2024

# DILF: Differentiable Rendering-Based Multi-View Image-Language Fusion for Zero-Shot 3D Shape Understanding

Xin Ning
*Chinese Academy of Sciences*

Zaiyang Yu
*Chinese Academy of Sciences*

Lusi Li
*Old Dominion University*

Weijun Li
*Chinese Academy of Sciences*

Prayag Tiwari
*Halmstad University*



## Original Publication Citation

Ning, X., Yu, Z., Li, L., Li, W., & Tiwari, P. (2024). DILF: Differentiable rendering-based multi-view image–language fusion for zero-shot 3D shape understanding. *Information Fusion*, *102*, 1-12, Article 102033. https://doi.org/10.1016/j.inffus.2023.102033

Information Fusion 102 (2024) 102033

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

# DILF: Differentiable rendering-based multi-view Image–Language Fusion for zero-shot 3D shape understanding

Xin Ning [a,1], Zaiyang Yu [a,d,1], Lusi Li [b,*], Weijun Li [a,*], Prayag Tiwari [c,*]

[a] Institute of Semiconductors, Chinese Academy of Sciences, Beijing, 100083, China
[b] Department of Computer Science, Old Dominion University, VA, 23529, United States
[c] School of Information Technology, Halmstad University, Sweden
[d] University of Chinese Academy of Science, Beijing, 100049, China

ARTICLE INFO

*Keywords:*
Zero-shot 3D shape understanding
Differentiable rendering
Text–image fusion
Information fusion

ABSTRACT

Zero-shot 3D shape understanding aims to recognize "unseen" 3D categories that are not present in training data. Recently, Contrastive Language–Image Pre-training (CLIP) has shown promising open-world performance in zero-shot 3D shape understanding tasks by information fusion among language and 3D modality. It first renders 3D objects into multiple 2D image views and then learns to understand the semantic relationships between the textual descriptions and images, enabling the model to generalize to new and unseen categories. However, existing studies in zero-shot 3D shape understanding rely on predefined rendering parameters, resulting in repetitive, redundant, and low-quality views. This limitation hinders the model's ability to fully comprehend 3D shapes and adversely impacts the text–image fusion in a shared latent space. To this end, we propose a novel approach called Differentiable rendering-based multi-view Image–Language Fusion (DILF) for zero-shot 3D shape understanding. Specifically, DILF leverages large-scale language models (LLMs) to generate textual prompts enriched with 3D semantics and designs a differentiable renderer with learnable rendering parameters to produce representative multi-view images. These rendering parameters can be iteratively updated using a text–image fusion loss, which aids in parameters' regression, allowing the model to determine the optimal viewpoint positions for each 3D object. Then a group-view mechanism is introduced to model interdependencies across views, enabling efficient information fusion to achieve a more comprehensive 3D shape understanding. Experimental results can demonstrate that DILF outperforms state-of-the-art methods for zero-shot 3D classification while maintaining competitive performance for standard 3D classification. The code is available at https://github.com/yuzaiyang123/DILP.

## 1. Introduction

Three-dimensional (3D) shape understanding is a critical task in the field of computer vision and pattern recognition. It involves analyzing the geometric structure and spatial relationships of 3D data, which can be represented as point clouds, meshes, or volumetric data (voxels). The objective of this task is to classify or categorize 3D objects based on their shapes, distinguishing them by distinct features and characteristics [1]. Applications of 3D shape understanding are diverse and widespread across various fields, such as autonomous driving, robotics, virtual reality, and environmental monitoring.

3D shape understanding encompasses both standard 3D shape understanding and zero-shot 3D shape understanding. For standard 3D understanding, there are two main strategies: the point-based strategy and the view-based strategy [2]. The point-based strategy involves directly defining a network on available raw point clouds or meshes of 3D objects, and then leveraging the defined point-based network to learn representations from the point clouds. On the other hand, the view-based strategy renders 3D objects into multiple 2D views from different perspectives and then employs a 2D-based architecture to extract semantic representations from these views [3]. While both of them can achieve impressive results on seen 3D object categories during training with available labeled 3D modality data [4], they often face challenges when it comes to unseen categories in zero-shot settings. That is because the intricate details and complexities of unseen 3D categories are hard to capture from either point clouds or multi-view images alone [5], making the models unable to generalize effectively

* Corresponding authors.
*E-mail addresses:* ningxin@semi.ac.cn (X. Ning), yuzaiyang@semi.ac.cn (Z. Yu), lusili@cs.odu.edu (L. Li), wjli@semi.ac.cn (W. Li), prayag.tiwari@ieee.org (P. Tiwari).

[1] Equal contribution.

https://doi.org/10.1016/j.inffus.2023.102033
Received 4 August 2023; Received in revised form 15 September 2023; Accepted 18 September 2023
Available online 22 September 2023

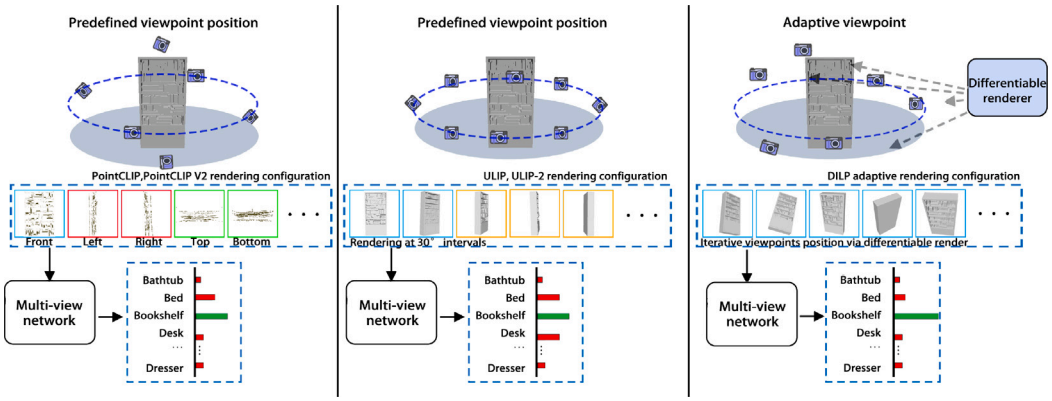


**Fig. 1.** The comparison of predefined rendering and differentiable rendering. Previous zero-shot 3D classification methods (**Left, Middle**) rely on predefined viewpoint positions for rendering multi-view images of 3D objects. However, the predefined viewpoint positions result in redundant and repetitive multi-view images (**red, green, and yellow box**). In contrast, DILF can introduce a differentiable rendering module (**Right**) that predicts optimal viewpoints in a data-driven manner. Therefore, this differentiable rendering process allows DILF to generate multi-view images that are more semantically representative.

to unseen categories. Inspired by this, incorporating knowledge from other modalities, such as language, can be a powerful and straightforward approach to overcoming the challenges in zero-shot 3D shape classification [6]. By fusing the information from multiple modalities, the model can gain a more comprehensive understanding of 3D objects and improve its ability to classify unseen categories effectively.

The language modality can provide valuable complementary information to the 3D modality [7]. By using text prompts to describe 3D scenes from a macroscopic perspective [8], the language modality can help the network achieve a more comprehensive understanding of unseen 3D categories in zero-shot learning scenarios [9]. Recent research has shown the potential of integrating language information into zero-shot learning through an "information fusion" strategy, where both the language and 3D modalities are combined to enhance the overall interpretation of the data. OpenAI's Contrastive Learning In Pretraining (CLIP) is a notable example [10], which bridges the domain gap between vision and language modalities by utilizing pre-trained text and image correlations to enable open-vocabulary recognition between the two modalities. A few extensions of CLIP have been proposed to handle the challenges in zero-shot 3D classification. For instance, PointCLIP [11] designs an inter-view adapter that utilizes text prompts to enhance the understanding of 3D multi-view images. Another method, ULIP [12] learns a unified representation of 2D RGB images, texts, and 3D point clouds by pre-training with object triplets from the three modalities. The previous methods demonstrate that the language modality can supplement extra information for unseen 3D categories, enhancing the potential of integrating the CLIP model in handling zero-shot 3D classification.

While applying CLIP to zero-shot 3D classification tasks shows promise, its success relies on the quality of rendered multi-view images since it can influence the effectiveness of the text–image fusion process. For effective zero-shot 3D classification, the model needs to understand and recognize 3D shapes from different viewpoints without direct training on these specific views. This requires having accurate and informative multi-view representations of the 3D objects. However, the exploration of a suitable rendering configuration for rendering optimal multi-view images has been lacking in previous works [6,11–13]. As shown in Fig. 1, predefined rendering configurations may not adequately capture the variations and complexities present in the 3D shapes, leading to the limited coverage of viewpoints and potentially missing important details. On the other hand, most previous CLIP-based methods treat all views equally to generate the shape descriptors, which neglects the essential role that the content relationship and discriminative information among the views play in understanding the 3D modality. Additionally, they mainly adopt CLIP 2D prompt structure, e.g., "`a photo of a [CLASS]`", and append elementary domain-specific terms, such as "`3D object`". Nevertheless, these methods confront the challenge of Naive Textual Prompting (NTP), in which the simplistic textual prompt fails to represent 3D shapes adequately and adversely affects the pre-trained language–image fusion in the embedding space [13].

To address these limitations, we propose Differentiable rendering-based multi-view Image–Language Fusion (DILF), as shown in Fig. 2. DILF consists of three modules. (1) To produce informative multi-view images, DILF employs a differentiable renderer to iteratively update the rendering parameters, enabling end-to-end training of the zero-shot 3D classification task. DILF allows the language modality to supply additional information for the 3D modality, enabling the selection of appropriate rendering viewpoints under the guidance of the language prompts, as shown in Fig. 3(a). (2) To mine the content relationship and discriminative information among the multi-view images, a group-view mechanism is introduced into DILF for enhancing the network's ability to model complex interdependencies among these views, as shown in Fig. 3(b). (3) Motivated by automatic prompt designs [13], DILF proposes the large-scale language model LLM-assisted textual feature learning which leverages the descriptive capabilities of LLMs to generate 3D-specific prompts enriched with 3D semantics to overcome the NTP challenge, as shown in Fig. 2(a). The contributions of this study can be summarized as follows:

(1) We propose a novel approach called Differentiable rendering-based multi-view Image–Language Fusion (DILF) for zero-shot 3D shape understanding. DILF combines Differentiable Rendering (DR) with language prompts to generate multi-view images with more comprehensive information. It enables the iterative optimization of viewpoint selection by fusion among 3D and language modalities. This approach integrates the principles of DR, allowing the network to refine the rendering process under the explicit guidance of language descriptions.
(2) For accurate text–image fusion, DILF introduces the group-view mechanism and LLM-assisted textual feature learning to bridge the domain gap. The group-view mechanism aims at modeling interdependencies across views, enabling a more comprehensive 3D shape understanding. The LLM-assisted textual feature learning utilizes the descriptive capabilities of LLMs, enabling a comprehensive interpretation of 3D objects. A 3D object includes its 3D and language modality, the group-view mechanism and LLM-assisted textual feature learning aim at making a complete interpretation of the 3D object at different levels and therefore facilitating accurate text–image fusion.
(3) We conduct comprehensive experiments on ModelNet40 and ScanObjectNN datasets, evaluating DILF's performance on two 3D tasks: zero-shot 3D classification and standard 3D classification. The experimental results indicate that DILF exhibits superior

performance in comparison to state-of-the-art (SOTA) for zero-shot 3D classification while maintaining competitive performance for standard 3D classification. The superior results validate the effectiveness of DILF for achieving accurate and comprehensive 3D shape understanding.

The remainder of this paper is organized as follows. Section 2 introduces related works. Section 3 presents the proposed DILF approach. Section 4 shows the related experiment results and analyses. Finally, we conclude the whole paper and direct our future research.

## 2. Related work

### 2.1. Standard 3D shape understanding

The research on 3D shape understanding can be broadly categorized into two main streams: point-based methods and view-based methods. The point-based methods involve defining the network directly on a point cloud or mesh representation of a 3D object. PointNet [14] and its variants [15–19] are examples of point-based networks used to learn representations from point clouds. These methods operate directly on the raw point data and have shown effectiveness in certain 3D tasks. The view-based methods, on the other hand, generate multiple views of a given 3D model by rendering the object from different viewpoints, resulting in multi-view images. These images are then used to solve downstream tasks such as 3D classification and segmentation. View-based methods have achieved state-of-the-art performance in these tasks [20]. Among these view-based researches, Wei et al. [2] claimed that view-based methods are effective at revealing the essential structure and contour changes of 3D objects. Su et al. [21] suggested that view-based methods align with how the human brain associates 2D appearances with prior knowledge of a 3D shape, which could contribute to their superior performance. Qi et al. [3] argued that view-based methods can complement each other's detailed features from multi-view images by setting different rendering viewpoints and thereby providing a complete interpretation of an occluded object.

The 3D multi-view images can accurately capture the characteristics of non-rigid 3D objects, providing a robust depiction of their geometry and structure [3]. Consequently, DILF employs a view-based approach to extract representations from 3D models. The difference between DILF and prior 3D shape understanding methods lies in DILF's fusion of language modality knowledge, which offers an additional complement to 3D modality. This is attributed to the fact that multimodal learning facilitates the acquisition of cross-modality information about 3D objects, enabling a comprehensive interpretation of these objects [22]. Meanwhile, a few view-based methods [2,23] have explored the position of viewpoint for rendering optimal multi-view images. However, these previous methods rely on predefined configurations for rendering 3D objects, resulting in limited success and requiring an elaborate training process [20]. The difference between DILF and these previous methods is that DILF iteratively adjusts the rendering configuration by fusing explicit text guidance into the rendering process, thereby obtaining multi-view images that are more semantically representative in a data-driven manner.

### 2.2. The extension of CLIP in zero-shot 3D shape understanding

CLIP is a type of pretraining that uses contrastive learning to learn representations from large-scale image–text pairs [24]. CLIP bridges the gap between the language and image modalities, enabling it to capture rich semantic relationships between them [25]. The efficiency of the CLIP in understanding 3D scenes has been validated through various applications, such as 3D object generation [26–28], 3D recognition [29,30], and 3D editing [31,32]. Recently, there have been several attempts to extend the application of CLIP to the task of zero-shot 3D shape understanding. Among these attempts, PointCLIP [11,13] combines visual and textual information to achieve cross-modal language–image embedding, while ULIP [6,12] employs large multimodal models to generate detailed language descriptions of 3D objects, addressing limitations in existing 3D object datasets regarding the quality and scalability of language descriptions. These methods demonstrate the effectiveness of extending the CLIP-based strategy in zero-shot 3D shape understanding.

The common point between previous CLIP-based methods and DILF lies in their fusion of language modality into zero-shot learning to enhance the understanding of 3D objects. The distinction lies in the fact that previous CLIP-based methods either render multi-view images by setting a virtual camera by 30° interval [6,12] or project the point cloud from 6 orthogonal views [11,13]. However, these rendering strategies rely on the assumption of homogeneous space (e.g., icosahedron) for predefining view configurations, resulting in leading to redundant and repetitive views [2]. In contrast to previous methods, DILF regresses suitable viewpoints with a differentiable renderer under the supervision of language modality. Consequently, this allows DILF to learn semantically representative views in a data-driven manner.

### 2.3. Differentiable Rendering (DR)

The paradigm of perception by predicting the optimal environment parameters that generated the rendering image is called Differentiable Rendering (DR) [33]. In the context of DR, appropriate viewpoint positions, lighting, and other physical properties serve as latent variables, which can be inferred to comprehend 3D scenes [34]. Recent DR approaches focus on making the graphics operations differentiable, allowing gradients to flow from the image to the rendering parameters directly [20]. Among these approaches, Abdullah et al. [20] introduce the Multi-View Transformation Network (MVTN) that regresses optimal viewpoints for 3D shape recognition, building upon advances in differentiable rendering. Tulsiani et al. [35] proposes Factored 3D Representation Learning(F3DRL) for learning disentangled 3D object representations by factorizing object shape, appearance, and viewpoint. Liu et al. [36] introduces a differentiable renderer that enables gradient-based optimization for 3D object understanding tasks. Previous research confirms that DR facilitates the examination of 3D objects from appropriate viewpoints, thereby offering a comprehensive interpretation of the 3D object.

The method most closely related to our work is MVTN [20]. In contrast to the previous work, the MVTN employs the labels of 3D shape categories to implicitly regress optimal viewpoints, whereas DILF leverages language information as explicit guidance for iteratively updating rendering parameters. Recent studies [7–9] have demonstrated the potential of fusing language modality as a supplementary component to the 3D modality. Consequently, fusing language modality in DR processing enables a more comprehensive interpretation of 3D objects.

## 3. Proposed method

Our proposed DILF approach consists of two training stages. (1) Differentiable rendering-based multi-view Image–Language Fusion (DILF) (Section 3.1), which utilizes language prompts as supervisory signals, and iteratively selects appropriate rendering parameters via a differentiable renderer. (2) DILF for 3D classification (Section 3.2), which contains two downstream tasks, namely zero-shot 3D classification and standard 3D classification.

### 3.1. Differentiable rendering-based multi-view Image–Language Fusion (DILF)

The DILF algorithm consists of three modules: (1) LLM-assisted textual feature learning (Section 3.1.1), which utilizes large-scale language models, i.e. GPT-3 [37], to generate language prompts that are rich

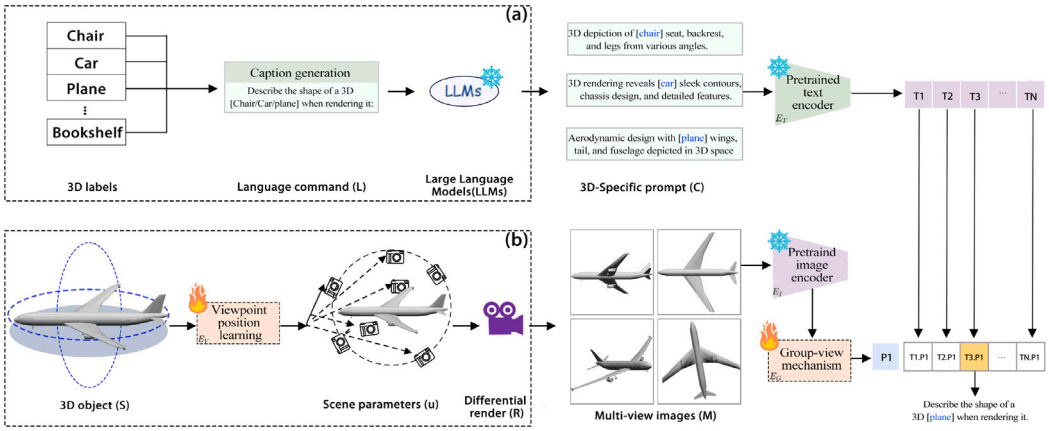


**Fig. 2.** The DILF network architecture. DILF employs a differentiable renderer that fuses explicit text guidance into the rendering process to produce informative multi-view images. DILF comprises three major components. (1) The large-scale language models (LLMs) are utilized to generate 3D-specific prompts ($C$) enriched with 3D semantics derived from the input language commands ($L$). (2) Viewpoint position learning ($E_V$) is utilized to learn the scene parameters ($u$) from the input 3D object, and then a differentiable renderer ($R$) is utilized to generate the multi-view images ($M$) based on the $u$. (3) The group-view mechanism ($E_G$) is adopted to mine the interdependencies among $M$, and to fuse $M$ into a unified feature space. The snowflake represents the freeze of the model weights, while the flame denotes iterative the model weights.

in 3D semantics. (2) Visual feature learning via differentiable renderer (Section 3.1.2), where the viewpoint position learning network is employed to estimate optimal scene parameters for rendering multi-view images. Subsequently, the group-view mechanism combines the multi-view images into a unified representation space. (3) Fusion loss (Section 3.1.3) uses language prompts as supervisory signals to select suitable rendering parameters in a data-driven manner. By pre-training on tuples from both modalities, a unified representation space for information fusion among multi-view images and language is established. The network structure of DILF is illustrated in Fig. 2.

### 3.1.1. LLM-assisted textual feature learning

In this section, we leverage the descriptive capabilities of LLMs to generate 3D-specific prompts $C_i$ enriched with 3D semantics derived from the input language commands $L_i$. Specifically, the LLMs are utilized to convert language commands $L_i$ into 3D-Specific prompts $C_i$, and then the textual features $T_i$ are extracted from 3D-Specific prompts $C_i$ via a pre-trained text encoder $E_T$.

We leverage the GPT-3 [37] to convert language commands $L_i$ into 3D-Specific prompts $C_i$. e.g., Input: "`Describe the shape of a 3D [plane] when rendering it.`"; Output: "`Aerodynamic design with plane wings, tail, and fuselage depicted in 3D space`". By doing so, the representative character of the 3D target is fully described, making the textual prompt of the 3D target more complete and comprehensive. The transformation of the language commands $L_i$ into 3D-Specific prompts $C_i$ is formulated as:

$$C_i = LLMs\left(L_i\right). \tag{1}$$

After that, DILF leverages the pre-trained vision-language model SLIP [38]($E_T$) to extract textual features $T_i$ from language commands $C_i$. The extraction of the textual features $T_i$ is formulated as:

$$T_i = E_T\left(C_i\right). \tag{2}$$

After the LLM-assisted textual feature learning, the language commands $L_i$ are transformed into textual features $T_i$, providing a feature representation of the language modality for subsequent multi-view image–language fusion.

### 3.1.2. Visual feature learning via differentiable rendering

This section demonstrates the data flow of visual feature extraction, as shown in Fig. 3. The viewpoint position learning network $E_V$ learns scene parameters $u$ from the input 3D object $S$. Subsequently, a differentiable renderer $R$ generates multi-view images $M$ based on the scene parameters $u$. The multi-view features $f_{ori}^{i}$ are then extracted from the multi-view images $M_i$ using a pre-trained image encoder $E_I$. Finally, the group-view mechanism $E_G$ is utilized to obtain the global multi-view feature $P$ by fusing the multi-view features $f_{ori}^{i}$.

***Differentiable renderer.*** The differentiable renderer $R$ allows gradients to flow from the loss of image–text alignment to the scene parameters $u$, and therefore enables obtaining suitable viewpoint position in a data-driven manner. The viewpoint position layer $E_V$ is utilized to calculate the scene parameters from the 3D object $S$:

$$u = E_V\left(S\right). \tag{3}$$

The scene parameters $u$ contain azimuth angles $u_a$ and elevation angles $u_e$ of each rendering camera. Azimuth and elevation angles are used in 3D rendering to describe the position of a rendering camera in a spherical coordinate system [39], as shown in Fig. 4. The azimuth angles $u_a$ and elevation angles $u_e$ are formulated as:

$$u_a = u_{bound} \cdot \left|\tanh\left(Chunk\left(u, 2\right)[0]\right)\right|, \tag{4}$$

$$u_e = u_{bound} \cdot \left|\tanh\left(Chunk\left(u, 2\right)[1]\right)\right|. \tag{5}$$

The $Chunk$ function is utilized to divide the $u$ into $u_a$ and $u_e$. The $u_{bound}$ is positive and it defines the permissible range for $u_a$ and $u_e$. We set $u_{bound}$ to 360°. The azimuth and elevation angles can determine the position of the rendering camera in 3D coordinates. After that, the 3D object is rendered based on the position of the rendering camera in 3D coordinates:

$$M = R\left(u_a, u_e, S\right). \tag{6}$$

During the rendering process, $R$ has two components: a rasterizer and a shader. First, the rasterizer transforms the 3D object $S$ from the world to view coordinates given the camera viewpoint and assigns faces to pixels. Using these face assignments, the shader creates multiple values for each pixel and then blends them. By doing so, the input 3D object $S$ is transformed into multi-view images $M$.

***Group-view mechanism.*** Rendering a 3D object often results in the generation of redundant and repetitive multi-view images [23]. To extract the content relationship and discriminative information from the views, we incorporate a residual connection into the group-view mechanism, as shown in Fig. 3(b). The residual structure enables the network to learn the residual mapping between the input and output of the layer, enhancing the network's ability to model complex interdependencies [40]. Consequently, incorporating the residual structure into the group-view mechanism enables a deeper exploration of the intrinsic connections among multi-view images. Specifically, we first

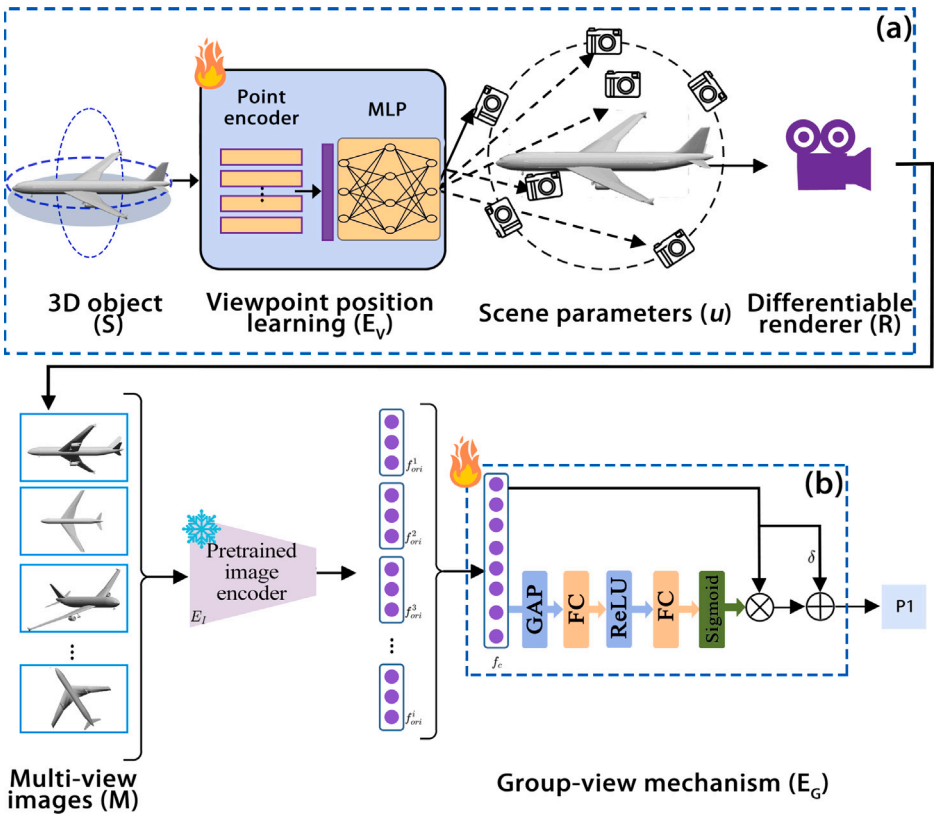


**Fig. 3.** The network structure of visual feature extraction. The viewpoint position learning($E_V$) is utilized to learn the scene parameters($u$) from 3D objects($S$), and then the differentiable renderer($R$) generates multi-view images($M$) based on the $u$. After that, the pre-trained image encoder is utilized to extract multi-view features($f_{ori}$) from $M$, and then the group-view mechanism($E_G$) is utilized to fuse the $f_{ori}$ into a unified representation space. The snowflake represents the freeze of the model weights, while the flame denotes iterative the model weights.

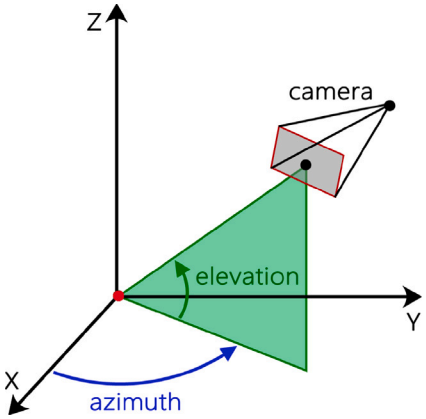


**Fig. 4.** The scene parameters of the differential render. The red point represents the 3D object. The azimuth and elevation angles can determine the position of the rendering camera in 3D coordinates.

extract the multi-view features $f_{ori}^{i}$ from multi-view images $M_i$ by the pre-trained image encoder $E_I$. Subsequently, the Multi-view features are concatenated by:

$$f_c = \text{concate}\left(f_{ori}^{1\sim i}\right), \tag{7}$$

where concate represents the vertical concatenation. After that, a residual layer is utilized, which improves the model's capacity to describe inter-dependencies between views. The global multi-view feature $P \in \mathbb{R}^{1\times N}$ are defined as:

$$P = f_c \cdot w + \delta \cdot f_c \cdot \beta, \tag{8}$$

The weight $w \in \mathbb{R}^{1\times N}$, $\delta \in \mathbb{R}^{1\times N}$ and $\beta \in \mathbb{R}^{1}$ are employed to fine-tune the weights of $f_c \in \mathbb{R}^{1\times N}$. The unlearnable hyperparameter $\beta$ serves as the balance factor for the residual connection, similar to ResNext [41]. The learnable weights $w$ and $\delta$ of the residual connection are dot multiplied with $f_c$. This process harnesses the explanatory prowess of the residual structure, thereby directing the DILF model to focus on complex inter-dependencies. The weight $w$ and $\delta$ are defined as:

$$w = Sigmod\left(W_2 ReLU\left(W_1 f_{GAP}(f_c)\right)\right), \tag{9}$$

$$\delta = Normalize\left(f_c\right), \tag{10}$$

where $W_1$ and $W_2$ are learning weights of linear layers, $f_{GAP}$ denotes the global average pooling.

After the differentiable rendering and group-view mechanism, the input 3D object $S$ is transformed into global multi-view features $P$, providing a feature representation of the 3D modality for subsequent multi-view image–language fusion.

### 3.1.3. Fusion loss in contrastive learning

Following the extraction of textual features $T$ and global multi-view features $P$, DILF pre-trained to fuse these two modalities' representations into a unified representation space. Throughout the network training, we maintain the pre-trained vision-language models, i.e., SLIP [38]($E_T$, $E_I$) in a frozen state and train the viewpoint position learning($E_V$) and group-view mechanism($E_G$) by aligning an object's textual features $T$ with its corresponding multi-view features $P$. This approach enables DILF to learn scene rendering parameters through natural language supervision, thereby obtaining suitable multi-view images in a data-driven manner. Specifically, the fusion loss in contrastive

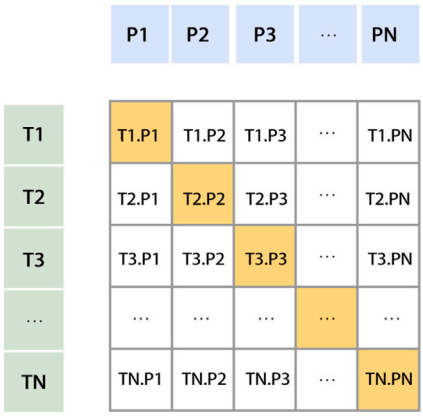


**Fig. 5.** The fusion loss in contrastive learning. The elements in the diagonal(orange region) represent the mutual match between the features $T$ and multi-view features $P$. The fusion loss aims to maximize the product of $T \cdot P$ in the diagonal.

learning is formulated as:

$$L_{(T,P)} = \sum_{(i,j)} -\frac{1}{2}\log\frac{\exp\left(\frac{T_i P_j}{\tau}\right)}{\sum_k \exp\left(\frac{T_i P_k}{\tau}\right)} - \frac{1}{2}\log\frac{\exp\left(\frac{T_i P_j}{\tau}\right)}{\sum_k \exp\left(\frac{T_k P_j}{\tau}\right)}, \quad \text{s.t. } i \neq j \tag{11}$$

where $i$ and $j$ are index of the textual features $T$ and multi-view features $P$ respectively. A positive pair in the training batch is indicated when $i = j$. Conversely, when $i \neq j$, this signifies that the textual features $T_i$ do not match the multi-view features $P_j$, as shown in Fig. 5. We use a learnable temperature parameter $\tau$ as well, similar to CLIP [10]. During pre-training, we find that if we update SLIP's [38] image and text encoders $(E_T, E_I)$ will result in catastrophic forgetting due to limited data size. This will lead to a significant performance drop when applying DILF to downstream tasks. Therefore we freeze the weights of $E_T$ and $E_I$ during the entire pre-training and only update $E_V$ and $E_G$.

In the process of multi-view image–language fusion, we utilize language as explicit guidance to obtain suitable rendering scene parameters in a data-driven manner. This approach allows DILF to benefit from the large-scale pretrained vision-language encoders, which enable the language modality to provide supplementary information for the 3D modality, and therefore augmenting zero-shot 3D shape understanding.

### 3.2. DILF for 3D classification

By leveraging the pre-trained weights of DILF as described in Section 3.1, we implement both zero-shot and standard 3D classification tasks separately based on the pre-trained weights of DILF, as shown in Fig. 6. The weight training status of DILF's different modules during pretraining, zero-shot classification, and standard classification stage, as shown in Table 1. ShapeNet [42] is employed as the pretraining dataset, while ModelNet40 [43] and ScanObjectNN [44] are used for 3D classification. To facilitate zero-shot learning, we removed overlapping classes from ShapeNet, ModelNet40, and ScanObjectNN, thereby ensuring that the unseen objects were not part of the pretraining samples. In the zero-shot classification process, we freeze the weights of DILF and input the unseen category of 3D objects into the network. Conversely, during the standard 3D classification, we freeze the weights of $E_V$ and $E_I$ and fine-tune the weights of $E_G$ and $E_C$.

## 4. Experiments

In this section, we first outline the experimental settings, encompassing the downstream datasets, implementation details, evaluation metrics, and computation costs. Subsequently, we provide the quantitative outcomes for both standard 3D classification and zero-shot 3D classification. Finally, we conduct the ablation study to verify the efficacy of each component within the DILF framework.

**Table 1**
The weight training status of DILF's different modules during pretraining, zero-shot classification, and standard classification stage. The ∇ signifies that the module weights are frozen during training. The ✓ represents the iteration of the module weights during the training process. The × represents that the module is not used in the present training stage.

| Modules | Pretraining | Zero-shot classification | Standard classification |
|---|---|---|---|
| Image encoder $(E_I)$ | ∇ | ∇ | ∇ |
| Text encoder $(E_T)$ | ∇ | ∇ | × |
| Viewpoint position learning $(E_V)$ | ✓ | ∇ | ∇ |
| Group-view mechanism $(E_G)$ | ✓ | ∇ | ✓ |
| Classification head $(E_C)$ | × | × | ✓ |
| Large-scale language models $(LLMs)$ | ∇ | ∇ | × |

### 4.1. Downstream datasets

To facilitate zero-shot learning, we employ different datasets for DILF pretraining and classification tasks. Specifically, ShapeNet [42] serves as the pretraining dataset, while ModelNet40 [43] and ScanObjectNN [44] are used for zero-shot learning. We adopts the segmentation rule of ULIP [12] for the training and evaluation dataset to ensure performing a fair comparison with previous zero-shot methods. Specifically, we eliminate overlapping classes among ShapeNet, ModelNet40, and ScanObjectNN to ensure that the unseen objects are not included in the training samples.

#### 4.1.1. Pretraining dataset

**ShapeNet** is a richly annotated, large-scale shape repository of 3D CAD models representing various objects, with extensive annotations. The 3D models in ShapeNet cover a broad spectrum of semantic categories, organized according to the WordNet taxonomy. ShapeNet is widely employed in tasks such as point cloud classification, ShapeNet comprises more than 3 million 3D models, encompassing 55 prevalent object categories and 12,000 distinct object classes.

#### 4.1.2. Classification dataset

**ModelNet40** is a synthetic dataset of 3D CAD models, with a total of 9843 samples designated for training and 2468 samples for testing, spanning across 40 distinct categories.

**ScanObjectNN** is a dataset comprising of real-world 3D scanned objects, encompassing 2902 objects classified into 15 distinct categories. The dataset is available in three variants: 'OBJ-ONLY', which includes ground truth segmented objects extracted from scene meshes datasets; 'OBJ-BJ', which features objects coupled with background noise; and 'PB-T50-RS', which introduces perturbations such as translation, rotation, and scaling to the dataset.

### 4.2. Implementation details

DILF's pretraining and fine-tuning experiments are conducted on 4 NVIDIA RTX 5000 GPUs with 16 GB of memory each. The network structure is implemented using Pytorch [45]. The differentiable renderer is implemented using Pytorch3D [46]. During the differentiable rendering, we established the rendering configurations of camera position and light position as unlearnable parameters, while the azimuth and elevation angles of the rendering camera were set as learnable parameters, in line with previous differentiable rendering-based studies [20,47,48].

During the multi-view rendering, the original point clouds in the ModelNet40 and ScanObjectNN datasets do not include color information. Consequently, colors were manually selected for the point clouds

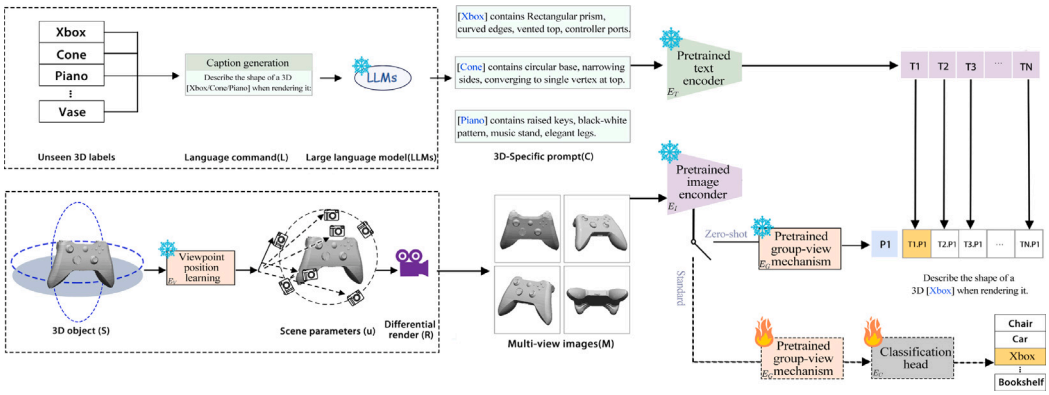


**Fig. 6.** DILF for zero-shot and standard 3D classification. The switch selects direct zero-shot classification without fine-tuning or standard classification with fine-tuning. The snowflake represents the freeze of the model weights, while the flame denotes iterative the model weights.

before their rendering into multi-view images. To ensure a fair comparison with prior 3D shape classification studies [20,21,23,49,50], the 3D objects are rendered in gray, aligning with the color configuration used in previous methods.

#### 4.2.1. Pretraining

For the network input, we uniformly sample 3072 points from each 3D object. As mentioned in Section 3.1, we choose the differentiable renderer $R$ from Pytorch3D in our pipeline for its speed and compatibility rendering. The powerful language model, GPT-3 [37], is adopted to convert the language commands into 3D-Specific prompts as described in Section 3.1.1. As mentioned in Section 3.1.2, the PointNet [14] is leveraged as the backbone of the viewpoint position learning network $E_V$. During pretraining, we utilize an advanced version of CLIP, namely SLIP [38], to perform as the visual and text encoder($E_T$, $E_I$). We freeze the weights of $E_T$ and $E_I$ and fine-tune the weights of $E_V$ and $E_G$ for 100 epochs, with the 64 as the batch size, $10^{-3}$ as the learning rate, and AdamW as the optimizer.

#### 4.2.2. Zero-shot 3D classification

In accordance with [12], the process of zero-shot 3D classification is performed by quantifying the distances between the features of a 3D object and the textual features of potential categories, as depicted in Fig. 6. It is notable that the categories utilized in zero-shot 3D classification are excluded from the pretraining phase, thereby preserving the purpose of zero-shot learning. The category that yields the minimal distance is chosen as the predicted category, as delineated in Section 3.1.3. Our pretrained models are directly utilized in zero-shot classification, without the necessity for a fine-tuning stage.

#### 4.2.3. Fine-tuning on standard 3D classification

In Standard 3D classification, we introduced the softmax layer as the classification head $E_C$. As mentioned in Section 3.2, $E_G$ and $E_C$ are fine-tuned, while the rest of the modules are frozen, as depicted in Fig. 6. We set the learning rate to $10^{-4}$ and fine-tune our model for 60 epochs, with a batch size of 64, and AdamW as the optimizer.

### 4.3. Evaluation metrics and computation cost

In accordance with 3D classification community standards, we evaluated each method using top-1 accuracy for the zero-shot 3D classification task; mean Average Precision (mAP), and Cumulative Matching Characteristic (CMC) curves for the standard 3D classification task.

We record the number of floating-point operations (GFLOPs) and the time of a forward pass for a single input sample. In the zero-shot classification evaluation stage, the floating point operations (FLOPs) of DILF when retrieving one 3D object are limited to 0.132 GFLOPs, with a per-computation cost of 13.23 ms. In the standard classification evaluation stage, the FLOPs of DILF when retrieving one 3D object are limited to 0.104 GFLOPs, with a per-computation cost of 11.62 ms. The computation cost of DILF validates that DILF is affordable when implemented in deployment scenarios. It is notable that previous zero-shot classification works [6,11–13,51] have not accounted for the computational overhead of rendering 3D objects into multi-view images in their FLOPs reporting. In contrast, DILF incorporates the differentiable renderer into its network architecture, enabling end-to-end training. Consequently, DILF calculates the complete computational overhead of model inference, inclusive of the computational overhead of differentiable rendering.

### 4.4. Result comparison

This section offers a comparative analysis between DILF and current methods within the zero-shot 3D and standard 3D classification. DILF achieves state-of-the-art performance in zero-shot 3D classification and exhibits comparable performance in standard 3D classification with the current methods.

#### 4.4.1. Zero-shot 3D classification

In Table 2, we compare the zero-shot classification performance with existing methods. DILF achieves 67.7%/53.3%/47.6%/38.5% accuracy(top-1) on ModelNet40 and ScanObjectNN, respectively, surpassing previous methods by at least +3.5%/+3.2%/+6.4%/+3.1%. The result demonstrates the superiority of DILF in handling zero-shot classification challenges, the reasons can be summarized as follows: (1) The previous zero-shot classification methods [6,11,13,51,52] utilize predefined rendering parameters to generate multi-view images. However, the predefined rendering parameters lead to redundant and repeated multi-view images when rendering a 3D object, which further leads to network overfitting [53]. With the difference from previous methods, DILF employs a differentiable renderer that fuses explicit text guidance into the rendering process to achieve iterative updates of rendering parameters, thereby guaranteeing optimal views as network input. (2) The domain gap exists in the multi-view images during 3D object rendering, as different views contain distinct attributes of the 3D object [54]. However, the previous zero-shot classification methods [6,11,13,51,52] treat all views equally to generate the shape descriptor, neglecting the significant role that the content relationship and discriminative information of the views play in understanding 3D modality. With the difference from previous methods, a group-view mechanism is adopted to mine the interdependencies among views, therefore facilitating a deeper exploration of the intrinsic connections among multi-view images. In summary, the introduction of the differentiable rendering mechanism and grouping-view mechanism augments the perceptual ability of 3D objects and, therefore enhances the performance of DILF in zero-shot 3D classification.

**Table 2**
Zero-shot 3D classification performance (top-1%) on ModelNet40 and ScanObjectNN.

| Method | Year | ModelNet40 | ScanObjectNN | | |
|---|---|---|---|---|---|
| | | | S_OBJ-ONLY | S_OBJ-BG | S_PB-T50-RS |
| PointCLIP [11] | 2022 | 23.8 | 21.3 | 19.3 | 15.4 |
| CLIP2Point [52] | 2022 | 49.4 | 35.5 | 30.5 | 23.3 |
| ULIP-2 [6] | 2023 | 61.5 | – | – | – |
| ReCon [51] | 2023 | 61.7 | 43.7 | 40.4 | 30.5 |
| PointCLIP V2 [13] | 2023 | 64.2 | 50.1 | 41.2 | 35.4 |
| DILF | 2023 | **67.7** | **53.3** | **47.6** | **38.5** |

**Table 3**
Standard 3D classification performance (Rank-1, mAP%) on ModelNet40 and ScanObjectNN. $^{\dagger}Avg$ represents the standard classification methods average performance, $^{\ddagger}Avg$ represents the zero-shot classification methods average performance.

| Method | Year | ModelNet40 | | ScanObjectNN | |
|---|---|---|---|---|---|
| | | Rank-1 | mAP | Rank-1 | mAP |
| MVTN [20] | 2021 | 93.8 | 92.2 | – | – |
| HyCoRe [55] | 2022 | 94.5 | 91.9 | 88.3 | 87.0 |
| PointNeXt [56] | 2022 | 94.0 | 91.1 | 88.2 | 86.8 |
| PointStack [57] | 2022 | 93.3 | 89.6 | 87.2 | 86.2 |
| PointMLP [58] | 2022 | 94.5 | 91.4 | 83.8 | 81.8 |
| RepSurf-U [59] | 2023 | 94.7 | – | 84.6 | – |
| Point2Vec [60] | 2023 | 94.8 | 92.0 | 87.5 | 86.0 |
| SPoTr [61] | 2023 | – | – | 88.6 | 86.8 |
| PointConT [62] | 2023 | 93.5 | **92.6** | 90.3 | 88.5 |
| PointGPT [63] | 2023 | **94.9** | – | **93.4** | – |
| $^{\dagger}Avg$ | – | 94.2 | 91.5 | 88.0 | 86.2 |
| ULIP-2 [6] | 2023 | 94.7 | 92.4 | 91.5 | 91.2 |
| ReCon [51] | 2023 | 94.7 | – | 91.2 | – |
| $^{\ddagger}Avg$ | – | 94.7 | 92.4 | 91.4 | 91.2 |
| DILF | 2023 | 94.8 | 92.4 | 91.9 | **91.6** |

#### 4.4.2. Standard 3D classification

The recent zero-shot 3D Classification methods always fall short when applied to standard 3D classification tasks. In terms of this, we validate the effectiveness of DILF on standard 3D Classification to validate its adaptability in handling standard 3D classification challenges, as shown in Table 3. Two kinds of methods are compared: methods concentrate on standard 3D classification [20,55–63] and methods concentrate on zero-shot 3D classification [6,51]. The results indicate that DILF outperforms the previous zero-shot 3D classification on standard 3D classification tasks. This is because the proposed method allows the network to obtain accurate text–image fusion in a data-driven manner, enabling a comprehensive perception of the 3D object by leveraging the generalizing power of language modality. Meanwhile, DILF shows slightly inferior to standard 3D classification methods on standard 3D classification tasks. Since the CLIP is a vision-language model, directly utilizing the CLIP in 3D shape understanding requires rendering the 3D object into multi-view images and then aligning between the text and images. However, the conversion from a 3D object into multi-view images exists in information loss, as the positional relationships between points in the 3D object are not explicitly represented in the multi-view images [64], therefore leading to a decline in network performance. Despite these limitations, DILF outperforms the average performance of standard 3D classification methods by + 0.6%/+ 0.9%/+ 3.9%/+ 5.4% in terms of Rank-1/mAP on ModelNet40 and ScanObjectNN, and the average performance of zero-shot 3D classification methods by + 0.1%/+ 0.0%/+ 0.5%/+ 0.4% in terms of Rank-1/mAP on ModelNet40 and ScanObjectNN. The above results validate the robustness of DILF on standard 3D classification tasks.

### 4.5. Ablation study

DILF comprises three components, namely LLM-assisted textual feature learning, differentiable renderer, and group-view mechanism. In this section, we perform ablation studies on the zero-shot classification task to analyze each component of DILF.

#### 4.5.1. Analysis of differentiable rendering

In this section, we conduct the ablation study for the differentiable renderer, as shown in Table 4. Specifically, we compare the effect of different rendering configurations on zero-shot classification accuracy, and then we explore the suitable viewpoints numbers in zero-shot learning.

***Analysis of alternative rendering configuration.*** To evaluate the effectiveness of the differentiable renderer in 3D shape understanding, we employ the alternative rendering configurations for producing multi-view images, while maintaining the remaining components of the DILF module. We chose PointCLIP V2 [13] rendering configuration(i.e. project the point cloud from 6 orthogonal views, index-1) and ULIP-2 [6] rendering configuration(i.e. rendering a multi-view image for every 30° interval, index-2) as alternative rendering configurations for comparison with the differentiable renderer, as presented in Fig. 1. As shown in Table 4, DILF(index-4) surpasses previous rendering configurations(index-1∼2) by at least +19.1%/+15.1%/+10.3%/+4.8% accuracy(top-1) on ModelNet40 and ScanObjectNN. The reason is that the previous zero-shot classification methods [6,13] are based on predefined rendering parameters. However, the predefined rendering parameters lead to redundant and repeated multi-view images when rendering a 3D object, which further leads to network overfitting [23]. With the difference between previous methods, DILF employs a differentiable renderer that fuses explicit text guidance into the rendering process, thereby DILF is able to leverage a data-driven approach to mining views with more discriminative information.

***Analysis of view numbers in differentiable rendering***. To evaluate the effect of the number of views on the performance of DILF. We perform an ablation study to examine the impact of varying the number of rendering viewpoints, as depicted in Table 4. The experiment results(index-3∼6) demonstrate that an excessive or insufficient number of viewpoints does not augment the DILF's capacity in 3D shape classification. The reason is that an excess of rendering viewpoints can lead to redundant and repetitive images, while a shortage might not capture the key information of 3D objects [2]. In light of the experimental findings, DILF employs 6 rendering viewpoints for multi-view image rendering (index-4).

#### 4.5.2. Analysis of LLM-assisted textual feature learning

In this section, we conduct the ablation study for LLM-assisted textual feature learning. To fully adapt the descriptive capabilities of LLMs to generate 3D-Specific prompts enriched with 3D semantics, we propose the following four series of language commands:

**Words to Sentence.** Input: "`Make a sentence using these words: a [plane], 3D shape, rendering.`"; Output: "`A 3D plane's 3D shape includes aerodynamic design with wings and tail when rendering it`".

**Key Component Description.** Input: "`What are the key components of a 3D [table]?`"; Output: "`The key components of a 3D table are the tabletop, legs or support structure, and any additional features or embellishments`".

**Discriminative Component Description.** Input: "`Compared with other 3D objects, what are the exterior features of a 3D [car]?`"; Output: "`Sleek and aerodynamic body, wheels, windows, headlights, taillights, and a distinct front grille`".

**Shape Description.** Input: "`Describe the shape of a 3D [laptop] when rendering it.`"; Output: "`Rectangular prism with a thin and flat body, a screen, and a keyboard`".

As shown in Table 5, the results indicate that the use of LLM-assisted textual feature learning(index-2∼5) enhances the top-1 accuracy of

**Table 4**
Analysis of differential render (top-1%) on ModelNet40 and ScanObjectNN.

| Index | Rendering configuration | View | ModelNet40 | ScanObjectNN | | |
|---|---|---|---|---|---|---|
| | | | | S_OBJ-ONLY | S_OBJ-BG | S_PB-T50-RS |
| 1 | PointCLIP V2 | 6 | 48.6 | 38.2 | 37.3 | 33.7 |
| 2 | UILP-2 | 12 | 47.4 | 36.5 | 35.8 | 29.3 |
| 3 | DILF | 4 | 56.5 | 44.5 | 37.3 | 32.1 |
| 4 | | **6** | **67.7** | **53.3** | **47.6** | **38.5** |
| 5 | | 8 | 54.2 | 42.4 | 38.1 | 30.1 |
| 6 | | 16 | 50.1 | 39.9 | 36.7 | 29.2 |

**Table 5**
Analysis of LLM-assisted textual feature learning (top-1%) on ModelNet40 and ScanObjectNN.

| Index | Language command | ModelNet40 | ScanObjectNN | | |
|---|---|---|---|---|---|
| | | | S_OBJ-ONLY | S_OBJ-BG | S_PB-T50-RS |
| 1 | [b]A 3D object of a [CLASS]. | 43.2 | 38.2 | 37.7 | 29.7 |
| 2 | [a]Make a sentence using these words: a [CLASS], 3D shape, rendering. | 64.2 | 50.4 | 44.1 | 36.7 |
| 3 | [a]What are the key components of a 3D [CLASS]? | 60.7 | 52.2 | 44.7 | 33.2 |
| 4 | [a]Compared with other 3D objects, what are the exterior features of a 3D [CLASS]? | 64.5 | 49.8 | 45.3 | 33.7 |
| 5 | [a]Describe the shape of a 3D [CLASS] when rendering it. | **67.7** | **53.3** | **47.6** | **38.5** |

[a] Denotes the use of LLMs to generate 3D-Specific prompts based on the input sentence
[b] Denotes the direct input of the sentence into the DILF.

**Table 6**
Performance comparison with different components (top-1%) on ModelNet40 and ScanObjectNN. The $\mathcal{D}$, $\mathcal{L}$, and $\mathcal{G}$ represent differential render, LLM-assisted textual feature learning, and group-view mechanism respectively.

| Index | $\mathcal{D}$ | $\mathcal{L}$ | $\mathcal{G}$ | ModelNet40 | ScanObjectNN | | |
|---|---|---|---|---|---|---|---|
| | | | | | S_OBJ-ONLY | S_OBJ-BG | S_PB-T50-RS |
| 1 | × | × | × | 41.1 | 31.9 | 30.7 | 25.2 |
| 2 | × | × | ✓ | 42.6 | 34.2 | 33.4 | 27.7 |
| 3 | × | ✓ | ✓ | 47.4 | 36.5 | 35.8 | 29.3 |
| 4 | ✓ | × | ✓ | 43.2 | 38.2 | 37.7 | 29.7 |
| 5 | ✓ | ✓ | × | 54.2 | 42.4 | 38.1 | 30.1 |
| 6 | ✓ | ✓ | ✓ | **67.7** | **53.3** | **47.6** | **38.5** |

DILF by at least +17.5%/+11.6%/+6.4%/+4.0% on ModelNet40 and ScanObjectNN, compared to the use of naive textual input(index-1). The reason is that the naive textual input cannot fully describe 3D shapes and harms the pre-trained language–image fusion in the embedding space [13]. To overcome the above challenge, we propose LLM-assisted textual feature learning to generate 3D-specific prompts that are rich in 3D semantics as described in Section 3.1. Based on the experimental results, DILF employs the language command structure "`Describe the shape of a 3D [CLASS] when rendering it`". and leverages LLMs to convert these language commands into 3D-Specific prompts.

#### 4.5.3. Performance comparison with different components of DILF

To evaluate each component of DILF, we perform ablation experiments including differential render($\mathcal{D}$), the LLM-assisted textual feature learning ($\mathcal{L}$), and the group-view mechanism($\mathcal{G}$), as shown in Table 6. When $\mathcal{D}$ is not adopted, we adopt the ULIP [12] rendering setting. When $\mathcal{L}$ is not adopted, We do not utilize LLMs to process input text prompts, but directly input the text prompts into DILF. When $\mathcal{G}$ is not adopted, we vertically concanate the multi-view features, and then a linear layer is adopted to resize the concentration features into specified dimensions.

***Analysis of differential render*** *($\mathcal{D}$).* For index-3 and index-6, the performance of DILF is improved by +20.3%/+16.8%/+11.8%/+9.2% accuracy(top-1) on ModelNet40 and ScanObjectNN when utilizing the $\mathcal{D}$ to generate the multi-view images. With the comparison of predefined rendering parameters(index-3), the differential render(index-6) utilizes language prompts as supervisory signals to iteratively select appropriate rendering. The experimental result demonstrates that $\mathcal{D}$ has the potential to obtain more informative rendering images and contribute to accurate text–image fusion. From index-2 and index-4, we can also observe that with the $\mathcal{D}$, the performance is improved by +0.6%/+4.0%/+3.7%/+2.0% accuracy(top-1) on ModelNet40 and ScanObjectNN, because the $\mathcal{D}$ enables a data-driven manner to obtain suitable rendering viewpoint positions.

***Analysis of LLM-assisted textual feature learning*** *($\mathcal{L}$).* For index-4 and index-6, the performance is improved by +24.5%/+15.1%/+9.9%/+8.8% accuracy(top-1) on ModelNet40 and ScanObjectNN. The reason for the decline in performance is that when directly inputting text prompts into the CLIP-based network due to the challenge of Naive Textual Prompting (NTP) [13]. NTP stems from the simplistic textual prompt's inability to adequately represent 3D shapes, which consequently impairs the pre-established language–image alignment in the embedding space. From index-2 and index-3, we can also observe that with the $\mathcal{L}$, the performance is improved by +4.8%/+2.3%/+2.4%/+1.6% accuracy(top-1) on ModelNet40 and ScanObjectNN because $\mathcal{L}$ enables to leverage the descriptive capabilities of LLMs to generate language prompts enriched with 3D semantics derived from the input textual prompts.

***Analysis of group-view mechanism*** *($\mathcal{G}$).* For index-5 and index-6, the performance is improved by +13.5%/+10.9%/+9.5%/+8.4% accuracy(top-1) on ModelNet40 and ScanObjectNN. The reason is that the $\mathcal{G}$ introduces a residual structure for feature fusion, which bolsters the network's capability to model complex interdependencies among multi-view features [40]. Therefore, incorporating the residual structure into the $\mathcal{G}$ facilitates a deeper exploration of the intrinsic connections among multi-view images. From index-1 and index-2, we can also observe that with the $\mathcal{L}$, the performance is improved by +1.5%/+2.3%/+2.7%/+2.5% accuracy(top-1) on ModelNet40 and ScanObjectNN. This enhancement is attributed to the extension of $\mathcal{G}$ into DILF, which enables DILF to extract discriminative information while preserving representative information among multi-view images.

### 4.6. Visualization result of differentiable rendering

In this section, we provide the visualization result of differentiable rendering, as shown in Fig. 7. It can be observed that as the number of epochs increases, there is progressively less redundant and repetitive information in the multi-view image. The reason is that DILF introduces a differentiable rendering module that predicts optimal viewpoints in a data-driven manner. Therefore, this differentiable rendering process allows DILF to generate multi-view images that are more semantically representative.

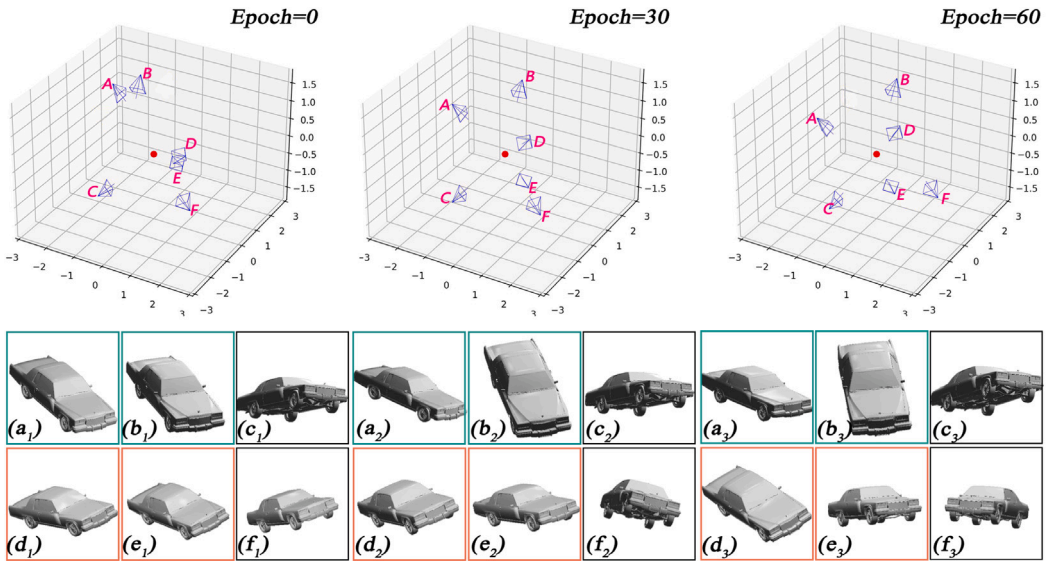


**Fig. 7.** The regression of rendering parameters in the differentiable renderer of DILF. The red point represents a 3D object, while (A-F) represent the rendering cameras, and (a-f) represent the corresponding rendering images. At epoch = 0, redundant and repetitive information can be observed in the multi-view images between $a_1$ and $b_1$, and between $d_1$ and $e_1$. DILF regresses the rendering parameters via a differentiable renderer to achieve optimal rendering viewpoint positions. Therefore, as the number of epochs increases, there is progressively less redundant and repetitive information in the multi-view image ($a_3$, $b_3$ and $d_3$, $e_3$).

## 5. Conclusion

We propose DILF, which fuses explicit text guidance into the rendering process for iteratively updating rendering parameters to produce informative multi-view images. DILF allows the network to obtain accurate text–image fusion in a data-driven manner, enabling a comprehensive perception of the 3D object by leveraging the generalizing power of language modality. Empirical results demonstrate the advantages of DILF in both standard and zero-shot 3D shape classification.

Future work may explore optimizing rendering settings. The DILF requires a predefined number of rendering viewpoints before the training process. However, the required number of viewpoints generally varies depending on the size of the 3D object. Therefore, it is crucial to dynamically acquire appropriate numbers of rendering viewpoints instead of relying on static, predefined ones. Furthermore, the proposed method can potentially tackle other zero-shot tasks, e.g., zero-shot 3D part segmentation and zero-shot 3D object detection.

## CRediT authorship contribution statement

**Xin Ning:** Data curation, Literature analysis, Interpretation of results, Preparation of the manuscript. **Zaiyang Yu:** Data curation, Literature analysis, Interpretation of results, Preparation of the manuscript. **Lusi Li:** Data curation, Literature analysis, Interpretation of results, Preparation of the manuscript. **Weijun Li:** Data curation, Literature analysis, Interpretation of results, Preparation of the manuscript. **Prayag Tiwari:** Data curation, Literature analysis, Interpretation of results, Preparation of the manuscript.

## Declaration of competing interest

The authors declare no conflict of interests

## Data availability

Github link is shared in the paper.

## Acknowledgments

We thank all reviewers and editors for their valuable feedback. This work is supported by the National Natural Science Foundation of China No. 6237334, Beijing Natural Science Foundation No. L233036.

## References

[1] Duarte Fernandes, António Silva, Rafael Névoa, Cláudia Simões, Dibet Gonzalez, Miguel Guevara, Paulo Novais, João Monteiro, Pedro Melo-Pinto, Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy, Inf. Fusion 68 (2021) 161–191.

[2] Xin Wei, Ruixuan Yu, Jian Sun, View-gcn: View-based graph convolutional network for 3d shape analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1850–1859.

[3] Shaohua Qi, Xin Ning, Guowei Yang, Liping Zhang, Peng Long, Weiwei Cai, Weijun Li, Review of multi-view 3D object recognition methods based on deep learning, Displays 69 (2021) 102053.

[4] Jiajing Chen, Burak Kakillioglu, Huantao Ren, Senem Velipasalar, Why discard if you can recycle?: A recycling max pooling module for 3D point cloud analysis, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, IEEE, 2022, pp. 549–557.

[5] Seyed Saber Mohammadi, Yiming Wang, Alessio Del Bue, Pointview-gcn: 3d shape classification with multi-view point clouds, in: 2021 IEEE International Conference on Image Processing (ICIP), IEEE, 2021, pp. 3103–3107.

[6] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, Silvio Savarese, ULIP-2: Towards scalable multimodal pre-training for 3D understanding, 2023, arXiv preprint arXiv:2305.08275.

[7] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, Hongsheng Li, Pointclip: Point cloud understanding by clip, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8552–8562.

[8] Ahmed Abdelreheem, Ujjwal Upadhyay, Ivan Skorokhodov, Rawan Al Yahya, Jun Chen, Mohamed Elhoseiny, 3DRefTransformer: fine-grained object identification in real-world scenes using natural language, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 3941–3950.

[9] Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sören Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas Guibas, Hao Zhang, Language-driven synthesis of 3D scenes from scene databases, ACM Trans. Graph. 37 (6) (2018) 1–16.

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.

[11] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, Hongsheng Li, Pointclip: Point cloud understanding by clip, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8552–8562.

[12] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, Silvio Savarese, ULIP: Learning unified representation of language, image and point cloud for 3D understanding, 2022, arXiv preprint arXiv:2212.05171.

[13] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyao Zeng, Shanghang Zhang, Peng Gao, PointCLIP V2: Adapting CLIP for powerful 3D open-world learning, 2022, arXiv preprint arXiv:2211.11682.

[14] Charles R. Qi, Hao Su, Kaichun Mo, 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, 2017, pp. 652–660.

[15] Xin Tian, Rui Liu, Zhongyuan Wang, Jiayi Ma, High quality 3D reconstruction based on fusion of polarization imaging and binocular stereo vision, Inf. Fusion 77 (2022) 19–28.

[16] Qing Ma, Junjun Jiang, Xianming Liu, Jiayi Ma, Learning a 3D-CNN and Transformer prior for hyperspectral image super-resolution, Inf. Fusion (2023) 101907.

[17] Chenru Jiang, Kaizhu Huang, Junwei Wu, Xinheng Wang, Jimin Xiao, Amir Hussain, PointGS: Bridging and fusing geometric and semantic space for 3D point cloud analysis, Inf. Fusion 91 (2023) 316–326.

[18] Saidi Guo, Xiujian Liu, Heye Zhang, Qixin Lin, Lei Xu, Changzheng Shi, Zhifan Gao, Antonella Guzzo, Giancarlo Fortino, Causal knowledge fusion for 3D cross-modality cardiac image segmentation, Inf. Fusion 99 (2023) 101864.

[19] Jiaqi Yang, Chen Zhao, Ke Xian, Angfan Zhu, Zhiguo Cao, Learning to fuse local geometric features for 3D rigid data matching, Inf. Fusion 61 (2020) 24–35.

[20] Abdullah Hamdi, Silvio Giancola, Bernard Ghanem, Mvtn: Multi-view transformation network for 3d shape recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1–11.

[21] Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 945–953.

[22] Albert Mosella-Montoro, Javier Ruiz-Hidalgo, 2D–3D geometric fusion network using multi-neighbourhood graph convolution for RGB-D indoor scene classification, Inf. Fusion 76 (2021) 46–54.

[23] Asako Kanezaki, Yasuyuki Matsushita, Yoshifumi Nishida, RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 5010–5019.

[24] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, Aditya Grover, Cyclip: Cyclic contrastive language-image pretraining, Adv. Neural Inf. Process. Syst. 35 (2022) 6704–6719.

[25] Jinmiao Fu, Shaoyuan Xu, Huidong Liu, Yang Liu, Ning Xie, Chien-Chih Wang, Jia Liu, Yi Sun, Bryan Wang, Cma-clip: Cross-modality attention clip for text-image classification, in: 2022 IEEE International Conference on Image Processing (ICIP), IEEE, 2022, pp. 2846–2850.

[26] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, Mark Chen, Point-e: A system for generating 3D point clouds from complex prompts, 2022, arXiv preprint arXiv:2212.08751.

[27] Heewoo Jun, Alex Nichol, Shap-e: Generating conditional 3d implicit functions, 2023, arXiv preprint arXiv:2305.02463.

[28] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, Seungryong Kim, Let 2d diffusion model know 3d-consistency for robust text-to-3d generation, 2023, arXiv preprint arXiv:2303.07937.

[29] Deepti Hegde, Jeya Maria Jose Valanarasu, Vishal M. Patel, Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition, 2023, arXiv preprint arXiv:2303.11313.

[30] Manh Tung Tran, Minh Quan Vu, Ngoc Duong Hoang, Khac-Hoai Nam Bui, An effective temporal localization method with multi-view 3D action recognition for untrimmed naturalistic driving videos, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3168–3173.

[31] Junha Hyung, Sungwon Hwang, Daejin Kim, Hyunji Lee, Jaegul Choo, Local 3D editing via 3D distillation of CLIP knowledge, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 12674–12684.

[32] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, Jing Liao, Clip-nerf: Text-and-image driven manipulation of neural radiance fields, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3835–3844.

[33] Sai Praveen Bangaru, Michaël Gharbi, Fujun Luan, Tzu-Mao Li, Kalyan Sunkavalli, Milos Hasan, Sai Bi, Zexiang Xu, Gilbert Bernstein, Fredo Durand, Differentiable rendering of neural sdfs through reparameterization, in: SIGGRAPH Asia 2022 Conference Papers, 2022, pp. 1–9.

[34] Lukasz Romaszko, Christopher K.I. Williams, Pol Moreno, Pushmeet Kohli, Vision-as-inverse-graphics: Obtaining a rich 3d explanation of a scene from a single image, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 851–859.

[35] Shubham Tulsiani, Saurabh Gupta, David F. Fouhey, Alexei A. Efros, Jitendra Malik, Factoring shape, pose, and layout from the 2d image of a 3d scene, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 302–310.

[36] Shichen Liu, Weikai Chen, Tianye Li, Hao Li, Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction, 2019, arXiv preprint arXiv:1901.05567.

[37] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[38] Norman Mu, Alexander Kirillov, David Wagner, Saining Xie, Slip: Self-supervision meets language-image pre-training, in: European Conference on Computer Vision, Springer, 2022, pp. 529–544.

[39] Thu H. Nguyen-Phuoc, Chuan Li, Stephen Balaban, Yongliang Yang, Rendernet: A deep convolutional network for differentiable rendering from 3d shapes, Adv. Neural Inf. Process. Syst. 31 (2018).

[40] Jie Hu, Li Shen, Gang Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[41] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500.

[42] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, Fisher Yu, ShapeNet: An information-rich 3D model repository, 2015, CoRR, abs/1512.03012.

[43] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, Jianxiong Xiao, 3D ShapeNets: A deep representation for volumetric shapes, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, IEEE Computer Society, 2015, pp. 1912–1920.

[44] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, Sai-Kit Yeung, Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1588–1597.

[45] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito, Zeming Lin, Alban Desmaison, Luca Antiga, Adam Lerer, Automatic differentiation in PyTorch, 2017.

[46] Nikhila Ravi, Jeremy Reizenstein, David Novotný, Taylor Gordon, Wan-Yen Lo, Justin Johnson, Georgia Gkioxari, Accelerating 3D deep learning with PyTorch3D, 2020, CoRR, abs/2007.08501.

[47] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, Bernard Ghanem, Pointnext: Revisiting pointnet++ with improved training and scaling strategies, Adv. Neural Inf. Process. Syst. 35 (2022) 23192–23204.

[48] Abdullah Hamdi, Bernard Ghanem, Matthias Nießner, SPARF: Large-scale learning of 3D sparse radiance fields from few input images, 2022, arXiv preprint arXiv:2212.09100.

[49] An-An Liu, Heyu Zhou, Weizhi Nie, Zhenguang Liu, Wu Liu, Hongtao Xie, Zhendong Mao, Xuanya Li, Dan Song, Hierarchical multi-view context modelling for 3D object classification and retrieval, Inform. Sci. 547 (2021) 984–995.

[50] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, Yue Gao, Gvcnn: Group-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 264–272.

[51] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, Li Yi, Contrast with reconstruct: Contrastive 3D representation learning guided by generative pretraining, 2023, arXiv preprint arXiv:2302.02318.

[52] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson W.H. Lau, Wanli Ouyang, Wangmeng Zuo, Clip2point: Transfer clip to point cloud classification with image-depth pre-training, 2022, arXiv preprint arXiv:2210.01055.

[53] Felix Petersen, Bastian Goldluecke, Christian Borgelt, Oliver Deussen, Gendr: A generalized differentiable renderer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4002–4011.

[54] Shuo Wang, Xinhai Zhao, Hai-Ming Xu, Zehui Chen, Dameng Yu, Jiahao Chang, Zhen Yang, Feng Zhao, Towards domain generalization for multi-view 3D object detection in bird-eye-view, 2023, CoRR, abs/2303.01686.

[55] Antonio Montanaro, Diego Valsesia, Enrico Magli, Rethinking the compositionality of point clouds through regularization in the hyperbolic space, in: NeurIPS, 2022.

[56] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, Bernard Ghanem, PointNeXt: Revisiting PointNet++ with improved training and scaling strategies, in: NeurIPS, 2022.

[57] Kevin Tirta Wijaya, Dong-Hee Paek, Seung-Hyun Kong, Advanced feature learning on point clouds using multi-resolution features and learnable pooling, 2022, CoRR, abs/2205.09962.

[58] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, Yun Fu, Rethinking network design and local geometry in point cloud: A simple residual MLP framework, 2022, arXiv preprint arXiv:2202.07123.

[59] Haoxi Ran, Jun Liu, Chengjie Wang, Surface representation for point clouds, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, IEEE, 2022, pp. 18920–18930.

[60] Karim Abou Zeid, Jonas Schult, Alexander Hermans, Bastian Leibe, Point2Vec for self-supervised representation learning on point clouds, 2023, CoRR, abs/2303.16570.

[61] Jinyoung Park, Sanghyeok Lee, Sihyeon Kim, Yunyang Xiong, Hyunwoo J. Kim, Self-positioning point-based transformer for point cloud understanding, 2023, CoRR, abs/2303.16450.

[62] Yahui Liu, Bin Tian, Yisheng Lv, Lingxi Li, Feiyue Wang, Point cloud classification using content-based transformer via clustering in feature space, 2023, CoRR, abs/2303.04599.

[63] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, Yufeng Yue, PointGPT: Auto-regressively generative pre-training from point clouds, 2023, arXiv preprint arXiv:2305.11487.

[64] Qijian Zhang, Junhui Hou, Yue Qian, PointMCD: Boosting deep point cloud encoders via multi-view cross-modal distillation for 3D shape recognition, IEEE Trans. Multimed. (2023).