Article

Not peer-reviewed version

PointBLIP: Zero-Training Point Cloud Classification Network Based on BLIP-2 Model

Yunzhe Xiao , Yong Dou , Shaowu Yang [*]

Posted Date: 8 January 2024

doi: 10.20944/preprints202401.0642.v1

Keywords: point cloud classification; zero-training; large-scale vision-and-language model; zero-shot classification; few-shot classification

Article

PointBLIP: Zero-Training Point Cloud Classification Network Based on BLIP-2 Model

Yunzhe Xiao [1,†], Yong Dou [2,†] and Shaowu Yang [1,*,†]

[1] Institute for Quantum Information & State Key Laboratory of High Performance Computing, College of Computer Science and Technology, National University of Defense Technology; xiaoyunzhe@nudt.edu.cn

[2] National Key Laboratory of Parallel and Distributed Computing, National University of Defense Technology; yongdou@nudt.edu.cn

[*] Correspondence: shaowu.yang@nudt.edu.cn

[†] Current address: Changsha, 410073, China.

Abstract: Leveraging the open-world understanding capacity of large-scale visual-language pre-trained models has become a hot-spot in point cloud classification. Recent approaches rely on transferable visual-language pre-trained models, classifying point clouds by projecting them into 2D images and evaluating consistency with textual prompts. These methods benefit from the robust open-world understanding capabilities of visual-language pre-trained models and require no additional training. However, they face several challenges summarized as prompt ambiguity, image domain gap, view weights confusion, and feature deviation. In response to these challenges, we propose PointBLIP, a zero-training point cloud classification network based on the recently introduced BLIP-2 visual-language model. PointBLIP is adept at processing similarities between multi-images and multi-prompts. We separately introduce a novel method for point cloud zero-shot and few-shot classification, which involves comparing multiple features to achieve effective classification. Simultaneously, we enhance the input data quality for both the image and text sides of PointBLIP. In point cloud zero-shot classification tasks, we outperform state-of-the-art methods on three benchmark datasets. For few-shot classification tasks, to the best of our knowledge, we present the first zero-training few-shot point cloud method, surpassing previous works under the same conditions and showcasing comparable performance to full-training methods.

Keywords: point cloud classification; zero-training, large-scale vision-and-language model; zero-shot classification; few-shot classification

1. Introduction

Point cloud represents one of the most commonly used formats for 3D data, comprising a set of points sampled from a scene. In various 3D computer vision applications, point clouds serve as either the sole data source [1–7] or essential data [8,9] for understanding 3D scenes. Point cloud classification is a fundamental task in 3D scene understanding. Simultaneously, classifying point clouds in an open-world scenario or for unknown new categories is a hot-spot issue [10,11]. Achieving this level of application requires a significant amount of human-labor data annotations for deployed 3D systems. Despite the increasing availability of point cloud data facilitated by the advancement of 3D scanning technologies, the valid point cloud data volume remains significantly insufficient. Besides, annotating point cloud data is notably more challenging compared to 2D image data due to its scattered and unordered nature [10], posing difficulties in collecting point cloud datasets for deep learning applications.

Visual-language pre-trained(VLP) models, learning from image-text pairs [12–14], have revolutionized 2D computer vision recognition over the last few years. Benifit from existing large-scale pre-training data, these models exhibit exceptional understanding of the open world at 2D image level [11,13,14]. Inspired by this, many downstream recognition tasks can be adapted by VLP model, and this extension also applies to the domain of point cloud classification. Some recent works have explored how to transfer knowledge structures to point cloud classification tasks [11,15,16]. Those transferred approaches utilizing VLP models follow a common process: 1) Encoding the projected point cloud images and textual prompt separately as a single feature. 2) Aligning image-text pair feature and determining the most corresponding category by cosine similarity. Typically, point cloud is projected into multi-view depth images and all image feature are aggregated into a single feature with predefined view weights.

We identify several limitations in those VLP-based methods: 1) **Prompt Ambiguity**: The choice of textual prompts for each category may involve predefined templates or generation from a large language model. However, the selection of specific textual prompts for classification relies on human expertise. 2) **Image Domain Gap**: VLP-based methods project point clouds as depth images. Nevertheless, depth images significantly differ from the image domain of the VLP model training dataset. 3) **View Weights Confusion**: Point clouds are often observed from multiple viewpoints during projection as images, and encoded image features are aggregated into a single feature through view weights. Predefined view weights require manual fine-tuning rather than automatic adjustment, making it challenging to determine which viewpoint is more beneficial for classification without prior knowledge. 4) **Feature Deviation**: Encoded features of objects with unique shapes may deviate from common shapes in the same category, as a single image encoder may not focus on their distinctive characteristics that distinguish them from other categories.

In this work, we introduce PointBLIP, a zero-training point cloud classification network. PointBLIP is built upon the BLIP-2 visual-language pre-trained model [14], enabling it to handle zero-shot and few-shot point cloud classification tasks. Unlike previous methods, PointBLIP proposes novel and improved approaches in input data construction and feature similarity comparison to address the aforementioned challenges.

To improve the quality of input data, we employ ray tracing method to render point clouds into simulated images instead of projecting them into depth maps, thereby making the input images more closely aligned with the image domain of VLP models. Simultaneously, we utilize a lasrge language model to generate shape-specific and more discriminative textual prompts. We treat multiple textual prompts for the same object category as a semantic description set for textual input, collectively enriching the descriptive features and eliminating the need for manual selection.

PointBLIP conducts comparisons between multiple image features and multiple text features. The various image features are derived from multiple projection perspectives of the point cloud and the encoding process of the BLIP-2 image encoder. Simultaneously, the multiple text features originate from the semantic description set on the textual side. Diverging from previous approaches that use predefined view weights to aggregate a single feature, we directly compare the similarities between multiple image features and multiple text features without any weight adjustments. We conceptualize the process of comparing multiple features as occurring microscopically in a feature grid. In order to measure reliable feature similarity from feature grid, we employ distinct strategies tailored to zero-shot and few-shot classification tasks.

For zero-shot classification, we propose a Max-Max-Similarity strategy, which entails comparing the maximum similarities between point cloud images and prompts. On the other hand, for few-shot classification, we suggest the Max-Min-Similarity strategy, which involves comparing the most challenging-to-match feature for point cloud images and example images. The selection between these strategies depends on whether the object is compared with features that have a explicit semantic description. It is worth noting that, to our best knowledge, we propose the first zero-training point cloud few-shot classification network by our strategy.

PointBLIP boosts baseline's performance and even surpasses state-of-the-art models. In zero-shot point cloud classification tasks, PointBLIP surpasses state-of-the-art methods by 1% to 3% on three benchmark datasets, including synthetic dataset ModelNet and real scanning dataset ScanObjectNN. In few-shot point cloud classification tasks, PointBLIP shows an enhancement of approximately 20% compared to other VLP-based methods under similar conditions. It is worth noting that as a zero-training model, PointBLIP remains comparable to full-training few-shot classification models on two standard datasets, ModelNet40-FS and ShapeNet70-FS.

Our contributions are summarized as follows:

- We introduce PointBLIP, a zero-training point cloud classification network based on visual-language pre-trained model.
- We improve the input data quality for VLP-based method. By employing ray tracing rendering, we address the gap between point cloud and image data. Additionally, we introduce a shape-specific textual prompt generation method.
- We employ distinct feature measurement strategies tailored to zero-shot and few-shot classification tasks. A Max-Max-Similarity strategy entails comparing the similarities between images and prompts for zero-shot classification task, while a Max-Min-Similarity strategy compare the similarities between point cloud images and example images for few-shot classification task.
- Comprehensive experiments conducted on benchmark datasets demonstrate that PointBLIP surpasses state-of-the-art performance. PointBLIP surpass previous work in both zero-shot and few-shot classification tasks. Moreover, in the few-shot classification task, PointBLIP remains comparable to full-training few-shot classification methods.

2. Related Work

2.1. Vision-Language Pre-Training

The surge in interest in vision-language pre-training (VLP) has given rise to various model architectures specifically designed for multi-modal tasks. Diverse structures, such as dual-encoder [12,17], fusion-encoder [18], and encoder-decoder [19], have emerged to cater to various downstream objectives. Over time, pre-training objectives like image-text contrastive learning [12,20,21], image-text matching [21,22], and masked language modeling [13,23], have converged towards approaches trained on large-scale datasets. VLP models typically undergo end-to-end pre-training on extensive image-text pair datasets, with the “image-to-text” interface becoming a standardized input-output format. This standardization facilitates task-agnostic architectures for zero-shot transfer, eliminating the need for specialized outputs or dataset-specific customization. A widely adopted model, CLIP [12], harnesses VLP for cross-modal knowledge transfer, enabling natural language to comprehend visual concepts.

2.2. Zero-shot Learning in Point Cloud

The objective of zero-shot learning is to identify objects not encountered during the training phase. While extensive attention has been given to 2D classification in zero-shot learning [24,25], few studies have explored its application in the 3D domain. Traditional methods for 3D open-world learning still necessitate 3D training data as a pretraining stage. Pioneering the exploration of zero-shot learning on point clouds, [26] partitions the 3D dataset into “seen” and “unseen” samples. It employs PointNet [27] to train on the former set and evaluates on the latter by measuring cosine similarities with category semantics. Building upon this foundation, [28] addresses the hubness problem [29] stemming from low-quality extracted 3D features, while [30] introduces a triplet loss for enhanced performance in transductive settings. This series of works trains zero-shot classifiers on “seen” 3D categories by maximizing inter-class divergence in the latent space, and subsequently tests on “unseen” ones.

2.3. Few-Shot Learning in Point Cloud

Few-shot learning (FSL) holds great promise in the realm of deep learning due to its ability to generalize well on new tasks despite having limited annotated data. In the customary N-way K-shot Q-query few-shot learning setting [31], the aim of FSL algorithms is to meta-train a predictor that can be generalized to new unlabeled query examples by few labeled support examples. Typically, existing FSL algorithms adopt a meta-learning framework and can be broadly categorized into metric-based and optimization-based methods. Metric-based approaches [32–34] center around learning an embedding space where similar sample pairs are brought closer together or involve designing a metric function to assess the feature similarity between samples. Conversely, optimization-based methods [35–37] treat meta-learning as an optimization process.

Although most current FSL methods operate within the 2D image domain, their application in 3D perception is an under-explored area [10,38]. 3D few-shot learning poses greater challenges due to sparse information in point clouds and smaller-scale annotated data. Additionally, the diverse architectures and learning algorithms further complicate efficacy in the 3D domain. Recent efforts have combined 2D FSL with 3D backbone networks to benchmark few-shot point cloud classification. [38] and [10] present adapted 3D FSL point cloud classification methods derived from typical 2D FSL algorithms, implemented on various point cloud learning architectures.

2.4. VLP-Based Point Cloud Adapted Network

The use of VLP models for open-world point cloud recognition is an emerging research area still in its early stages. Current approaches often adopt the strategy of aligning image-text pair features and determining the most corresponding category, predominantly relying on CLIP [12]. PointCLIP [15] pioneered this approach, aligning the depth maps of projected point clouds with object template sentences to identify the most similar category. However, the sparsely distributed points onto depth values result in scatter-style input images, significantly deviating from real-world pre-training images in both appearance and semantics. Moreover, object template sentences are insufficient for fully describing 3D shapes and negatively impact pre-trained language-image alignment. To address the domain gap between 3D and images, CLIP2Point [16] enforces alignment between depth features and visual CLIP features through an image-depth contrastive learning method. Nevertheless, this process requires additional training and may risk overfitting to the image style of a particular dataset. In contrast, PointCLIPv2 [11] generates CLIP-preferred images through realistic projection, achieved by a series of enhanced operations, ensuring time efficiency and eliminating the need for additional pre-training. Additionally, PointCLIPv2 leverages a large-scale language model [39] to generate text with richer 3D semantics, enhancing the input for the text encoder. However, even though PointCLIPv2 has effectively enhanced the projection quality of point clouds, the resulting images are still evidently far away from the real-world image domain.

These VLP-based point cloud adapted networks follow the common strategy of comparing image-text pair features. They respectively encode all projected images of each point cloud and textual prompt of each category into a single feature. Without prior knowledge, it poses challenges in setting weights for projected viewpoints and selecting textual prompts.

3. Method

In Section 3.1, we conduct a revisit of Bootstrapping Language-Image Pre-training with frozen unimodal models (BLIP-2) and present the essential components upon which PointBLIP relies. In Section 3.2, we delineate the methods employed for enhancing input data quality. In Section 3.3, we elucidate the procedures through which PointBLIP executes zero-shot point cloud classification. Finally, in Section 3.4, we elaborate on the methods employed by PointBLIP for conducting few-shot point cloud classification.

3.1. A Revisit of BLIP-2

BLIP-2 is a versatile and computationally efficient vision-language pre-training method that leverages off-the-shelf pre-trained vision and language models [14]. It comprises two stages: the vision-and-language representation stage and the vision-to-language generative stage. The former aligns image and text representations, while the latter generates linguistic interpretations for images. In this study, we primarily explore the cross-modal capabilities of the vision-and-language representation stage, serving as a feature encoder.

We elucidate the feature encoding process during inference in detail. For each single image, a fixed number of encoded features are extracted from the BLIP-2 image encoder instead of a single feature. The image encoder employs a set number of learnable query embeddings as input, interacting with image features from the frozen CLIP [12] image encoder through cross-attention layers. Subsequently, the image encoder produces a set of learned queries as the feature representation of an image. It is worth noting that, unlike traditional image encoders, BLIP-2 encodes an image into multiple features, with each feature representing a semantic aspect of the image. The image encoding process can be formulated as

$$F_{image} = f_i(I) \in \mathbb{R}^{n \times c}, \tag{1}$$

where I is an input image, n is query number, c is the embedding dimension.

In contrast, the text encoder encodes all words into output embeddings but focuses solely on the [CLS] token as a single classification feature. The text encoding process can be formulated as

$$F_{text} = f_t(T) \in \mathbb{R}^{1 \times c}, \tag{2}$$

where T is a descriptive sentence for corresponding image.

We construct the PointBLIP network using the image and text encoders in BLIP-2. The fundamental distinction between PointBLIP and previous work lies in encoding one image into multiple features, while previous work encodes one image into a single feature. We adopt this approach for the following reasons:

1) Enhancement in feature descriptive capabilities. Encoding into multiple features is advantageous for extracting more extensive and comprehensive information from an image. Multiple features imply that the encoder interprets the image from different semantic perspectives.
2) Advantageous for filtering out irrelevant information. Since the interpretations of multiple features differ, there is an opportunity to independently extract the features of interest while filtering out irrelevant ones. In contrast, the traditional image encoding process encodes all image information, including noise, into a single feature.

3.2. Prompting for Image and Text

To address the 3D model gap and generate meaningful textual prompts, we introduce two novel approaches to constructing input image and textual prompt for our method.

3.2.1. Ray Tracing for Point Cloud

Despite the existence of methods to improve image quality [11], projecting point clouds as depth maps still results in a model gap between point cloud and VLP model training images. Since VLP model training data predominantly originates from the real world, we contend that transforming point clouds into stereoscopic, clear-outlined 2D shapes is necessary. Therefore, we introduce ray tracing method to render point clouds into simulated images.

In this process, each point in the point cloud is represented as a white sphere with a radius of r, and the surface of sphere undergoes diffuse reflection of rays. We use parallel and inclined light sources to illuminate these spheres. To enhance the clarity of the complete outline, rays undergo a finite number of diffuse reflections on the spheres. For each point cloud object, we generate rendered images from four different perspectives around the object to obtain a comprehensive view. A comparison between the projection method and our rendering method is shown in Figure 1.

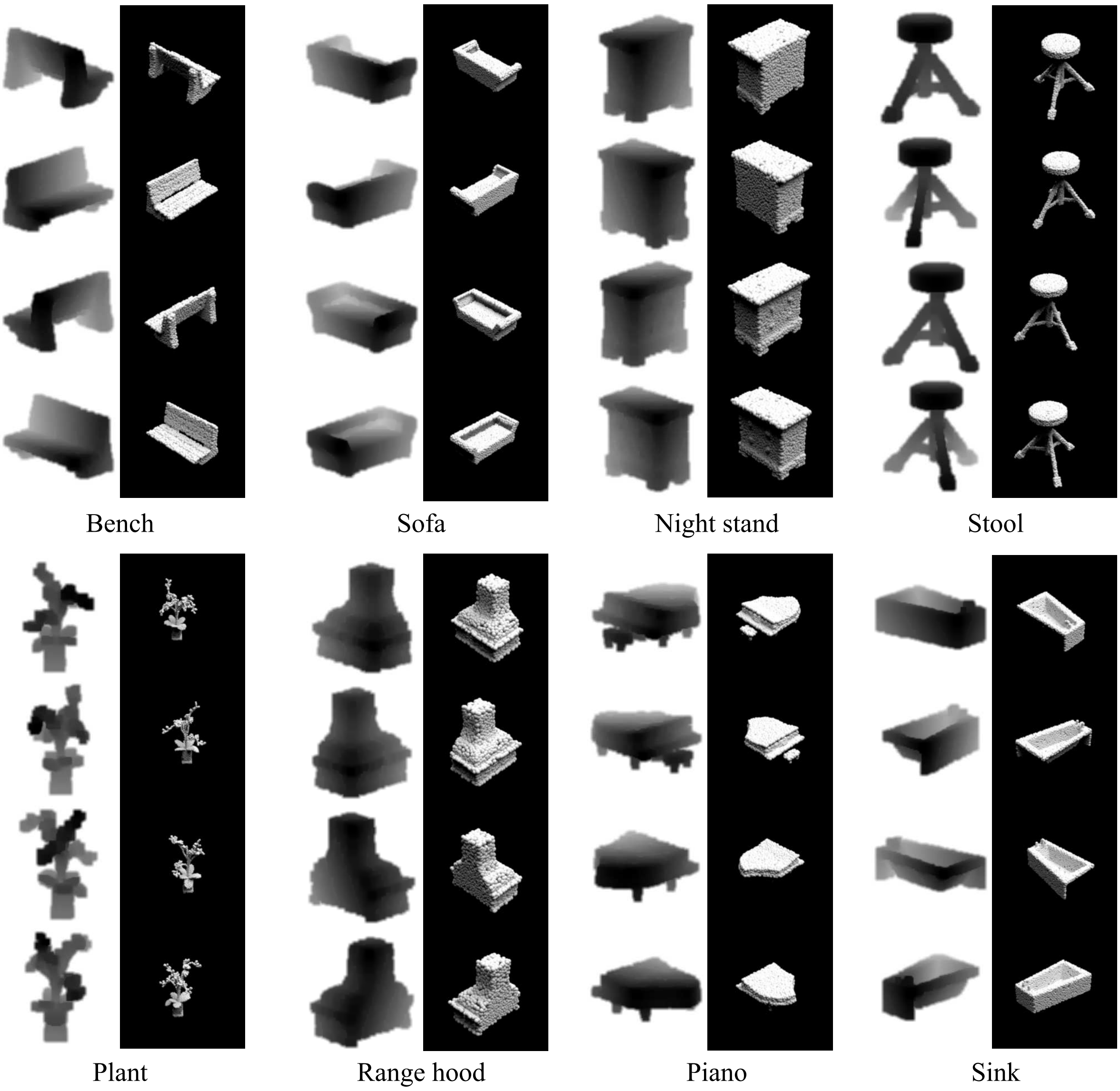


Figure 1. Visualization results comparing Projection vs. Ray Tracing on the ModelNet dataset. The visualizations on the left, with white backgrounds, depict the outcomes obtained through realistic projection in PointCLIPv2, whereas those on the right showcase our visualizations utilizing ray tracing. The point cloud images generated through ray tracing exhibit a closer resemblance to the visual style observed in the real-world scene.

3.2.2. Comparative Textual Prompts

Taking inspiration from PointCLIPv2 [11], we utilize a large language model GPT-3 [39] to generate 3D-specific text with category-wise shape characteristics as textual input. Since original point clouds lack texture information, we argue that textual input should distinguish different categories based on shape. We introduce two rules in command to generate distinctive descriptive sentences: 1) Specify all categories when providing commands to GPT-3 as input. 2) Request GPT-3 to offer the most distinctive shape description.

An example of generate textual prompt for *airplane* category in ModelNet dataset is illustrated as follows:

Question: *The following object categories: airplane, bathtub, bed...(list all category names in ModelNet). Describe the shape differences between the airplane and other categories in one short sentence.*

***GPT-3 Answering**: The airplane stands out with its elongated, winged structure and tail, distinctly different from the predominantly static and boxy forms of the other categories.*

In our work, we generate a set of descriptive sentences for each category as textual prompts and all sentences will be used for the classification of a category. Use "CLASS" as the name for a target category to be classified, we adopt the following three sentence generation templates:

1) CLASS.
2) Answering from GPT-3: The following object categories: ...(list all category names). Describe the shape differences between the CLASS and other categories in one short sentence.
3) Answering from GPT-3: The following object categories: ...(list all category names). Use one sentence to describe In what aspects does CLASS look different from other categories in terms of shape?

3.3. Zero-Shot Point Cloud Classification

The zero-shot point cloud classification process is illustrated in Figure 2. Following the data prompting methods detailed in Section 3.2, we generate simulated point cloud images formed from V observation perspectives and textual prompts for L target categories. The BLIP-2 image encoder produces n learned queries for each perspective image, and the image feature set $\{F_i\}_{i=1}^{V}$ (where $F_i \in \mathbb{R}^{n\times c}$) is extracted from one point cloud, where c is the embedding dimension. For the textual branch, assuming each category contains q textual prompts, the text feature set $\{W_j\}_{j=1}^{L}$ (where $W_j \in \mathbb{R}^{q\times c}$) is extracted from the textual prompts for all categories. Our objective is to determine the most likely category for the source point cloud.

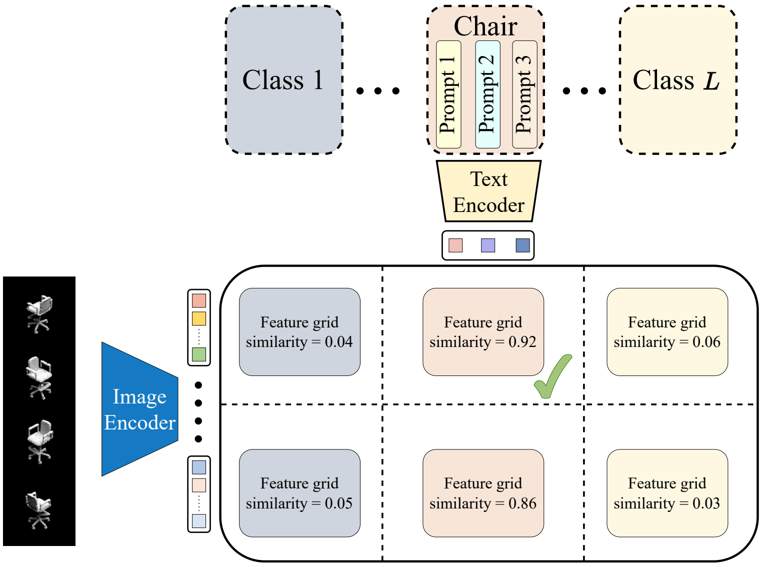

Class 1
Chair
Prompt 1
Prompt 2
Prompt 3
Class L
Text Encoder
Image Encoder
Feature grid similarity = 0.04
Feature grid similarity = 0.92
Feature grid similarity = 0.06
Feature grid similarity = 0.05
Feature grid similarity = 0.86
Feature grid similarity = 0.03

Figure 2. Overall architecture of PointBLIP for zero-shot classification. Each feature grid generates a similarity score by comparing a perspective image with all textual prompts corresponding to a specific category. The classification result is determined by selecting the category with the highest similarity score. Both image and text encoders employed in this architecture are derived from BLIP-2.

Previous methods typically compare the cosine similarity between single and aggregated features of images and text, *e.g.*, PointCLIPv2 calculates the final zero-shot classification logits by weighted summing multi-view image features to a single feature, formulated as

$$logits = (\omega F_o)W_o^T \in \mathbb{R}^{1\times L}, \tag{3}$$

where $F_o \in \mathbb{R}^{V\times c}$, $W_o \in \mathbb{R}^{L\times c}$, and $\omega \in \mathbb{R}^{1\times V}$ represents the view weights. However, our approach contains additional dimension for feature comparison. $\{F_i\}_{i=1}^{V}$ and $\{W_j\}_{j=1}^{L}$ cannot be directly used to calculate similarity following Equation (3), and ω may cause view weights confusion without prior knowledge.

We take two steps to address this issue. Firstly, we establish a minimal unit called **feature grid** between the image feature set $\{F_i\}_{i=1}^{V}$ and the text features set $\{W_j\}_{j=1}^{L}$. For zero-shot classification task, we define feature grid as a similarity matrix comparing the cosine similarity between image features from certain perspective and all text features for a specific category, formulated as

$$G_{ij} = F_i W_j^T. \tag{4}$$

The feature grid is represented as a cosine similarity matrix. For L classification task, each point cloud can generate $V \times L$ feature grids. We employ a strategy called Max-Max-Similarity to calculate similarity from each feature grid. The process of the Max-Max-Similarity strategy is illustrated in Figure 3(a). Max-Max-Similarity calculates the maximum values for both rows and columns in the feature grid matrix, which will be treated as the basis for the next classification step. The aim of Max-Max-Similarity is to provide the maximum similarity level between a simulated point cloud image and a specific category.

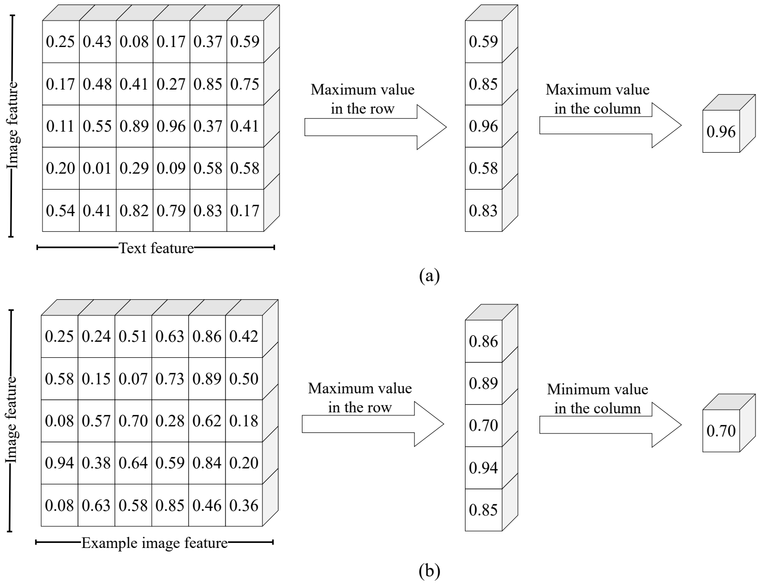
Secondly, we from a larger similarity matrix G formed by similarity values from feature grids to obtain the most relevant category, which can be formulated as

$$G = \{ColMax(RowMax(G_{ij}))\} \in \mathbb{R}^{V \times L}. \tag{5}$$

We search the maximum value in matrix G and take the category corresponding to this maximum value as the classification result, formulated as

$$PredIndex = SCM(G), \tag{6}$$

where $SCM(\cdot)$ represents the function searching for the category index of the maximum value in the matrix.



0.25	0.43	0.08	0.17	0.37	0.59
0.17	0.48	0.41	0.27	0.85	0.75
0.11	0.55	0.89	0.96	0.37	0.41
0.20	0.01	0.29	0.09	0.58	0.58
0.54	0.41	0.82	0.79	0.83	0.17

0.59
0.85
0.96
0.58
0.83

(a)

0.25	0.24	0.51	0.63	0.86	0.42
0.58	0.15	0.07	0.73	0.89	0.50
0.08	0.57	0.70	0.28	0.62	0.18
0.94	0.38	0.64	0.59	0.84	0.20
0.08	0.63	0.58	0.85	0.46	0.36

0.86
0.89
0.70
0.94
0.85

(b)

Figure 3. Different feature measurement strategies in the feature grid. Each cube represents the cosine similarity between two features. (**a**) Max-Max-Similarity strategy. The output similarity is the maximum similarity for both rows and columns in the feature grid. (**b**) Max-Min-Similarity strategy. The output similarity is the minimum value among the maximum similarities in each row of the feature grid.

3.4. Few-Shot Point Cloud Classification

The process of few-shot point cloud classification is illustrated in Figure 4. In this scenario, a limited number of point clouds from "unseen" categories are labeled as a reference, and our method aims to recognize new, unlabeled query point clouds under this few-shot condition. The labeled example point clouds constitute the support set $\mathcal{S} = \{P_i\}_{i=1}^{N \times K}$, encompassing N classes with K examples for each class, where P_i represents an example point cloud. Our objective is to identify unlabeled point clouds based on the support set $\mathcal{S}$.

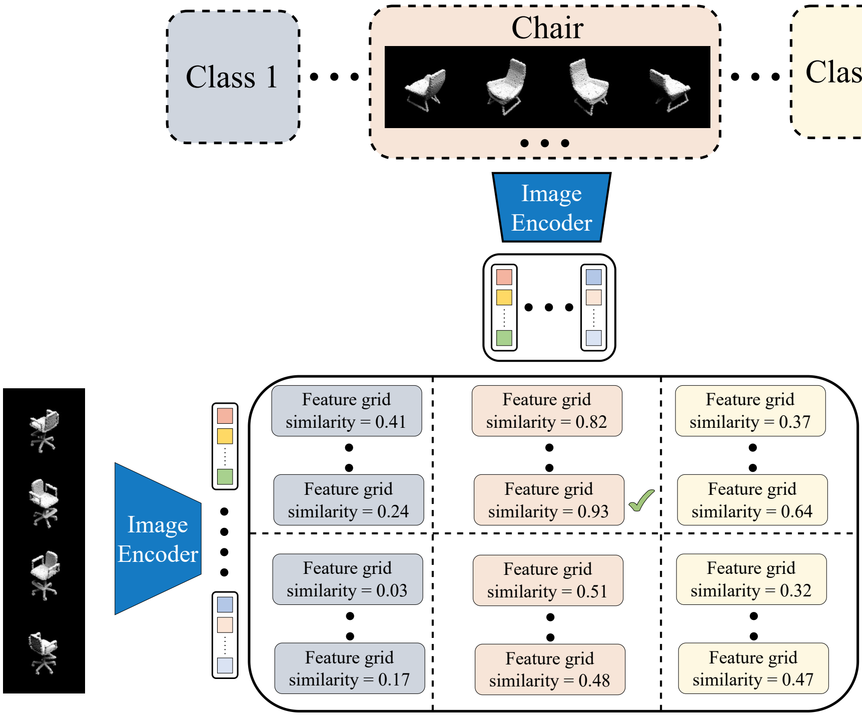

Class 1
Chair
Class N
Image Encoder
Image Encoder
Feature grid similarity = 0.41
Feature grid similarity = 0.82
Feature grid similarity = 0.37
Feature grid similarity = 0.24
Feature grid similarity = 0.93
Feature grid similarity = 0.64
Feature grid similarity = 0.03
Feature grid similarity = 0.51
Feature grid similarity = 0.32
Feature grid similarity = 0.17
Feature grid similarity = 0.48
Feature grid similarity = 0.47

Figure 4. Overall architecture of PointBLIP for few-shot classification. Each feature grid contributes to a similarity score through the comparison of a query image with an example image. The category associated with the feature grid exhibiting the highest similarity is designated as the classification result. Both the image and text encoders incorporated in this structure are derived from BLIP-2.

For an unlabeled query point cloud, we generate simulated images from V observation perspectives of this point cloud and extract image features using the BLIP-2 image encoder as $\{F_i\}_{i=1}^{V}$ (where $F_i \in \mathbb{R}^{n \times c}$). Simultaneously, example images generated from the support set $\mathcal{S}$ are encoded into $\{Z_j\}_{j=1}^{N \times K \times V}$ (where $Z_j \in \mathbb{R}^{n \times c}$) following the same process. In the traditional few-shot classification procedure, we need to calculate the feature similarity between query images and example images for matching. However, the BLIP-2 image encoding process introduces a challenge. Since each query and example image produces n features instead of a single feature, we need to measure a unique similarity among two sets of image features. Some image features may potentially describe irrelevant information, such as background and texture, which is not suitable for comparison. There is no explicit way to determine which feature represents category-discriminative characteristics and is beneficial for comparison.

To address this issue, we also establish a feature grid to compare multiple image features. In the few-shot classification task, the construction of the feature grid and measurement strategy differ from the zero-shot method in Section 3.3. This process is based on the following theory: if two objects belong to the same category, the similarity of their most challenging-to-match feature will still be higher than for other categories.

Specifically, we define this feature grid as a similarity matrix comparing the cosine similarity between query and example image features, formulated as

$$G_{ij} = F_i Z_j^T. \tag{7}$$

For each query point cloud in the classification process, $V \times N \times K \times V$ feature grids can be constructed. We calculate a similarity from each feature grid, representing the matching degree between the query image and example image. We employ a Max-Min-Similarity strategy to calculate this similarity. The process of the Max-Min-Similarity strategy is illustrated in Figure 3b. We calculate the maximum value for each row in the feature grid, representing the maximum matching level between each query image feature and example image features. Then, we compute the minimum value from this collection of maximum values, representing the maximum similarity level of the most challenging-to-match features. We treat this value as the similarity between the query image and example image.

Next, we utilize the similarities from feature grids to form a matrix G reflecting the similarities between all query images and all example images. The whole process can be formulated as

$$G = \{ColMin(RowMax(G_{ij}))\} \in \mathbb{R}^{V \times N \times K \times V}. \tag{8}$$

Finally, we search for the maximum value in matrix G and take the category corresponding to this maximum value as the classification result, which can be formulated as

$$PredIndex = SCM(G), \tag{9}$$

where $SCM(\cdot)$ represents the function searching for the category index of the maximum value in G.

4. Results

In this section, we first illustrate the implementation details of PointBLIP in Section 4.1 and evaluation dataset in Section 4.2. Then, we present the performance of PointBLIP on zero-shot classification in Section 4.3 and few-shot classification in Section 4.4. Finally, we conduct an ablation experiment in Section 4.5.

4.1. Implementation Details

We utilize Mitsuba software [40] for rendering point clouds into simulated images. Each point in the point cloud is represented as a sphere with a radius r, and the surface of the sphere is set as a white and diffuse surface. The value of r is determined based on the specific dataset. We render the point cloud from four views around object, creating images with a resolution of 224×224. A directional light source is added for each perspective, and the light rays undergo three reflections on the surface of the sphere. Figure 1 shows some instances from different categories.

4.2. Evaluation Dataset

We evaluate the performance of PointNLIP on several widely used benchmark datasets, including synthetic and real scan datasets.

ModelNet dataset. ModelNet [41] is a large-scale 3D CAD dataset containing 12,311 CAD models from 40 categories. ModelNet includes two subsets, ModelNet10 and ModelNet40, for classification tasks. ModelNet10 contains 4,899 CAD models from 10 categories, with 3,991 for training and 908 for testing. ModelNet40 contains 12,311 CAD models from 40 categories, with 9,843 for training and 2,468 for testing. We only apply its test set data since PointBLIP is a zero-training network.

ScanObjectNN dataset. ScanObjectNN [42] is a real-world dataset containing 2,902 samples of point cloud data from 15 categories. Unlike clean CAD models in ModelNet, objects in ScanObjectNN are partially presented and attached with backgrounds. Thus, it is more challenging than ModelNet. We test PointBLIP on ScanObjectNN under three data splits: S-OBJ_ONLY includes only ground truth segmented objects extracted from the scene; S-OBJ_BG includes objects attached with backgrounds; S-PB_T50_RS contains rotation, scaling, perturbations, and shifting the bounding box as the hardest split.

ModelNet40-FS dataset. ModelNet40-FS [38] is a new split of ModelNet40 [41], containing 30 training classes with 9,204 examples and 10 disjoint testing classes with 3,104 examples. This splitting of the raw dataset according to categories is done for few-shot classification evaluation.

ShapeNet70-FS dataset. ShapeNet70-FS [38] is adapted from ShapeNetCore and has a larger number of data than ModelNet40-FS, totaling 30,073 examples, with 50 classes having 21,722 samples for training and 20 classes with 8,351 samples for testing. ShapeNet70-FS is a benchmark dataset for few-shot classification evaluation.

4.3. Zero-Shot Point Cloud Classification

We evaluate PointBLIP on three widely-used benchmarks for zero-shot classification: synthetic datasets ModelNet and real scanning dataset ScanObjectNN. Two splits(ModelNet10, ModelNet40) in ModelNet and three splits(S-OBJ_ONLY, S-OBJ_BG, and S-PB_T50_RS) in ScanObjectNN, will be tested. Following the zero-shot principle, we directly test the classification performance on the full test set without learning from the training set. We render point clouds following the method in Section 3.2. In the ModelNet datasets, we set the value of r as the average minimum distance between points. In the ScanObjectNN dataset, we set the value of r as 2.5 times the average minimum distance between points. This choice of r is determined by the characteristics of the data distribution. The point distribution in ScanObjectNN is more sparse, while the point distribution in ModelNet is more balanced. Additionally, we generate textual prompts for target categories following the method outlined in Section 3.2. Each category includes three textual prompts.

We compare our method with several recent zero-shot point cloud classification methods, and the results are presented in Table 1. We use overall classification accuracy(%) as the experimental metric. PointCLIPv2 is currently the state-of-the-art method. We outperform PointCLIPv2 by 0.88%, 2.03%, 1.03%, 3.01%, and 3.54% in classification accuracy on ModelNet10, ModelNet40, S-OBJ_ONLY, S-OBJ_BG, and S-PB_T50_RS datasets, respectively.

Table 1. Zero-shot point cloud overall classification (%) for ModelNet and ScanObjectNN benchmark datasets. ModelNet10 and ModelNet40 are two data splits in ModelNet, S-OBJ_ONLY, S-OBJ_BG, and S-PB_T50_RS are three data splits in ScanObjectNN.

Method	ModelNet10	ModelNet40	S-OBJ_ONLY	S-OBJ_BG	S-PB_T50_RS
CLIP2Point [16]	66.63	49.38	35.46	30.46	23.32
Cheraghian [30]	68.50	-	-	-	-
PointCLIP [15]	30.23	23.78	21.34	19.28	15.38
PointCLIPv2 [11]	73.13	64.22	50.09	41.22	35.36
PointBLIP(Ours)	**74.01**	**66.25**	**51.12**	**44.23**	**38.90**

4.4. Few-Shot Point Cloud Classification

4.4.1. Comparison with Full-training Work

We compare the zero-training PointBLIP with some full-training methods. The current study on 3D data is relatively under-explored, but few-shot classification methods are well-established and diverse in 2D image tasks [38]. Consequently, we assess the performance of point cloud adaptations of the current state-of-the-art methods in 2D image few-shot classification.

Following the experimental procedures outlined in [10], we substitute the backbone networks of these 2D few-shot classification methods with the mainstream DGCNN network [43] designed for processing point clouds. Subsequently, we assess their classification performance in comparison with PointBLIP on the ModelNet40-FS and ShapeNet70-FS datasets. We respectively compare the performance of these methods under 5-way 1-shot and 5-way 5-shot settings, and report the mean classification results of the 700 episodes with 95% confidence intervals. The experimental results are shown in Table 2.

Table 2. Few-shot point cloud classification results with 95% confidence intervals on ModelNet40-FS and ShapeNet70-FS. Prior methods are trained with DGCNN as a backbone, while PointBLIP is zero-training.

Method	ModelNet40-FS		ShapeNet70-FS	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
ProtoNet [32]	69.95 ± 0.67	85.51 ± 0.52	69.03 ± 0.84	82.08 ± 0.72
Relation Net [44]	68.57 ± 0.73	82.01 ± 0.53	67.87 ± 0.86	77.99 ± 0.70
FSLGNN [45]	61.96 ± 0.76	80.22 ± 0.55	66.25 ± 0.88	76.20 ± 0.77
Meta-learner [37]	59.08 ± 0.86	76.99 ± 0.67	64.53 ± 0.83	74.61 ± 0.72
MAML [46]	62.57 ± 0.88	77.41± 0.73	64.39 ± 0.76	74.11 ± 0.68
MetaOptNet [46]	67.05 ± 0.78	85.05 ± 0.59	68.27 ± 0.93	81.06 ± 0.76
S2M2 [47]	69.73 ± 0.64	83.25 ± 0.43	68.53 ± 0.73	79.71 ± 0.73
Meta-Baseline [48]	71.33 ± 0.34	85.27 ± 0.23	70.16 ± 0.41	81.08 ± 0.33
SimpleTrans [49]	71.44 ± 0.33	86.78 ± 0.22	69.19 ± 0.40	**83.37 ± 0.32**
Sharma *et al.* [50]	64.89 ± 0.82	79.59 ± 0.73	65.76 ± 0.72	79.19 ± 0.71
LSSB(SimpleShot+SB) [51]	63.33 ± 0.75	76.41 ± 0.68	64.45 ± 0.83	73.77 ± 0.73
Point-BERT [52]	69.41 ± 3.16	**86.83 ± 2.03**	73.92 ± 3.60	82.86 ± 2.92
Feng *et al.* [53]	61.36 ± 0.41	73.20 ± 0.31	65.09 ± 0.44	75.89 ± 0.29
PointBilp(Ours)	**72.20 ± 1.00**	78.16 ± 1.05	**74.33 ± 1.05**	80.01 ± 0.97

As indicated in Table 2, our method consistently outperforms other approaches in the 5-way 1-shot setting on both the ModelNet40-FS and ShapeNet70-FS datasets. Notably, PointBLIP is zero-training, while the other methods undergo full training. Despite this, we consistently achieve superior performance compared to these fully trained methods. However, in the 5-way 5-shot setting, although PointBLIP demonstrates improvement compared to the 5-way 1-shot setting, its performance is relatively weaker compared to other methods.

4.4.2. Comparison with Proximity Work

To the best of our knowledge, we are pioneers in introducing a zero-training point cloud few-shot classification network. Given the absence of similar work to serve as a reference, we conduct a comparative analysis with the most relevant methods, PointCLIP [15] and PointCLIPv2 [11], under identical conditions. PointCLIP and PointCLIPv2 tackle few-shot classification challenges by incorporating an trainable inter-view adapter, aiming to fine-tune the original output features after pre-training. To ensure a fair comparison, we eliminate the inter-view adapter module, ensuring that all methods are evaluated in an untrained state.

Following the experimental procedures outlined in PointCLIP and PointCLIPv2, we assess the K-shot classification performance on the ModelNet40 and ScanObjectNN (S-PB_T50_RS) datasets, where $K \in \{1, 2, 4, 8, 16\}$. For K-shot scenarios, we randomly sample K point clouds from each category in the training set, employing these point clouds as examples for classification in the testing set. The comparison results with PointCLIP and PointCLIPv2 under different K values are illustrated in Figure 5.

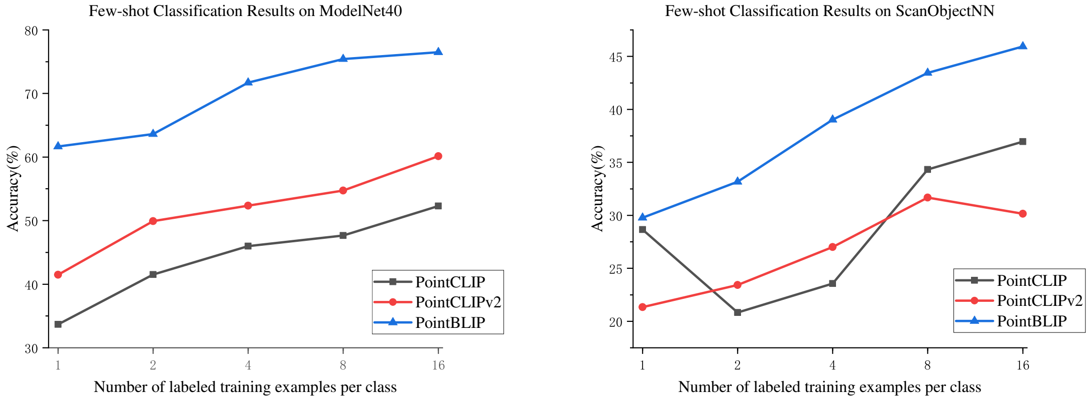


Figure 5. Comparison of zero-training few-shot classification performance between PointBLIP, PointCLIP, PointCLIPv2 on benchmark datasets ModelNet40 (left) and ScanObjectNN (right). The trainable inter-view adapter modules in PointCLIP and PointCLIPv2 have been excluded for a fair evaluation.

As shown, PointBLIP outperforms PointCLIP and PointCLIPv2 under zero-training conditions, demonstrating a significantly superior performance. On the ModelNet40 dataset, PointBLIP achieves an average increase of over 15% in classification accuracy for different values of K. On the real scan dataset ScanObjectNN, where the data is more complex than synthetic data, PointBLIP exhibits a decrease in performance compared to ModelNet40. However, it still maintains an average classification accuracy advantage of nearly 10% over PointCLIP and PointCLIPv2.

4.5. Ablation Study

4.5.1. Input Data Prompting

To assess the impact of our prompting approach on data quality, we perform an ablation study focusing on the prompting process of image and text generation. In the case of image input, we substitute our ray tracing rendering process with the realistic projection method used in PointCLIPv2 [11] as a reference. The realistic projection method involves additional processes, such as densification and smoothing, during point cloud projection, resulting in depth maps from ten perspectives that already capture the object's outline. For text input, we replace our method with only category names.

We first conduct ablation experiments on zero-shot classification, involving both image and text generation. The evaluations are conducted on the ModelNet40 dataset, and the experimental results are presented in Table 3.

Table 3. Ablation study on ModelNet40 zero-shot classification (%) with variations in ray tracing rendering and textual prompts. "×" indicates the substitution with the realistic projection for image generation or the use of category names for textual prompts. "✓" signifies the inclusion of our data prompting.

Ray tracing	Textual prompts	Acc
×	×	52.84
×	✓	52.76
✓	×	56.97
✓	✓	**66.25**

As presented in Table 3, the concurrent application of our data prompting method result in an increase in classification accuracy from 52.84% to 66.25%, indicating an improvement of 13.41%.

doi:10.20944/preprints202401.0642.v1

However, when employing the prompting method solely on the image side, the accuracy experience a modest increase of 4.13%. Similarly, deploying only the prompting method on the text side maintain the accuracy at the same level, but a notable increase of almost 10% in accuracy is observed after incorporating image data prompting.

Next, we conduct ablation experiments on few-shot classification, which only involves image generation. We evaluate on the ModelNet40-FS dataset under the 5-ways 1-shot setting, and the corresponding experimental results are presented in Table 4. Our image prompting method can improve by nearly 2.5% in few-shot classification.

Table 4. Ablation study of image generation methods on ModelNet40-FS 5-ways 1-shot classification (%).

Image generation method	95% confidence intervals
Realistic projection	69.69 ± 1.01
Ray tracing	**72.20 ± 1.00**

4.5.2. Measurement Strategy

To evaluate the impact of the feature grid measurement strategy, we conduct an ablation study on both zero-shot and few-shot classification. As a reference, we replace the original strategy with one that calculates the average value in the feature grid as the final similarity. We conduct separate evaluations to assess the effects of the Max-Max-Similarity strategy in zero-shot classification and the Max-Min-Similarity strategy in few-shot classification.

We conduct additional zero-shot classification tests on all benchmark datasets mentioned in Section 4.3, and the corresponding experimental results are outlined in Table 5. Notably, Max-Max-Similarity consistently outperforms average similarity across all benchmark datasets, resulting in an improvement of 3% to 6% in classification accuracy in the zero-shot classification task.

Table 5. Comparison of average similarity and Max-Max-Similarity strategy on various benchmark datasets for zero-shot classification (%).

Dataset	Average Similarity	Max-Max-Similarity
ModelNet10	70.15	**74.01**
ModelNet40	63.01	**66.25**
S-OBJ_ONLY	45.27	**51.12**
S-OBJ_BG	39.41	**44.23**
S-PB_T50_RS	35.36	**38.90**

We further validate few-shot classification on the benchmark datasets discussed in Section 4.4. The evaluations were performed under the 5-ways 5-shot setting, and the results, along with 95% confidence intervals, are presented in Table 6. It is evident that Max-Min-Similarity surpasses average similarity in the context of few-shot classification, exhibiting a more pronounced enhancement, particularly on the ShapeNet70-FS dataset.

Table 6. Comparison of average similarity and Max-Min-Similarity strategy on various benchmark datasets for few-shot classification under 5-ways 5-shot setting (95% confidence intervals).

Dataset	Average Similarity	Max-Min-Similarity
ModelNet40-FS	77.57 ± 1.06	**78.16 ± 1.05**
ShapeNet70-FS	78.37 ± 0.96	**80.01 ± 0.97**

5. Discussion

The experimental results on benchmarks presented in Section 4 demonstrate that our approach achieves state-of-the-art performance in point cloud classification. Despite surpassing prior work, several issues merit discussion.

5.1. Backbone Network Differences

In comparison to closely related VLP-based methods, our backbone VLP network differs. Most current relevant works employ CLIP [12] as the backbone network, while we utilize BLIP-2 [14]. This raises the question of whether the improved performance of PointBLIP is attributable to a stronger feature learning capability of the VLP backbone model. To investigate this, we refer to experiments in Table 3. In the scenario presented in the second row of Table 3, we use the realistic projection method for image generation and generate textual prompts using GPT-3, similar to PointCLIPv2 [11] except for differences in the backbone network and feature measurement strategy. However, PointBLIP achieves a zero-shot classification accuracy of 52.76%, while PointCLIPv2 achieves a higher classification accuracy of 64.22%. In this scenario, the feature extraction capability of the base model is a determining factor for classification performance, but BLIP-2 performs worse than CLIP. We argue that the performance improvement of PointBLIP does not rely on the feature extraction capability of the base VLP model.

5.2. Feature Grid Measurement

In both zero-shot and few-shot classification tasks, we establish a feature grid to measure feature similarity. The measurement strategies for the feature grid in different tasks are configured based on comparison targets. In zero-shot classification, we compare point cloud images with textual prompts. Due to textual prompt explicitly describing object characteristics and being encoded as a single feature, we use Max-Max-Similarity to find the most straightforward feature similarity between image and textual prompts. In few-shot classification, where example images are encoded as multiple features, some image features may potentially describe irrelevant information. To exclude noise interference, we use Max-Min-Similarity to find the similarity level between the most challenging-to-match features. From Tables 5 and 6, it can be observed that compared to a simple averaging, the similarity reflected by our strategy in the feature grid is more advantageous for distinguishing.

5.3. Viewpoint Weights

Another advantage of PointBLIP is the absence of manually setting weights for different viewpoints. We opt to search for the category with the maximum similarity from the feature grid, which is advantageous for identifying the most likely similar category. We posit that simulated point cloud images from some perspectives may not perfectly align with the textual prompts or example images, introducing noise perturbations during the weighting process. The strategy of searching for the maximum similarity captures the maximum similarity characteristic, thereby avoiding interference from other viewpoints on the overall confidence. Furthermore, it eliminates the need for manually setting viewpoint weights.

6. Conclusions

We introduce PointBLIP, a zero-training and powerful point cloud classification network that achieves state-of-the-art performance in both zero-shot and few-shot classification tasks. Built upon the vision-language pre-training model BLIP-2 as a backbone network, PointBLIP directly compares similarity between multiple image features or multiple text features without the need for pre-setting weights for observed viewpoints. We establish a minimal feature comparison unit called feature grid and employ different feature measurement strategies for zero-shot and few-shot classification tasks. Additionally, we enhance the input data quality by generating images through ray tracing and utilizing GPT-3 to generate comparative textual prompts. The innovations in PointBLIP address challenges such as prompt ambiguity, image domain gap, view weights confusion, and feature deviation observed in previous VLP-based classification methods, resulting in higher classification accuracy on benchmark datasets.

Author Contributions: Conceptualization, Y.X. and Y.D.; methodology, Y.X.; software, Y.X.; validation, Y.X., Y.D. and S.Y.; formal analysis, S.Y.; investigation, Y.X.; resources, Y.X.; data curation, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, Y.D. and S.Y.; visualization, Y.X.; supervision, Y.D. and S.Y.; project administration, S.Y.; funding acquisition, S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by National Key R&D Program of China grant number No. 2021ZD0140301.

Data Availability Statement: All datasets used in this study are publicly available. The ModelNet dataset is available at ShapeNet (https://3dshapenets.cs.princeton.edu/), the ScanObjectNN dataset is available at Github (https://hkust-vgd.github.io/scanobjectnn/), the ModelNet40-FS and ShapeNet70-FS datasetis available at Github (https://github.com/cgye96/A_Closer_Look_At_3DFSL).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

VLP	Vision-Language Pre-training
FSL	Few-shot learning

References

1. Wu, W.; Qi, Z.; Fuxin, L. Pointconv: Deep convolutional networks on 3d point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019; pp. 9621–9630.
2. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019; pp. 6411–6420.
3. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; pp. 16259–16268.
4. Yu, H.; Li, F.; Saleh, M.; Busam, B.; Ilic, S. CoFiNet: Reliable Coarse-to-fine Correspondences for Robust PointCloud Registration. *Adv. Neural Inf. Process. Syst.* **2021**, *34*.
5. Sheng, H.; Cai, S.; Liu, Y.; Deng, B.; Huang, J.; Hua, X.S.; Zhao, M.J. Improving 3d object detection with channel-wise transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; pp. 2743–2752.
6. Mazur, K.; Lempitsky, V. Cloud transformers: A universal approach to point cloud processing tasks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; pp. 10715–10724.
7. Qin, Z.; Yu, H.; Wang, C.; Guo, Y.; Peng, Y.; Xu, K. Geometric transformer for fast and robust point cloud registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 11143–11152.
8. Afham, M.; Dissanayake, I.; Dissanayake, D.; Dharmasiri, A.; Thilakarathna, K.; Rodrigo, R. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 9902–9912.
9. Chen, A.; Zhang, K.; Zhang, R.; Wang, Z.; Lu, Y.; Guo, Y.; Zhang, S. Pimae: Point cloud and image interactive masked autoencoders for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023; pp. 5291–5301.
10. Ye, C.; Zhu, H.; Zhang, B.; Chen, T. A Closer Look at Few-Shot 3D Point Cloud Classification. *Int. J. Comput. Vis.* **2023**, *131*, 772–795.
11. Zhu, X.; Zhang, R.; He, B.; Guo, Z.; Zeng, Z.; Qin, Z.; Zhang, S.; Gao, P. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023; pp. 2639–2650.
12. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; others. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning. PMLR, 2021; pp. 8748–8763.

13. Li, J.; Li, D.; Xiong, C.; Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Proceedings of the International Conference on Machine Learning. PMLR, 2022; pp. 12888–12900.
14. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In Proceedings of the ICML, 2023.
15. Zhang, R.; Guo, Z.; Zhang, W.; Li, K.; Miao, X.; Cui, B.; Qiao, Y.; Gao, P.; Li, H. Pointclip: Point cloud understanding by clip. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 8552–8562.
16. Huang, T.; Dong, B.; Yang, Y.; Huang, X.; Lau, R.W.; Ouyang, W.; Zuo, W. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023; pp. 22157–22167.
17. Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.H.; Li, Z.; Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In Proceedings of the International Conference on Machine Learning. PMLR, 2021; pp. 4904–4916.
18. Tan, H.; Bansal, M. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, 2019.
19. Cho, J.; Lei, J.; Tan, H.; Bansal, M. Unifying vision-and-language tasks via text generation. In Proceedings of the International Conference on Machine Learning. PMLR, 2021; pp. 1931–1942.
20. Yao, L.; Huang, R.; Hou, L.; Lu, G.; Niu, M.; Xu, H.; Liang, X.; Li, Z.; Jiang, X.; Xu, C. FILIP: Fine-grained Interactive Language-Image Pre-Training. In Proceedings of the International Conference on Learning Representations, 2021.
21. Li, J.; Selvaraju, R.; Gotmare, A.; Joty, S.; Xiong, C.; Hoi, S.C.H. Align before fuse: Vision and language representation learning with momentum distillation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 9694–9705.
22. Bao, H.; Wang, W.; Dong, L.; Liu, Q.; Mohammed, O.K.; Aggarwal, K.; Som, S.; Piao, S.; Wei, F. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 32897–32912.
23. Wang, W.; Bao, H.; Dong, L.; Bjorck, J.; Peng, Z.; Liu, Q.; Aggarwal, K.; Mohammed, O.K.; Singhal, S.; Som, S.; others. Image as a Foreign Language: BEiT Pretraining for Vision and Vision-Language Tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023; pp. 19175–19186.
24. Xian, Y.; Akata, Z.; Sharma, G.; Nguyen, Q.; Hein, M.; Schiele, B. Latent embeddings for zero-shot classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016; pp. 69–77.
25. Karessli, N.; Akata, Z.; Schiele, B.; Bulling, A. Gaze embeddings for zero-shot image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017; pp. 4525–4534.
26. Cheraghian, A.; Rahman, S.; Petersson, L. Zero-shot learning of 3d point cloud objects. 2019 16th International Conference on Machine Vision Applications (MVA). IEEE, 2019; pp. 1–6.
27. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017; pp. 652–660.
28. Cheraghian, A.; Rahman, S.; Campbell, D.; Petersson, L. Mitigating the Hubness Problem for Zero-Shot Learning of 3D Objects. In Proceedings of the British Machine Vision Conference, 2019.
29. Zhang, L.; Xiang, T.; Gong, S. Learning a deep embedding model for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017; pp. 2021–2030.
30. Cheraghian, A.; Rahman, S.; Chowdhury, T.F.; Campbell, D.; Petersson, L. Zero-shot learning on 3d point cloud objects and beyond. *Int. J. Comput. Vis.* **2022**, *130*, 2364–2384.
31. Chen, W.Y.; Liu, Y.C.; Kira, Z.; Wang, Y.C.F.; Huang, J.B. A Closer Look at Few-shot Classification. In Proceedings of the International Conference on Learning Representations, 2019.
32. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.

33. Li, W.; Wang, L.; Xu, J.; Huo, J.; Gao, Y.; Luo, J. Revisiting local descriptor based image-to-class measure for few-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019; pp. 7260–7268.
34. Oreshkin, B.; Rodríguez López, P.; Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
35. Lee, K.; Maji, S.; Ravichandran, A.; Soatto, S. Meta-learning with differentiable convex optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019; pp. 10657–10665.
36. Rusu, A.A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; Hadsell, R. Meta-Learning with Latent Embedding Optimization. In Proceedings of the International Conference on Learning Representations, 2018.
37. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the International Conference on Learning Representations, 2016.
38. Ye, C.; Zhu, H.; Liao, Y.; Zhang, Y.; Chen, T.; Fan, J. What makes for effective few-shot point cloud classification? In Proceedings of the IEEE/CVF winter Conference on Applications of Computer Vision, 2022; pp. 1829–1838.
39. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; others. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
40. Jakob, W.; Speierer, S.; Roussel, N.; Nimier-David, M.; Vicini, D.; Zeltner, T.; Nicolet, B.; Crespo, M.; Leroy, V.; Zhang, Z. Mitsuba 3 renderer. 2022. Available online: https://mitsuba-renderer.org.
41. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015; pp. 1912–1920.
42. Uy, M.A.; Pham, Q.H.; Hua, B.S.; Nguyen, T.; Yeung, S.K. 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, 2019; pp. 1588–1597.
43. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph. (tog)* **2019**, *38*, 1–12.
44. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018; pp. 1199–1208.
45. Satorras, V.G.; Estrach, J.B. Few-shot learning with graph neural networks. International Conference on Learning Representations, 2018.
46. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning. PMLR, 2017; pp. 1126–1135.
47. Mangla, P.; Kumari, N.; Sinha, A.; Singh, M.; Krishnamurthy, B.; Balasubramanian, V.N. Charting the right manifold: Manifold mixup for few-shot learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020; pp. 2218–2227.
48. Chen, Y.; Liu, Z.; Xu, H.; Darrell, T.; Wang, X. Meta-baseline: Exploring simple meta-learning for few-shot learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; pp. 9062–9071.
49. Luo, X.; Xu, J.; Xu, Z. Channel importance matters in few-shot image classification. In Proceedings of the International Conference on Machine Learning. PMLR, 2022; pp. 14542–14559.
50. Sharma, C.; Kaul, M. Self-supervised few-shot learning on point clouds. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7212–7221.
51. Stojanov, S.; Thai, A.; Rehg, J.M. Using shape to categorize: Low-shot learning with an explicit shape bias. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021; pp. 1798–1808.

52. Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; Lu, J. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp. 19313–19322.

53. Feng, H.; Liu, W.; Wang, Y.; Liu, B. Enrich features for few-shot point cloud classification. In Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022; pp. 2285–2289.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.