Attention4Align: Align Multi-View Parts Via Part2Part Hierarchical Attention Map for Fine-Grained 3D Object Classification

Runchu Zhang, Jiahe Yue, Zhe Zhang, and Jie Ma(✉)

Huazhong University of Science and Technology, Luoyu Road 1037, Wuhan, China
{runchuzhang,yuejiahe2000,zhangzhe1997,majie}@hust.edu.cn

Abstract. Multi-view methods offer an effective approach to address 3D object classification, and there is a growing interest in handling more practical scenarios, such as fine-grained distinctions and arbitrary viewpoints. Fine-grained object distinctions often appearance around local parts, inspiring various part-based methods. However, the parts generated by these methods are typically unordered, significantly impacting the aggregation operation across different views and consequently diminishing overall performance. To address this issue, we propose a Part2Part Hierarchical Attention, facilitating information exchange and feature enhancement among parts in different viewpoints through attention within and across views. Subsequently, the Part2Part Attention Map generated during the attention process is utilized to measure the distance between multi-view parts, aiding in their alignment. We also employ multi-scale feature fusion to enhance the quality of parts generated by weakly supervised learning. Experimental results indicate that, under the same settings, our approach achieves state-of-the-art performance on the FG3D and MVP-N datasets.

Keywords: Multi-View · Fine-grained shape classification · 3D Objects · Parts Alignment

1 Introduction

Humans live in a world composed of 3D objects, and understanding these 3D objects is crucial for our perception of the world. In reality, human perception of objects is largely derived from 2D observations. However, a single 2D view often fails to comprehensively convey the information of a 3D object. Therefore, combining information from multiple views is an effective approach to address this limitation.

Currently, methods based on multi-views [3,5,13,21,29,36,37] have made significant contributions in the fields of 3D object recognition, retrieval, generation, *etc*. Compared to methods based on voxels [16,20,23,26,41] and point clouds [14, 22, 24, 28], multi-view approaches can leverage existing works [2, 10, 27, 31–33] in the 2D image processing domain for feature extraction and pre-training. As a

result, they can achieve state-of-the-art performance with lower computational costs and faster processing speeds.

However, for the majority of current datasets [15, 34, 41], the appearance variations of 3D objects are substantial, and a single view may already encompass sufficient discriminative information. In such cases, the improvement from multi-view methods might not be substantial. Therefore, some studies [19, 35], have begun to explore a more practically valuable topic — Fine-grained Classification of 3D objects based on Multi-view methods.

Fine-grained image classification [39] poses a challenging task in computer vision. Subtle differences between images often concentrate on specific parts/regions, leading to the development of a series of part-based fine-grained classification methods [1, 4, 12, 18, 25, 30, 42, 44], which have achieved state-of-the-art performance. In these methods, it is common to generate multiple distinct parts for an object and then concatenate the features of these parts to form the final descriptor for fine-grained classification.

For 3D objects requiring fine-grained classification, certain crucial and discriminative parts/regions may be unobservable from certain viewpoints. In such cases, multi-view-based classification methods become more competitive compared to approaches relying solely on a single viewpoint. Therefore, there is a need to design a robust, accurate, and efficient multi-view 3D object fine-grained classification model.

Designing an effective part-based multi-view 3D fine-grained classification framework poses several challenges. First, it involves generating high-quality part-level features without relying on additional annotations. Second, there is the challenge of aggregating part-level features from multiple views robustly to form a comprehensive representation of the 3D object. Finally, the framework should perform well under both fixed and arbitrary viewpoint settings to enhance its practical utility.

To address these challenges, we propose a weakly supervised end-to-end 3D fine-grained classification network. Specifically, for a sequence of multi-view images of a 3D object, we first utilize a Weakly Supervised Part-features Generation module to generate multiple part features for each view. Subsequently, we employ a Part2Part Hierarchical Attention Encoder to enhance the feature of all parts in different views. Then, we leverage the Part2Part Similarity Map obtained during the attention process to align the parts across different views. Finally, to preserve information from different parts within the same viewpoint and aggregate similar parts across different viewpoints, features of aligned parts within different views are first transformed into 3D part-features through max-pooling. These 3D part-features are then concatenated to form the 3D descriptor of the object, which is subsequently fed into a classifier for classification.

Our contributions include:

1.Proposing a Weakly Supervised, Part-based, End-to-End 3D multi-view fine-grained classification model and enhancing the quality of generated parts through multi-scale fusion operations.

2.Designing a Part-to-Part Hierarchical Attention Encoder, facilitating information flow and feature enhancement among parts from different views through Part2Part Attention Within&Across View.

3.Introducing a Multi-View Part-Features Alignment module in conjunction with the Attention module above, allowing for the aggregation of semantically similar parts from different views to form the 3D part-features of an object, thereby improving the effectiveness of the multi-view aggregation.

4.Achieving state-of-the-art performance on two existing 3D fine-grained classification datasets, FG3D [19] and MVP-N [35].

2 Related Works

2.1 Multi-view based method for 3D Object Classification

For Multi-view-based 3D Object Classification Method, how to effectively aggregate information from different views is especially important.

MVCNN [29] aggregates multi-view information via max-pooling on feature vectors. RotationNet [13] jointly predicts object category and pose by treating the view as an optimizable latent variable. GVCNN [3] uses a Grouping Module to cluster views, applies pooling to form Group Descriptors, and fuses them to create a 3D shape descriptor. LR [43] enhances regions across views using a Region Reinforcing block and integrates them via a Region Integration block. View-GCN/View-GCN++ [37, 38] introduces Graph Convolutional Networks, encoding view positions and enabling inter-view information propagation.

Most of the studies mentioned above focus on Fixed viewpoints and Coarse-grained datasets. In fixed viewpoints, cameras capture 3D objects from predefined angles, with viewpoint information either explicitly (e.g., View-GCN [37]) or implicitly (e.g., RotationNet [13]) integrated into the model. Coarse-grained datasets have large inter-category differences, enabling accurate classification from single views and allowing simpler methods to perform well. 1 illustrates the differences between these settings.

As research progresses, an increasing number of scholars are directing their attention to the study of multi-view methods in more practical scenarios, such as arbitrary viewpoint settings and fine-grained classification.

CVR [36] is a dedicated study on multi-view classification under arbitrary viewpoint settings. It achieves state-of-the-art performance in this context by transforming the view features from each arbitrary viewpoint to Canonical View Representation through optimal transport before aggregation.

FG3D [19] released the first 3D multi-view fine-grained dataset, which includes three coarse-grained categories: Airplane, Car, Chair, and 33 fine-grained categories. Simultaneously, the paper proposed a multi-view network — FG3D-Net, designed for 3D fine-grained classification. MVP-N [35] introduced the first real-world 3D multi-view fine-grained dataset, encompassing 19 groups with a total of 44 challenging-to-differentiate retail commodity categories. It also evaluated the performance of some coarse-grained methods on this dataset.

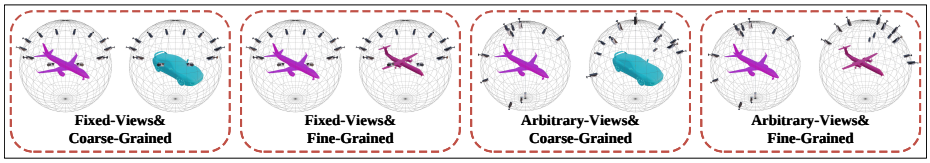

Fig. 1. Comparison of Multi-View 3D object classification tasks in different Settings. Unlike the Fixed-Views setting, the Airbitrary-Views setting does not require prior knowledge of the number of cameras or their positions relative to the object. Additionally, Fine-Grained classification represents a more detailed categorization compared to the usual Coarse-Grained classification.

Attention mechanisms also play a significant role in aggregating information from multiple views. CVR [36] combines a multi-head Transformer Encoder with ResNet [10] as the backbone network for view feature extraction, and another Transformer Encoder is used for view aggregation. DAN [21] explores the aggregation capability of the Deep-Attention network for view information. In FG3D-Net [19], twice attention operations are employed for Part-level and View-level aggregation.

2.2 Part-based method for Fine-Grained Image Classification

Parts/Regions with discriminative information are crucial for distinguishing fine-grained categories, leading to the development of a series of methods. While how to generate high-quality components only by category labels is a hot research topic.

RA-CNN [4] was the first to use attention mechanisms to enhance the accuracy of CNNs in fine-grained recognition. It employed a Recurrent Visual Attention model to iteratively generate region attention maps, progressively focusing on different parts of the image from coarse to fine. MA-CNN [44] proposed a Multi-Attention CNN to generate multiple region attentions and concatenate the features from these regions for final classification. Subsequent methods largely followed the practice of concatenating multiple parts of an object to form the final descriptor. MAMC [30] draws inspiration from the concept of channel attention [11], directly generating multiple part-level features on the feature map without cropping the original image. This approach enables end-to-end training of the network.

WS-DAN [12] also generates part-level features directly on the feature map after downsampling. It divides the process into two steps: firstly, using a simple convolutional network to perform channel attention and generate multiple attention mask maps, and then combining each attention mask map with the feature map through Bilinear Attention [17] Pooling to generate part-level features. CAL [25] introduces a Counterfactual Attention learning method based on causal inference to learn more effective attention and enhance the quality of attention map generation. P2P-Net [42] considers part-level features as a form of regularization for global features. To improve the regularization effect, it also

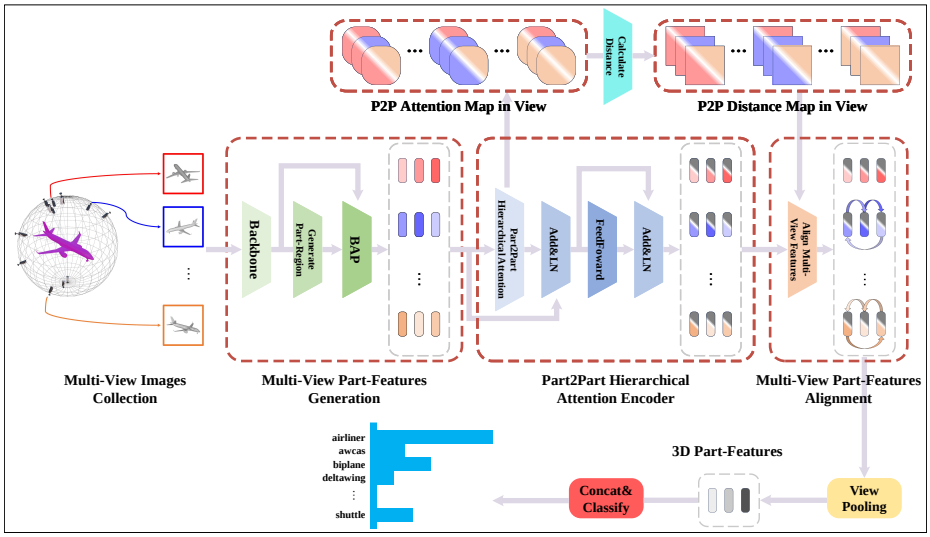

Fig. 2. Our method's overall framework. It consists primarily of three modules: Multi-View Part-Features Generation, Part-to-Part Hierarchical Attention Encoder, and Multi-View Part-Features Alignment. Detailed information can be found in 3.

introduces a Self-Supervised Pose Alignment module to reorganize multiple distinct parts.

3 Methodology

In this section, we will provide a detailed introduction to the architecture of our Part-based Multi-View 3D object classification model. As illustrated in 2, the primary component of our model comprises three modules:

Multi-View Part-Features Generation. This module is responsible for generating distinct part-features for each view.

Part2Part Hierarchical Attention Encoder. It enables similar parts from different views to exchange messages and mutually enhance features.

Multi-View Part-Features Alignment. This module aligns semantically similar parts from different views to facilitate subsequent aggregation.

3.1 Multi-View Part-Features Generation

After image capture, we can obtain a set of images $\{I_i\}_{i=1}^{V}$ from different views. Our goal is to predict M part-level attention regions for each view's image and form $V \times M$ Part-features $\left\{p_i^j\right\}$. Note that our approach is weakly supervised, which means we only use objects' category labels to predict attention regions.

For a generic backbone like ResNet [10], it typically forms a multi-dimensional small-sized feature map by downsampling the image multiple times. Then, a

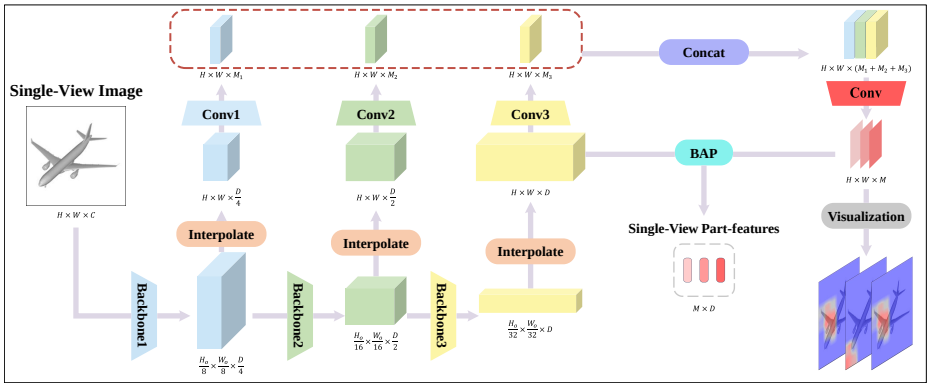

Fig. 3. The overall process of Multi-View Part-Features Generation. Our method performs multi-scale feature extraction, concatenation, and BAP operations on single-view images, ultimately obtaining multiple part-level features from a single view.

global average pooling (GAP) operation is applied to generate image-level features. WS-DAN [12] generates multiple local attention maps by applying a convolutional function on the final feature map. Then, through masking the attention maps with the final feature map sequentially and employing global pooling, different part-level features are obtained (this operation is referred to as BAP: Bilinear-Attention-Pooling in the paper). We draw inspiration from this approach and fuse features maps from multiple scales during the downsampling process to generate local attention maps.

As shown in 3, taking ResNet18 [10] as an example, after passing through three segments of the backbone, a single-view image $I_i \in R^{H_o \times W_o \times 3}$ generates three feature maps of different scales $F_i^1 \in R^{\frac{H_o}{8} \times \frac{W_o}{8} \times \frac{D}{4}}$, $F_i^2 \in R^{\frac{H_o}{16} \times \frac{W_o}{16} \times \frac{D}{2}}$, $F_i^3 \in R^{\frac{H_o}{32} \times \frac{W_o}{32} \times D}$. Firstly, we align the feature maps F_i^s to the same scale through interpolation in 1.

$$F_i^{s'} = f_{interpolate}\left(F_i^s\right) \in R^{H \times W \times \frac{D}{2^{3-s}}}, \forall s \in \{1,2,3\} \tag{1}$$

Furthermore, the aligned feature maps $F_i^{s'}$ are separately encoded in 2 and then concatenated together in 3, where the encoding function involves a convolution operation.

$$F_i^{s''} = f_{conv(s)}\left(F_i^{s'}\right) \in R^{H \times W \times M_s}, \forall s \in \{1,2,3\} \tag{2}$$

$$F_i = Concat\left(\left[F_i^{1''}, F_i^{2''}, F_i^{3''}\right]\right). \quad F_i \in R^{H \times W \times (M_1+M_2+M_3)} \tag{3}$$

Subsequently, the concatenated multi-scale feature maps F_i undergo another round of convolution and ReLU activation in 4, forming M local attention maps A_i.

$$A_i = ReLU\left(f_{conv}\left(F_i\right)\right). \quad A_i \in R^{H \times W \times M} \tag{4}$$

Finally, Multi-view Part-feature are generated using the BAP [12] in 5, where $A_i^j \in R^{H \times W}$, p_i^j represents the j-th part feature of the i-th view and $(\cdot)$ represents the multiplication operation between matrices.

$$p_i^j = GAP\left(F_i^{3'} \cdot A_i^j\right). \quad p_i^j \in R^D \tag{5}$$

Applying the same operation to V views, we will obtain V part sets $P_i = \left[p_i^1, \cdots, p_i^M\right] \in R^{M \times D}$ and a total set $P = [P_1, P_2, \cdots, P_V] \in R^{V \times M \times D}$.

Compared to WS-DAN [12], our approach generates local attention maps by combining feature maps from multiple scales, rather than solely relying on the deepest one. Fusion of feature maps from lower layers enhances the quality of generated part-level features due to the richer texture details present in lower-level feature maps.

3.2 Part-to-Part Hierarchical Attention Encoder

After the Multi-View Part-Features Generation module in 3.1, we obtain $V \times M$ part-level features for V views. To enhance the representational power of each part-feature and promote robustness, we aim for each part to gather information from similar parts in other views for feature enhancement. Due to the hierarchical organization of parts (different parts in different views), we have designed a Part-to-Part(P2P) Hierarchical Attention Encoder for inter-view transmission of information and feature enhancement among different parts.

Our P2P Hierarchical Attention Encoder draws inspiration from the structure of Transformer encoder. As shown in 2, it mainly consists of the P2P Hierarchical Attention module, Feed Forward module, and some layer normalization operations. We also employ residual connections to enhance the stability of the model. Among these components, the P2P Hierarchical Attention module serves as the core of our encoder.

Our P2P Hierarchical Attention is primarily composed of three operations: first, the formation of a similarity map between each pair of parts through the P2P Scaled Dot-Product; then, different calculations on the similarity map yield attention maps for Part-to-Part and Part-to-View. Finally, utilizing these two maps to implement Attention Within & Across View for each part. We detail these processes in 4.

For Raw Multi-view Part-features P, we first obtain the Query, Key and Value Part-features through an embedding operation in 6. For simplicity of calculation and clarity of the formula, we flattened the first two dimensions of P as $P \in R^{(V \times M) \times D}$.

$$Q = PW^Q, K = PW^K, V = PW^K. \quad Q, K, V \in R^{(V \times M) \times D} \tag{6}$$

Next, we use the Query part-features and Key part-features to compute the Part2Part Similarity Map (SM_{P2P}) in 7. Here, $SM_{P2P}(i, j, k, l)$ represents the similarity between p_i^j and p_k^l.

$$SM_{P2P} = \frac{QK^T}{\sqrt{D}}. \quad SM_{P2P} \in R^{(V \times M) \times (V \times M)} \tag{7}$$

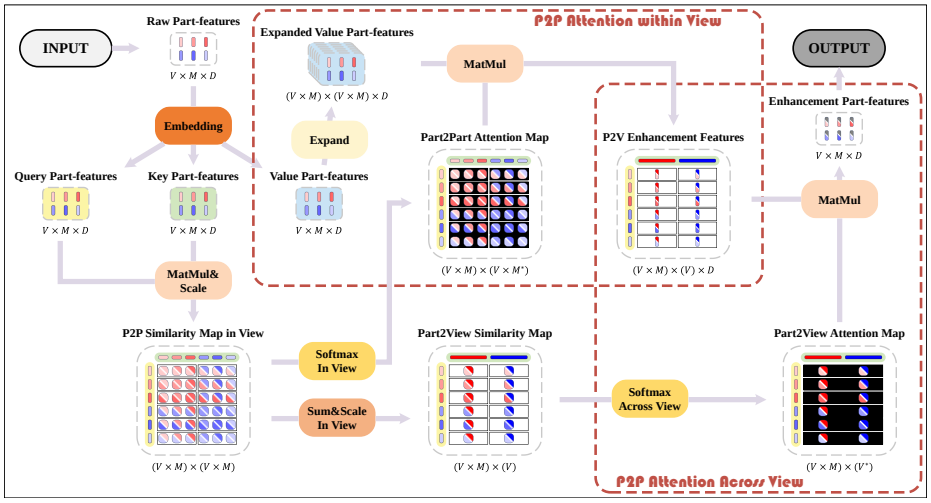

Fig. 4. The main process of Part 2Part Hierarchical Attention includes two levels of attention operations: P2P attention Within & Across view. For ease of understanding, different colors of Part-features signify their origin from distinct views, while the intensity of color indicating varying semantic content. The black box indicates that the contents within it are in the same view and * indicates that the current dimension has undergone a softmax operation.

Subsequently, a softmax operation is applied along the last dimension of SM_{P2P} to obtain the P2P Attention Map in View(AM_{P2P}) in 8. While in View

$$AM_{P2P} = Softmax\left(SM, dim = -1\right). \quad AM_{P2P} \in R^{(V\times M)\times(V\times M)} \tag{8}$$

Here, $AM_{P2P}\left(i,j,k,\bullet\right) \in R^{M}$ denotes the attention weights list of the part p_i^j to all M parts P_k in the k-th view, which represents the meaning of "in view". A larger value for $AM_{P2P}\left(i,j,k,\bullet\right)$ indicates that, part P_k^l is semantically closer to P_i^j, conveying more information to P_i^j, and contributing more to the feature enhancement of P_i^j.

Utilizing the Part2Part Attention Map in View, we can perform the Part-to-Part Attention Within the view in 9, where $V\left(k\right) \in R^{M\times D}$ and $(\cdot)$ represents the multiplication operation between vectors and matrices.

$$PV(i,j,k) = AM_{P2P}\left(i,j,k,\bullet\right) \cdot V\left(k\right). \quad PV(i,j,k) = pv_{i,j}^{k} \in R^{D} \tag{9}$$

$PV\left(i,j,k\right)$ is the enhancement vector from the k-th view to the part p_i^j. It is a linear combination of the M Value Part-Features from the V-th view. We refer to PV as the Part2View Enhancement Feature.

Since each part has V enhancement features, we need to perform another attention operation to aggregate them into the overall enhancement feature. As we cannot directly compute the similarity between parts and views, we approximate it in 10 by summing the similarities of part p_i^j to all parts P_k in a view

and then scaling.

$$SM_{P2V}(i,j,k) = \frac{\sum_{l=1}^{M}(SM_{P2P}(i,j,k,l))}{\sqrt{M}}. \quad SM_{P2V} \in R^{(V\times M)\times V} \tag{10}$$

We also apply softmax along the last dimension of SM_{P2V} in 11 to obtain the Part2View Attention Map(AM_{P2V}).

$$AM_{P2V} = Softmax(SM_{P2V}, dim = -1). \quad AM_{P2V} \in R^{(V\times M)\times V} \tag{11}$$

Finally, utilizing AM_{P2V}, in 12, we perform the Part-to-Part Attention Across views. Here, $AM_{P2V}(i,j,\bullet) \in R^{M\times M}$, $PV(i,j,\bullet) \in R^{M\times D}$, and $(\cdot)$ represents the multiplication operation between matrices.

$$E(i,j) = AM_{P2V}(i,j,\bullet) \cdot PV(i,j,\bullet). \quad E(i,j) = e_i^j \in R^D \tag{12}$$

$E \in R^{V\times M\times D}$ is the final Enhancement Part-feature, e_i^j representing the information p_i^j gathers from corresponding parts in other views. Through the Hierarchical Attention operation above, we efficiently achieved interaction between different parts in different views.

As 2 shows, after the Part2Part Hierarchical Attention operation, we add the Raw Part-features with the Enhancement Part-features using a residual structure and apply layer normalization(LN). Then, we use a Feed Forward Network (FFN) for encoding and reapply Add&LN to obtain the Encoded Part-Features $P_E \in R^{V\times M\times D}$ which consists of some $p_{E_i}^j \in R^D$. Here, FFN is a two-layer fully connected network with hidden layer nodes equal to $4 \times D$ and output layer nodes equal to D.

3.3 Multi-view Part-Features Alignment

For the Multi-view Parts-features generated by 3.1, we cannot guarantee their semantic alignment. This implies that the semantic similarity between $p_{E_i}^j$ and $p_{E_k}^j$ may be smaller than the similarity between $p_{E_i}^j$ and $p_{E_k}^l$. However, during aggregation, $p_{E_i}^j$ is pooled with $p_{E_k}^j$ instead of $p_{E_k}^l$. This will lead to semantic confusion in the aggregated features, thereby reducing performance.

In 3.2, the Encoder module enhances the representation of Part-features but does not alleviate semantic misalignment. This is because attention operations are position-independent. So we designed a non-learning Multi-view Part-Features Alignment module in this section to align different parts from different views. Simply put, we allocate each of the M Encoded Parts in each view to M groups to achieve alignment.

In the P2P Hierarchical Attention module, we obtain an attention map AM_{P2P} measuring the attention between parts. By transforming AM_{P2P} in 13, we derive a P2P distance map DM_{P2P} to approximate the distance of part $p_{E_i}^j$ to M parts in each view.

$$DM_{P2P} = -\ln(AM_{P2P}). \quad DM_{P2P} \in [0, +\infty] \tag{13}$$

Clearly, $AM_{P2P} \in [0,1]$. And when $AM_{P2P}(i,j,k,l)$ is larger than others in $AM_{P2P}(i,j,k,\bullet)$, it indicates more information flow between $p_{E_i}^{\ j}$ and $p_{E_k}^{\ l}$ than other parts in k-th view, suggesting closer semantic similarity between them.

Next, for a total set of encoded parts $P_E = [P_{E1}, P_{E2}, \cdots, P_{EV}]$ that need alignment, we assign the parts $\left[p_{E_i}^{\ 1}, p_{E_i}^{\ 2}, \cdots, p_{E_i}^{\ M}\right]$ to M groups $[G_1, \cdots, G_M]$ through DM_{P2P} from P_{E1} to P_{EV}.

For ease of representation, we define an allocation matrix $X_i \in \{0,1\}^{M \times M}$ as a binary matrix for the i-th view. If $X_i(j,t) = 1$, it indicates that the j-th encoded part in the i-th view is assigned to the t-th group. At the same time, we have $\sum_j X_i(j,t) = 1$, $\sum_t X_i(j,t) = 1$.

When we align the encoded parts set P_{E_i}, the previous $i-1$ views have already been aligned so each group already has $i-1$ encoded parts. As shown in 14, we approximate the distance from $p_{E_i}^{\ j}$ to each group G_t by summing the distances from $p_{E_i}^{\ j}$ to the all $i-1$ allocated encoded parts in group G_t.

$$\begin{cases} DM_{P2G}^{i}(j,t) = \sum_{k=1}^{i-1} \left(DM_{P2P}\left(i,j,k,l_k^t\right)\right) \\ s.t. \quad X_k\left(l_k^t, t\right) = 1. \quad \forall k \in [1, i-1], t \in [1, M] \end{cases} \tag{14}$$

In this context, l_k^t represents the index of the encoded part allocated to group G_t in the k-th already allocated view.

To ensure semantic similarity among parts within the same group, we aim for the minimal distance generated by the every allocation. So the allocation problem of the i-th view can be modeled as 15.

$$min\, z_i = \sum_{j=1}^{M}\sum_{t=1}^{M} D_{P2G}^{i}(j,t)\, X_i(j,t). \quad s.t. \begin{cases} \sum_j X_i(j,t) = 1 \\ \sum_t X_i(j,t) = 1 \\ X_i(j,t) \in \{0,1\} \end{cases} \tag{15}$$

This is a typical assignment problem that can be solved using the Hungarian algorithm [40].

By solving the optimization problem multiple times, we obtained the assignment matrix X_i for each view. Alignment operation only needs to be performed $V-1$ times as we do not adjust the order of $\left[p_{E_1}^{\ 1}, p_{E_1}^{\ 2}, \cdots, p_{E_1}^{\ M}\right]$ and set them as the initial points for the M groups.

After alignment, we perform max-pooling within each group to generate M 3D Part-features. Finally, these 3D Part-features are concatenated to form a robust 3D feature descriptor and fed into a single-layer MLP network for classification.

4 Experiment

4.1 Datasets

We experimented with our method on two existing multi-view fine-grained datasets, FG3D and MVP-N.

FG3D [19] dataset contains three sub-datasets: air, car, and chair, with 13, 20, and 33 fine-grained categories, respectively. We generated multi-view data under two settings: fixed and arbitrary viewpoints. In the fixed setting, 12 virtual cameras are placed 30 degrees apart and elevated 30 degrees, capturing 12 views. In the arbitrary setting, 4-12 random viewpoints on a spherical surface were chosen, with cameras directed at the object's centroid. 1 visually illustrates these setups.

MVP-N [35] dataset contains real-world data with 19 groups and 44 categories of indistinguishable retail products. Each instance has 2-6 images from different views. Notably, MVP-N includes a validation set for better performance assessment. In the training set, each instance includes at least one informative and one uninformative view, while the test and validation sets only provide one informative view.

4.2 Implementation Details

All views' images are resized to $H_o \times W_o \times 3 = 224 \times 224 \times 3$ for our network input. We use ResNet18 [10] as our backbone. In the Multi-View Part-Features Generation module, the interpolated feature maps has a size of $H \times W = 14 \times 14$, the channel numbers for the convolutions on feature maps at different scales are $M_1 = M_2 = M_3 = 32$. The number of part-level features generated for each view is $M = 8$, and the channel number of these features is $D = 512$.

To train our model, we utilize SGD with momentum as the optimizer. For the FG3D dataset, the initial learning rate, weight decay, and momentum are set to 10^{-3}, 10^{-3}, and 0.9, respectively. During the training process, the learning rate decreases following a cosine annealing schedule. For the MVP-N dataset, the only modification is the learning rate, which is adjusted to 4×10^{-3}, while other parameters remain unchanged.

For both viewpoint settings, we train the model with a batch size of 20. Each batch comprises object's 12 images with fixed viewpoints or 4-12 (for FG3D) / 2-6 (for MVP-N) images with arbitrary viewpoints.

We finish all experiments on a RTX-3090 GPU. Our code will be released on https://github.com/RunchuZhang/Attention4Align/.

4.3 Method Comparisons

In 1 and 2, we conducted comparisons between different methods through three experiments on two datasets.

On the FG3D dataset, we used Class Accuracy (Class Acc.) and Instance Accuracy (Inst. Acc.) as evaluation metrics. For the MVP-N dataset, due to balanced sample sizes, we employed Multi-view Accuracy (MVA) as the sole metric. We also report additional metrics for comprehensive comparison: Mean Confidence for Correct Prediction (MCC), Mean Confidence for Wrong Prediction (MCW), Model Size (MS), FLOPs (calculated with 12 images), and Runtime (total time for 1 epoch).

Table 1. The classification performance on the FG3D dataset under the Fixed and Arbitrary viewpoints setting. *indicates that the results are from our implementation.

	Airplane		**Car**		**Chair**	
	Fixed Views	**Arb. Views**	**Fixed Views**	**Arb. Views**	**Fixed Views**	**Arb. Views**
Method	Class/Inst. Acc.	Class/Inst. Acc.	Class/Inst. Acc.	Class/Inst. Acc.	Class/Inst. Acc.	Class/Inst. Acc.
MVCNN [29]	82.57/91.11	—	71.88/76.12	—	76.27/82.90	—
RotationNet [13]	89.11/92.76	—	72.53/75.59	—	78.45/82.07	—
3D2SeqViews [7]	89.41/92.21	—	63.85/66.39	—	76.26/77.10	—
3DViewGraph [9]	88.21/93.85	—	71.65/77.11	—	79.88/83.58	—
View-GCN [37]	87.44/93.58	—	73.68/77.34	—	79.70/83.63	—
SeqV2SeqL [8]	88.52/93.44	—	72.23/75.36	—	79.89/82.54	—
Parts4Feature [6]	82.55/91.39	—	73.42/75.44	—	77.08/81.61	—
MVCNN-new* [29]	88.88/93.31	85.81/92.08	73.57/76.20	69.85/72.55	77.74/82.84	71.95/78.91
GVCNN* [3]	91.33/93.85	87.62/92.62	73.82/76.20	69.48/72.62	80.41/83.78	73.24/78.55
DAN* [21]	90.57/93.85	85.70/92.08	74.65/75.74	69.22/72.09	79.05/82.90	72.57/79.27
CVR* [36]	92.72/93.99	87.56/89.21	74.92/76.27	61.60/65.86	81.06/84.09	72.45/74.87
FG3D-Net [19]	89.44/93.99	—	74.03/**79.47**	—	80.04/83.94	—
Ours	**93.77**/**95.63**	**87.79**/**92.76**	**75.47**/78.48	**71.77**/**73.76**	**81.09**/**84.40**	**76.96**/**79.79**

Table 2. Comparison of various methods on the MVP-N dataset under the Arbitrary viewpoints setting. MS and RT represent Model Size and Runtime. M, G, and S denote million, billion, and seconds.

Method	**Views**	**Val**			**Test**			**MS(M)**	**FLOPs(G)**	**RT(S)**
		MVA	MCC	MCW	MVA	MCC	MCW			
MVCNN-new [29]	2-6 Views	89.29	88.12	65.68	89.35	87.92	65.52	**11.18**	**21.88**	**50.80**
GVCNN [3]	2-6 Views	85.69	82.75	**60.95**	85.42	82.67	**60.55**	24.03	22.04	62.03
DAN [21]	2-6 Views	92.05	85.92	61.92	91.61	86.02	62.11	17.49	21.96	69.15
CVR [36]	2-6 Views	79.95	83.47	65.64	79.99	83.39	64.57	34.35	22.11	138.30
Ours	2-6 Views	**96.61**	**99.11**	84.95	**96.68**	**99.23**	86.26	14.41	22.25	85.42

From 1, it can be observed that our weakly supervised approach surpasses all non-part-based multi-view methods in both fixed and arbitrary viewpoints experiments on the FG3D, including several algorithms that we implemented and fine-tuned ourselves. Moreover, it outperforms the strongly supervised FG3D-net method on the air and chair datasets. Even on the most challenging car dataset, our method remains highly competitive compared to FG3D-net.

2 demonstrates the performance of our method on the validation and test sets in the MVP-N. Our method's MVA and MCC significantly outperform existing methods, reaching high values. Moreover, it can be observed that our model has an advantage in size compared to existing methods, with only a slight increase in FLOPs. However, due to the increase in FLOPs and the additional computation required for solving the assignment problem, our runtime exceeds some methods. Nevertheless, this increase in runtime is acceptable given the performance improvements we achieved, and it remains significantly lower than that of CVR, which also requires additional optimization problem solving.

Simultaneously, we have also noted the relatively high MCW of our method, indicating difficulty in correcting misclassified samples.

we believe it should be considered together with the MCC metric. It can be observed that both metrics have significantly improved compared to the baseline

Table 3. Ablation study on the proposed components with the backbone ResNet18. Baseline: Only Multi-view Part-Features Generation, SA: Part2Part Single Attention Encoder, HA: Part2Part Hierarchical Attention Encoder, PA: Multi-view Part-Features Alignment.

	Airplane		**Car**		**Chair**	
	Fixed Views	**Arb. Views**	**Fixed Views**	**Arb. Views**	**Fixed Views**	**Arb. Views**
Method	Class/Inst. Acc.	Class/Inst. Acc.	Class/Inst. Acc.	Class/Inst. Acc.	Class/Inst, Acc.	Class/Inst. Acc.
(a) Baseline	90.06/94.13	86.16/91.26	72.37/75.21	68.68/71.33	78.39/82.95	73.49/78.47
(b) Baseline+HA	92.03/94.26	87.59/91.80	74.35/76.35	71.3/72.62	80.3/84.15	74.98/**80.05**
(c) Baseline+HA+PA	**93.77**/**95.63**	**87.79**/**92.76**	**75.47**/**78.48**	71.77/**73.76**	**81.09**/**84.4**	**76.96**/79.79
(d) Baseline+SA	91.31/94.13	87.44/91.62	73.53/75.59	70.92/72.47	79.56/83.73	74.41/79.84
(e) WS-DAN [12]+HA+PA	93.05/94.26	86.23/91.26	74.75/75.97	**72.22**/72.85	79.97/83.89	75.27/79.02

(nearly 100%), indicating that our model's predictions tend to be generally high (rather than just having high confidence for misclassified samples).

Therefore, we consider this reasonable, which may be related to the design of our model. For instance, the length of the vector inputted into the classifier is 8*512 instead of the 512 used by other models, or it could be due to using the same parameter settings for fair comparison, resulting in our model not having the optimal parameters.

4.4 Ablation

We conduct ablation studies to validate the effectiveness on the key components of our method on the FG3D dataset. Our baseline method utilizes ResNet18 [10] as the backbone and obtains Part-features through our Multi-view Part-Features Generation module. We evaluate the impact of each module by progressively adding modules on the baseline, as shown in 3 (a)-(c). Meanwhile, to verify the advantages of our method, we conducted experiments (d) and (e): Experiment (d) replaced the Hierarchical attention from Part2Part Encoder in Experiment (b) with a Single attention operation which means all parts from all views were flattened into a sequence and subjected to conventional attention operations. Experiment (e) replaced the operation of fusing multi-scale feature maps in the Generation module with the WS-DAN [12] method which only use the feature map from the final layer, and integrated it with all our other modules.

The impact of fusing multi-scale information. By comparing experiments (a) and (e), it can be observed that fusing multi-scale information leads to improvement in the majority of cases, although some improvements are relatively small. This could be because the fusion of multi-scale information enhances the quality of the generated region attention maps and focuses on the detailed information lost as the network deepens.

The effectiveness of Part2Part Hierarchical Attention Encoder. We conducted experiments (a), (b), (d) to validate the improvements in this aspect. The comparison between experiments (a) and (b)&(d) demonstrates the effectiveness of our P2P Encoder: both attention operations yielded consistent improvements on the FG3D dataset, indicating that parts can enhance their representational capacity by acquiring information from semantically similar parts

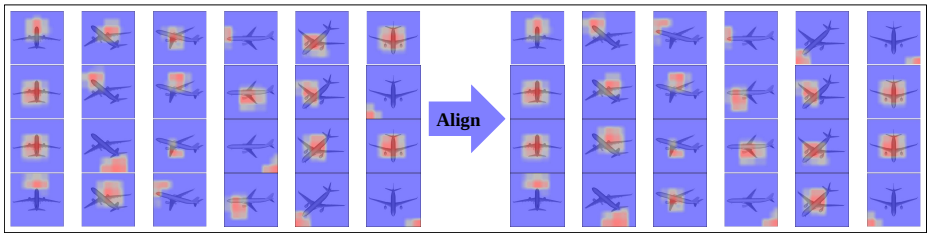

Fig. 5. The Visualization of Multi-view Part Alignment. The images in the same column represent different parts of the same view and the alignment operation only changes the order of parts. For clearer visualization, we preprocessed the attention maps by removing values that are too small or stray.

in other views. Moreover, the comparison between experiments (b) and (d) highlights the advantages of our Hierarchical attention operation over a Single attention operation. Through two levels of attention operations, better information retrieval from parts in other views can be achieved.

The effectiveness of Multi-view Part Alignment operation. The comparison between experiments (b) and (c) reveals a significant improvement in classification performance with the alignment of parts from different views, particularly in the fixed viewpoint setting. This may be attributed to the fact that images in a fixed viewpoint sequence exhibit a gradual transition, making alignment more feasible and accurate. In 5, we visualize attention regions and demonstrate the effectiveness of the alignment module. It can be observed that due to the weakly supervised nature of the part generation module, our part localization results are not as accurate as those of strongly supervised methods, leading to instances of meaningless localization. However, with the influence of the localization module, some semantically similar parts can still be clustered to some extent.

5 Conclusion

We propose a part-based approach for Multi-view 3D Fine-grained Classification, using a weakly supervised method for part generation, feature enhancement, and alignment. Specifically, we introduce Part2Part Hierarchical Attention to enable information transfer between parts from different views and use the attention map for Multi-view Part Alignment to create robust 3D descriptors. Experiments in 4 demonstrate its effectiveness.

This is the first End-to-End part-based model for 3D fine-grained classification. Future work could explore improved Part Generation and Part Aggregation/Alignment.

Acknowledgments. This research was funded by the Interdisciplinary Research Support Program of HUST, grant number 2024JCYJ027

References

1. Behera, A., Wharton, Z., Hewage, P.R., Bera, A.: Context-aware attentional pooling (cap) for fine-grained visual classification. In: AAAI. vol. 35, pp. 929–937 (2021)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
3. Feng, Y., Zhang, Z., Zhao, X., Ji, R., Gao, Y.: Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In: CVPR. pp. 264–272 (2018)
4. Fu, J., Zheng, H., Mei, T.: Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In: CVPR. pp. 4438–4446 (2017)
5. Hamdi, A., Giancola, S., Ghanem, B.: Mvtn: Multi-view transformation network for 3d shape recognition. In: ICCV. pp. 1–11 (2021)
6. Han, Z., Liu, X., Liu, Y.S., Zwicker, M.: Parts4feature: Learning 3d global features from generally semantic parts in multiple views. In: IJCAI. pp. 766–773 (2019)
7. Han, Z., Lu, H., Liu, Z., Vong, C.M., Liu, Y.S., Zwicker, M., Han, J., Chen, C.P.: 3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation. IEEE TIP **28**(8), 3986–3999 (2019)
8. Han, Z., Shang, M., Liu, Z., Vong, C.M., Liu, Y.S., Zwicker, M., Han, J., Chen, C.P.: Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention. IEEE TIP **28**(2), 658–672 (2018)
9. Han, Z., Wang, X., Vong, C.M., Liu, Y.S., Zwicker, M., Chen, C.L.P.: 3dviewgraph: Learning global features for 3d shapes from a graph of unordered views with attention. In: IJCAI. pp. 758–765 (2019)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR. pp. 7132–7141 (2018)
12. Hu, T., Qi, H., Huang, Q., Lu, Y.: See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification. arXiv preprint arXiv:1901.09891 (2019)
13. Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: CVPR. pp. 5010–5019 (2018)
14. Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: ICCV. pp. 863–872 (2017)
15. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: IEEE international conference on robotics and automation. pp. 1817–1824 (2011)
16. Li, Y., Pirk, S., Su, H., Qi, C.R., Guibas, L.J.: Fpnn: Field probing neural networks for 3d data. NeurIPS **29** (2016)
17. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: ICCV. pp. 1449–1457 (2015)
18. Liu, C., Xie, H., Zha, Z.J., Ma, L., Yu, L., Zhang, Y.: Filtration and distillation: Enhancing region attention for fine-grained visual categorization. In: AAAI. vol. 34, pp. 11555–11562 (2020)
19. Liu, X., Han, Z., Liu, Y.S., Zwicker, M.: Fine-grained 3d shape classification with hierarchical part-view attention. IEEE TIP **30**, 1744–1758 (2021)
20. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 922–928 (2015)

21. Nie, W., Zhao, Y., Song, D., Gao, Y.: Dan: Deep-attention network for 3d shape recognition. IEEE TIP **30**, 4371–4383 (2021)
22. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR. pp. 652–660 (2017)
23. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: CVPR. pp. 5648–5656 (2016)
24. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. NeurIPS **30**, 5105–5114 (2017)
25. Rao, Y., Chen, G., Lu, J., Zhou, J.: Counterfactual attention learning for fine-grained visual categorization and re-identification. In: ICCV. pp. 1025–1034 (2021)
26. Sedaghat, N., Zolfaghari, M., Amiri, E., Brox, T.: Orientation-boosted voxel nets for 3d object recognition. BMVC (2017)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. pp. 1–14 (2015)
28. Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.H., Kautz, J.: Splatnet: Sparse lattice networks for point cloud processing. In: CVPR. pp. 2530–2539 (2018)
29. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: ICCV. pp. 945–953 (2015)
30. Sun, M., Yuan, Y., Zhou, F., Ding, E.: Multi-attention multi-class constraint for fine-grained image recognition. In: ECCV. pp. 805–821 (2018)
31. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. vol. 31, pp. 4278–4284 (2017)
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. pp. 1–9 (2015)
33. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. pp. 2818–2826 (2016)
34. 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: ICCV. pp. 1588–1597 (2019)
35. Wang, R., Wang, J., Kim, T.S., KIM, J., Lee, H.J.: Mvp-n: A dataset and benchmark for real-world multi-view object classification. NeurIPS **35**, 20536–20550 (2022)
36. Wei, X., Gong, Y., Wang, F., Sun, X., Sun, J.: Learning canonical view representation for 3d shape recognition with arbitrary views. In: ICCV. pp. 407–416 (2021)
37. Wei, X., Yu, R., Sun, J.: View-gcn: View-based graph convolutional network for 3d shape analysis. In: CVPR. pp. 1850–1859 (2020)
38. Wei, X., Yu, R., Sun, J.: Learning view-based graph convolutional network for multi-view 3d shape analysis. IEEE TPAMI **45**(6), 7525–7541 (2023)
39. Wei, X.S., Song, Y.Z., Mac Aodha, O., Wu, J., Peng, Y., Tang, J., Yang, J., Belongie, S.: Fine-grained image analysis with deep learning: A survey. IEEE TPAMI **44**(12), 8927–8948 (2021)
40. Wright, M.: Speeding up the hungarian algorithm. Computers & Operations Research **17**(1), 95–96 (1990)
41. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR. pp. 1912–1920 (2015)

42. Yang, X., Wang, Y., Chen, K., Xu, Y., Tian, Y.: Fine-grained object classification via self-supervised pose alignment. In: CVPR. pp. 7399–7408 (2022)
43. Yang, Z., Wang, L.: Learning relationships for multi-view 3d object recognition. In: ICCV. pp. 7505–7514 (2019)
44. Zheng, H., Fu, J., Mei, T., Luo, J.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: ICCV. pp. 5209–5217 (2017)