}-u_m\|_2^2 + \frac{1}{M_{pcl}}\sum_{m=1}^{M_{pcl}}\|\tilde{y}-v_m\|_2^2$$

where $u_m = f^I(\tilde{x}_m)$, $\tilde{x}_m$ is the output embedding of the m^{th} masked image token, and where $v^m = f^P(\tilde{y}_m)$, $\tilde{y}_m$ is the output embedding of the m^{th} masked point token.

4. Experimental Protocols

Datasets details: Shapenet [4] consists of textured CAD models of 55 object categories. We use Shapenet to pretrain the point cloud branch for better self-training initialization. **ModelNet40** [47] is a synthetic dataset of 3D CAD models containing 40 categories. We pair 2D renderings of CAD models with the point clouds to create **Modelnet40** and **ModelNet10** (a subset of 10 common classes). We follow the realistic 2D views generated using [57] to generate the dataset **ModelNet40-d** (depth). **Redwood** [5] is a dataset of real-life high-quality 3D scans and their mesh reconstructions. We randomly sample 20 frames from the RGB videos of each object scan and use them in our image encoder. **Co3D** [36] is a large-scale dataset of real multi-view images capturing common 3D objects, and their SLAM reconstructions. We sample 20 GRB images per object for our image encoder. For *Scanobjectnn* [43] with real 3D point cloud scans, we report results on 3 different versions; **Sc-obj** - scans of clean point cloud objects, **Sc-obj withbg** - scans of objects with backgrounds, and **Sc-obj hardest** - scans with backgrounds and additional random scaling and rotation augmentations. Multiview images for all versions of scanobjectnn are generated using realistic 2D view rendering from point clouds[57]. Detailed descriptions of the sizes, number of categories, and pre-processing applied to each dataset are provided in Appendix **??**.

Implementation details: We use ViTB/16 as the image encoder, and a standard transformer [44] with multi-headed self-attention layers and FFN blocks as our point cloud encoder. We use AdamW [19] optimizer with a weight decay of 0.05. The batch size is 512, and the learning rate is scaled linearly with batch size as (lr= base_lr*batchsize/256). We used 4 V100 GPUs for training. Further details are in Appendix **??**.

Image encoder: We use a ViT-B/16 model pretrained by [34] for the image branch. After resizing to the side

224×224, random cropping is applied as a weak augmentation on the input to the teacher model to generate pseudo-labels. Stronger augmentations; RandomResizedCrop+Flip+RandAug [6] are applied to the inputs to the student model. We implement a patch-aligned random masking strategy where multiple image patches are randomly masked with a fixed ratio of 30%.

Point cloud encoder: We used Shapenet [4] dataset and its rendered 2D views from [51] to pre-train the point cloud encoder and the projection layers f^P, f^S, and f^I. Pre-training is done for 250 epochs with a learning rate of 10^{-3} with AdamW optimizer with a batch size of 64. We divide each point cloud into 64 sub-clouds and use a Mini-Pointnet to extract embeddings of each sub-cloud, followed by a pointbert[52] encoder to convert each sub-cloud into point embeddings. Rotation perturbation and random scaling are applied on the input point cloud to the teacher model to generate pseudo-labels. Stonger augmentations; random cropping, input dropout, rotate, translate, and scaling are applied to the input to the student model. Random masking is applied to 30% to 40% of the point embeddings.

4.1. Results

We perform experiments on 4 datasets with point clouds and associated images, which span different types of 3D point clouds; *Modelnet*- sampled from CAD models, *Redwood*- real 3D scans, and *Co3D*- SLAM reconstructed point clouds. We also use different types of images; *Modelnet*- 2D rendered CAD models, *Redwood, Co3D*- real images, and *Scanobjectnn*- realistic depth renderings from point clouds. Table 1 shows the results of our proposed cross-modal self-training.

Baselines: 1) *Baseline Zeroshot*- we used ULIP [51] trained with CLIP [34] initialization. We use the embeddings of text prompts `"a photo of a {category}"`, `"a 3D model of a {category}"` to initialize the classifier and directly evaluate the classification. 2) *Baseline self-training [24]*- We removed all cross-modal modules and cross-modal losses, and applied self-training *individually* on 2D and 3D branches using their own pseudo-labels without any cross-modal combination. For both baselines, ULIP pretraining was done on the point cloud branch to provide consistent initialization.

Evaluation: The same classifier operates on both images and point cloud embeddings. To emulate real-life scenarios where both image and point cloud data are simultaneously available for a test sample, as well as to show how both modalities benefit from each others' unique knowledge, we report the classification accuracies for *Image* and *Point cloud* encoders separately. *Image** is calculated by using the average of image embeddings of all 2D views corresponding to the test object, hence is often higher than *Image* due to richer information from multiple views. Datasets such as Co3D have very noisy and occluded point clouds, but are accompanied by high-quality RGB images; leading to higher accuracy on *Image* and *Image**. We report the accuracies of both branches to demonstrate the cross-modal learning impacts the two modalities.

As shown in Table 1, we substantially improve upon the baselines on all datasets. Especially, comparison between Baseline Self-training (individual self-training on each branch without any cross-modal label or feature exchange) and Cross-modal self-training for image and point cloud branches suggests that our proposed method enables both modalities to learn from each other

An important result is that for datasets such as *Modelnet40-d* whose zero-shot accuracies on the point cloud branch are higher than that of the image (57.74% and 30.31% respectively), Cross-modal self-training significantly improves point cloud branch accuracy above baseline self-training (from 61.26% to 66.17%). This effect is even more pronounced in the variants of *Scanobjectnn* resulting in an even higher performance on *Image** compared to *Point cloud*. Similarly, for datasets such as *Redwood* whose zero-shot accuracies on image branch are higher than that of point clouds (85.71% and 55.95% respectively), Cross-modal self-training significantly improves image branch accuracy above baseline self-training (from 91.67% to 94.05%). This shows that *even with low zero-shot performance, unique knowledge of the 3D branch due to their rich geometric details and that of the image branch due to large-scale CLIP pertaining can provide strong complementary training signals to the other modality.*

Comparisons with SOTA: In Table.6, we report the performance of state-of-the-art pre-training methods on modelnet40. To preserve consistency, we only include the methods that use Shapenet for pre-training. However, it is important to distinguish between open-vocabulary and self-training settings. Although no labels are used in the process, self-training can be framed as a further adaptation of an open-vocabulary model for a specific set of categories. Another implication is that recent improvements in pre-training [22, 26] could further improve the performance of cross-modal self-training by providing even better initialization than ULIP[51].

4.2. Ablative Analysis

Effect of proposed objectives: In Table 3, we ablate the main components of the proposed cross-modal self-training setting for Modelnet40, and illustrate their contribution to the final architecture. The best performance is achieved when all the components are used in combination. It is also important to note that pretraining the point cloud encoder on even a limited dataset such as Shapenet dramatically improves the performance on both point cloud and image branches after Cross-modal self-training.

Method	**Datasets** →	ModelNet10	ModelNet40	ModelNet40-d	Redwood	Co3d	Sc-obj	Sc-obj withbg	Sc-obj hardest
Baseline Zeroshot	Image	55.00	54.00	23.18	85.71	90.30	19.11	16.70	13.12
	Image*	65.50	56.25	30.31	85.71	94.01	26.16	19.28	13.85
	Point cloud	75.50	58.75	57.74	55.95	13.20	46.99	42.86	31.57
Baseline Self-training	Image	78.00	73.13	35.58	86.91	91.08	23.24	20.65	18.25
	Image*	85.50	78.88	46.80	91.67	92.44	27.54	28.23	23.87
	Point cloud	85.50	69.13	61.26	63.10	16.90	47.33	49.57	37.47
Cross-modal self-training	Image	86.50	79.50	52.92	88.10	93.15	52.84	49.57	40.60
	Image*	89.50	82.75	62.89	**94.05**	**94.22**	**58.86**	**55.94**	**44.14**
	Point cloud	**90.00**	**83.13**	**66.17**	75.00	83.45	48.02	51.46	41.64

Table 1. We evaluate the classifier on 2D views/images and 3D point clouds separately. Image* indicates the performance of the classifier averaged over all rendered views. For each dataset, we highlight the highest achieved accuracy among the three evaluation settings. Cross-modal self-training consistently improves over Zeroshot and self-training baselines for both images and point clouds highlighting the potential of Cross-modal learning.

Method	**Datasets** →	ModelNet10	ModelNet40	ModelNet40-d	Redwood	co3d	Sc-obj	Sc-obj withbg	Sc-obj hardest
Cross-modal Self-training	Image	86.50	79.50	52.92	88.10	93.15	52.84	49.57	40.60
	Image*	89.50	82.75	62.89	**94.05**	**94.22**	**58.86**	55.94	**44.14**
	Point cloud	90.00	**83.13**	66.17	75.00	83.45	48.02	51.46	41.64
Without L-Align	Image	86.50	78.38	53.57	88.10	88.30	48.71	48.02	39.24
	Image*	**91.00**	83.00	62.24	**94.05**	90.30	55.25	55.25	43.13
	Point cloud	**91.00**	81.75	65.40	75.00	81.39	48.19	50.60	39.59
Without MM	Image	87.50	80.13	57.66	83.33	81.74	50.26	52.32	40.94
	Image*	**91.00**	82.88	66.37	88.10	83.81	54.39	**59.55**	43.72
	Point cloud	89.50	81.75	**67.10**	73.81	79.53	49.91	55.94	42.05

Table 2. Ablations on all datasets. **Without L-Align** refers to ablating the cross-modal feature alignment by removing L_{align}. **Without MM** refers to ablating the masked-modeling in both branches; image and point cloud by removing both L_{mim} and L_{mpm}.

Align	Comb	MM	Init.	Image	Image*	Pointcloud
✗	✓	✓	✓	78.38	83.00	81.75
✗	✗	✓	✓	73.13	78.88	69.13
✓	✗	✓	✓	73.75	77.38	72.63
✓	✓	✗	✓	80.13	82.88	81.75
✓	✓	✓	✗	73.00	78.63	21.88
✓	✓	✓	✓	79.50	82.75	**83.13**

Table 3. **Aign**- Cross-modal feature alignment, **Comb**- Joint pseudo-label, **MM**-Image and Point masked modeling, and **Init.**- point cloud encoder initialization using ULIP pre-training. Results are reported on Modelnet40.

Pseudo-labels	Image	Image*	Point cloud
No Comb.	73.75	77.38	72.63
Image only	72.50	76.88	78.38
Point cloud only	71.63	74.38	70.50
Random	78.25	81.88	81.88
Our (Joint)	79.50	82.75	**83.13**

Table 4. Effect of pseudo-labels on self-training for modelnet40. **No comb.** does self-training on individual modalities using its own pseudo-labels. **Only image/pointcloud** adapts the pseudo-label from one of the modalities for both.

Views	Image	Image*	Pointcloud
1	76.75	76.75	78.50
2	78.38	80.75	80.13
4	78.50	82.38	80.50
8	78.88	82.00	81.25
12	79.50	82.75	**83.13**

Table 5. Increasing the number of 2D rendered views per point cloud improves the performance of our cross-modal self-training. Results are reported on Modelnet40.

Method	Modelnet40 (top1) accuracy
Openshape[26]-Pointbert	70.3
VIT-LENS-Datacomp-L14[22]	70.6
ULIP-Pointbert[51]	60.4
ULIP-Pointbert with Cross-MoST	**83.13** (+22.73)

Table 6. Reported performance of state-of-the art Zero-shot models trained on Shapenet[4] compared with proposed Cross-MoST

In Table 2, we report the results of ablating Cross-modal feature alignment and Masked-modeling and confirm that these design elements lead to a clear improvement of performance in a majority of datasets. As we qualitatively compare in Appendix **??**, the quality, hence the difficulty of these datasets varies dramatically in both modalities. Masked modeling is more advantageous to datasets (such as Redwood and Scanobjectnn) with point clouds with large distribution shifts from Shapenet due to obfuscations and heavy augmentations. Furthermore, in Co3D image branch, ablating masked modeling leads to a degradation in performance indicating the importance of its regularization effects.

Effect of pseudo-label combination: Table 4 compares the performance of the model with different approaches to derive pseudo-labels. *Random* refers to randomly picking a prediction from either branch image or point cloud to act as the pseudo-label for a given input pair. Cross-modal learn-

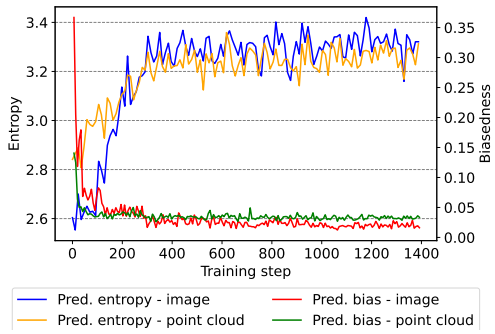

Figure 3. As the training progresses, biasness towards certain classes is significantly reduced in both branches. Predictions on each branch become more sharp, as indicated by increasing entropy. (modelnet40)

	Class Type →	All classes	Medium classes	Hard classes
Baseline	Image	54.00	47.50	44.71
Zeroshot	Image*	56.25	**49.55**	**47.65**
	Point cloud	**59.50**	42.73	36.76
Self-training	Image	73.13	67.73	66.18
Baseline	Image*	78.88	73.64	72.06
	Point cloud	69.25	57.27	52.06
Cross-modal	Image	79.50	72.73	72.65
self-training	Image*	82.75	**77.27**	**77.65**
	Point cloud	**83.13**	76.82	76.47

Table 7. Accuracy on unseen classes for Modelnet40. Cross-modal self-training significantly outperforms self-training in *medium* and *hard* classes, especially in the point cloud branch.

ing significantly improves the performance on both image and point cloud branches even with a random combination of pseudo-labels. The proposed score-based method further improves accuracy by using the most confident predictions between the modalities to act as the pseudo-label.

Effect of the number of 2D views: In our experiments with ModelNet40 [41] we have 12 2D views for each point-cloud object. The ablation results in Table 5 indicate that object understanding benefits from using multiple different views to extract more detailed visual understanding. This improvement is also reflected in the point cloud branch due to the cross-modal training.

Training Analysis: By thresholding the score of teacher predictions from each branch and combining them to generate joint pseudo-labels, we implicitly encourage the model to give sharp predictions. We calculate *prediction entropy* as the *KL divergence between a uniform distribution and the softmax predictions* of each branch to quantify this behavior. Figure 3 illustrates how the entropy of predictions increases with self-training. CLIP models are known to result in predictions biased towards certain classes [45]. This hinders the ability to self-train since such biases can be further amplified [1]. Figure 3 shows how our regularization and cross-modal learning objectives discourage this confirmation bias as training progresses. The *biasedness* is calculated as *KL divergence between a uniform distribution and the class distribution of the predictions for a balanced test set* (further details on the calculation of entropy and biasedness are in Appendix **??**).

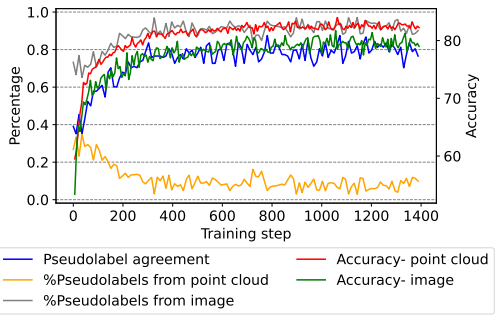

Figure 4. The percentage of pseudo-labels selected from each modality for combined self-training. The agreement between pseudo-labels increases as our training progresses. (modelnet40)

Figure 4 shows the percentage of pseudo-labels picked from each modality for the combined self-training. Although the accuracy of each individual branch is comparable, the modal steers itself to pick more pseudo-labels from the image branch as training progresses.

Self-training on unseen classes: Certain classes of ModelNet40, have been already introduced to the model during supervised pre-training of the point cloud encoder by Shapenet55. For a fairer comparison of zeroshot and label-free classification performance, therefore we evaluate our model on 2 other splits of ModelNet40 as proposed by [51]- *medium* and *hard*, with non-overlapping object classes (further details of these splits in Appendix **??**). Results in table 7 show that cross-modal self-training significantly improves the accuracy of hard and medium classes.

5. Conclusion

In this paper, we proposed a simple framework to adapt an open-vocabulary 3D vision model to downstream classification without using any labels. The core of the proposed approach is to enable cross-modal self-training by leveraging pseudo-labels from point clouds and their corresponding 2D views and additionally aligning their feature representations at the instance level. The proposed method is orthogonal to pretrained foundational models and the quality of 2D images that provide complementary information. Therefore, improvements in pre-trained foundational models or the quality of 2D views/renderings of point clouds will further improve the results of self-training in our proposed framework. Our work highlights how the rich knowledge of CLIP-based models can be adapted to better understand 3D realities even in the absence of class-level labels.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning, 2020. 8

[2] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring, 2020. 2

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021. 2

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012*, 2015. 1, 4, 5, 6, 7

[5] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans, 2016. 5

[6] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019. 6

[7] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects, 2022. 1

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 2

[9] Siqi Fan, Qiulei Dong, Fenghua Zhu, Yisheng Lv, Peijun Ye, and Feiyue Wang. Scf-net: Learning spatial contextual features for large-scale point cloud segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14499–14508, 2021. 2

[10] Kexue Fu, Peng Gao, ShaoLei Liu, Renrui Zhang, Yu Qiao, and Manning Wang. Pos-bert: Point cloud one-stage bert pre-training, 2022. 2

[11] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A. Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining, 2022. 2

[12] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, page 529–536, Cambridge, MA, USA, 2004. MIT Press. 2

[13] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 2

[14] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation, 2020. 2

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 2

[16] Deepti Hegde, Jeya Maria Jose Valanarasu, and Vishal M. Patel. Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition, 2023. 1, 2

[17] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson W. H. Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training, 2022. 2

[18] Jacob Kahn, Ann Lee, and Awni Hannun. Self-training for end-to-end speech recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020. 2

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

[20] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning, 2017. 2

[21] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2013. 2

[22] Weixian Lei, Yixiao Ge, Jianfeng Zhang, Dylan Sun, Kun Yi, Ying Shan, and Mike Zheng Shou. Vit-lens: Towards omni-modal representations, 2023. 2, 6, 7

[23] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems*, 2021. 2

[24] Junnan Li, Silvio Savarese, and Steven C. H. Hoi. Masked unsupervised self-training for label-free image classification, 2023. 1, 2, 3, 4, 6

[25] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds, 2022. 2

[26] Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding, 2023. 2, 6, 7

[27] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 2

[28] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training, 2021. 2

[29] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning, 2022. 2

[30] Omid Poursaeed, Tianxing Jiang, Han Qiao, Nayun Xu, and Vladimir G. Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028, 2020. 2

[31] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas 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)*, 2017. 2

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on

point sets in a metric space. In *Advances in Neural Information Processing Systems*, 2017. 2

[33] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies, 2022. 2

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021. 1, 2, 3, 5, 6

[35] Yongming Rao, Jiwen Lu, and Jie Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds, 2020. 2

[36] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction, 2021. 5

[37] Attaullah Sahito, Eibe Frank, and Bernhard Pfahringer. Better self-training for image classification through self-supervision, 2021. 2

[38] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space, 2019. 2

[39] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection, 2021. 2

[40] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence, 2020. 2, 4

[41] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition, 2015. 8

[42] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, 2018. 2

[43] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1588–1597, 2019. 2, 5

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 5

[45] Xudong Wang, Zhirong Wu, Long Lian, and Stella X. Yu. Debiased learning from naturally imbalanced pseudo-labels, 2022. 4, 8

[46] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019. 2

[47] 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 (CVPR)*, 2015. 2, 5

[48] Saining Xie, Jiatao Gu, Demi Guo, Charles Qi, Leonidas Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[49] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling, 2022. 2

[50] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3173–3182, 2021. 2

[51] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding, 2023. 1, 2, 4, 6, 7, 8

[52] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling, 2022. 2, 4, 5, 6

[53] 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, 2021. 2

[54] Renrui Zhang, Ziyu Guo, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, Hongsheng Li, and Peng Gao. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training, 2022. 2

[55] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16259–16268, 2021. 2

[56] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale, 2023. 2

[57] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyao Zeng, Shanghang Zhang, and Peng Gao. Pointclip v2: Adapting clip for powerful 3d open-world learning, 2022. 2, 5

[58] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2