Exploring Cross-video Matching for Few-shot Video Classification via Dual-Hierarchy Graph Neural Network Learning

Fuqin Deng[1,3,4*], Jiaming Zhong[1*], Nannan Li[2,†], Lanhui Fu[1], Dong Wang[1], Tin Lun Lam[3,4,5,†]

***Abstract*—Few-shot video classification methods aim to recognize a new class with only a few training examples. Distinct from previous few-shot methods, we explicitly consider the relations in cross-video domains and take full advantage of the cross-video frame matching in a hierarchy learning fashion. In this paper, we propose a Dual-Hierarchy Graph Neural Network to realize comprehensive cross-video frame matching and video relation modeling. In the first hierarchy of the graph neural network, we build a Cross-video Frame Matching Graph to extract robust frame-level features via accumulating information across frames sampled from both query and support videos. Then, frame representations are accumulated to obtain the video-level features. In the second hierarchy of the graph neural network, we construct a Video Relation Graph by taking the video-level features as nodes, which can adaptively learn positive relations between query and support videos. We get the predicted label of the query video through the matching learning of edges connecting video nodes. We evaluate the model on three benchmarks: HMDB51, Kinetics, and UCF101. Extensive experiments on benchmark datasets demonstrate that our model significantly improves few-shot video classification across a wide range of competitive baselines and showcases the strong generalization of our framework. The source code and models will be publicly available at https://github.com/JiaMingZhong2621/DHGNN.**

***Index Terms*—Video classification, few-shot learning, hierarchy graph neural network**

I. Introduction

In the past few years, with the development of deep learning, video classification has attracted more and more attention in both academic research and industrial applications. Currently, most state-of-the-art video classification methods are based on deep neural networks. However, many of these methods need to manually collect large amounts of labeled video datasets, which is tedious and labor intensive. Therefore, few-shot learning is proposed to tackle this problem.

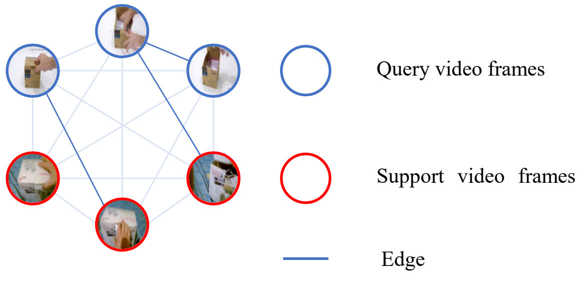

Fig. 1. Example of cross-video frame matching. We find the best-matched frames of the query video to the support video. Nodes of semantically matched frames are connected with a darker edge.

Few-shot learning [1] aims to recognize a new class with a couple of labeled examples. A direct and intuitive approach is to train a model on an extensive dataset and fine-tune it on the target dataset. However, it suffers from overfitting issues. As a promising approach for few-shot learning, meta-learning paradigm has become increasingly popular, which is explicitly used to train the few-shot learning model. For example, Ravi et al. [2] propose to train meta-learners that can perform few-shot learning.

Most few-shot learning methods focus on image classification [3], [4]. Compared with static images, videos contain hundreds of frames with dynamic temporal relations, which is much more challenging than images. Furthermore, the lack of sufficient supervised information is even more severe when only a few labeled samples are provided. Therefore, few-shot video classification remains a challenging problem.

Recently, some efforts [5], [6] have been made to tackle the task. Zhu et al. [5] introduce a multi-saliency embedding algorithm to encode a variable-length video sequence into a fixed-size matrix representation. It can store the long-term information of a sample and classify videos by matching. Hu et al. [6] design a dual-pooling graph neural networks, which aims to model intra-video and inter-video relations. It models intra-video relationships based on a frame-level feature extracted from one video, not explicitly exploiting the matching relation between frame-level features of the query video and support video.

Although having achieved promising performance, these methods fail to consider the relations in cross-video domains.

[*] Authors contributed equally, ranked alphabetically. [1] School of Intelligent Manufacturing, the Wuyi University, Jiangmen, China. [2] School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Macau, China. [3] Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China. [4] The 3irobotix Co.,Ltd, Shenzhen, China. [5] School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen, China. [†] Corresponding authors are Nannan Li nnli@must.edu.mo and Tin Lun Lam tllam@cuhk.edu.cn.

This work was supported in part by the National Key R&D Program of China (2020YFB1313300), Wuyi University-Hong Kong-Macau Joint-Funding-Scheme (2022WGALH17,2021WGALH18), the funding for General Scientific Research of Macau University of Science and Technology (Grant No. FRG-22-102-FIE), the funding from the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AC01202101103), Shenzhen Peacock Plan of Shenzhen Science and Technology Program (KQTD2016113010470345), and the special projects in key fields of Guangdong Provincial Department of Education (2019KZDZX1025).

Inspired by part-based few-shot learning [6], [7], we consider that, within a few-shot regime, it is advantageous to match the query video's frame-level features to those of the support video when constructing video-level features. This better accumulates related feature information by matching frame-level features at various positions in cross-video domains. Fig. 1 shows query video frames match to relevant frames in the support video.

In this paper, we propose a Dual-Hierarchy Graph Neural Network (DHGNN) that jointly models cross-video frame-level and video-level relations for few-shot video classification. As shown in Fig. 2, in order to match the query video's frame-level features to the most relevant ones from the support videos, we construct cross-video frame matching graphs to accumulate the information from cross-video frames. In this type of graph, frames sampled from a video are considered as nodes, and edges connecting two nodes represent the strength of their matching relations. A feature aggregation operation is adopted to extract representations depicting support and query video contents. This video-level feature is then used to construct the video relation graph for video classification.

We construct the video relation graph by taking the video-level features as nodes. Edges connecting two nodes represent relations among videos. The label of the query video can be predicted from that of the support video node, which has the closest representation similarity with the query node. Owing to the dual-hierarchical process in our DHGNN, the most matching relations among frame-level and video-level representations are preserved, which makes the classification results better. Extensive experimental results demonstrate the significant performance of our DHGNN.

Our key **contributions** can be summarised as follows:

- We propose a dual-hierarchy graph neural network to perform few-shot video classification by leveraging cross-video frame matching graph and video relation graph. Our DHGNN achieves significant performance improvement by explicitly considering the matching relation between frame-level features of the query video and support video.

- By adopting a dual-hierarchy structure to construct a cross-video frame matching graph module and a video relation graph module, our proposed model is able to simultaneously consider the accumulation of features at both the cross-video frame level and the video level. This allows for a finer representation of representative video content.

- We evaluate the proposed method in three standard video datasets, i.e., HMDB51 [8], Kinetics [9] and UCF101 [10]. Extensive experiments show that using cross-video frame-level relations can significantly improve the performance of the model.

II. Related Work

In this section, we discuss the methods in terms of video classification, few-shot learning and graph neural networks.

A. Video Classification

With the popularity of intelligent terminals, video has become one of the most common media. Video classification methods have evolved from hand-crafted representation learning [11], [12], [13] to deep-learning-based models. Success in deep learning for image classification has triggered recent progress in video classification. C3D [14] utilizes 3D convolutional filters to extract spatio-temporal features from sequences of RGB frames. ViViT [15] extracts spatio-temporal tokens from the input video, which are encoded by a series of transformer layers. TSN [16] and I3D [9] use two-stream 2D or 3D CNNs on RGB and optical flow sequences. Chen et al. [17] employed several variants of recurrent neural networks (RNN) and generalized Vector of Locally Aggregated Descriptors (VLADs) to aggregate the given frame-level feature representations and produce video-level predictions. An issue of these methods is their dependence on large-scale video datasets for training. Models with many learnable parameters tend to fail when only a few training samples are available. Cao et al. [7] proposed a model explicitly leverages the temporal ordering information in video data through ordered temporal alignment. Bishay et al. [18] utilize attention mechanisms to perform temporal alignment and learn a deep-distance measure on the aligned representations at the video segment level. Hu et al. [6] designed a novel GNN to model intra-video and inter-video relations. However, these methods inevitably neglect to match the query video's frame-level features to the most relevant ones from the support video.

B. Few-shot Learning

A direct approach to few-shot learning is to train a model on a large dataset and fine-tune it with the small data samples. Since limited samples from novel classes are insufficient to fine-tune the model with general deep-learning techniques, some methods are proposed to learn an initialization model. We can classify these methods into three categories. 1) The Memory Network [19], [20] methods extract knowledge from the training set and store it in a memory module. A key-value memory module [21] is usually used in few-shot learning. The memory slot of the key-value stores the most similar sample representation, which is used as input to a simple classifier to make the prediction. 2) Generative methods [22] learn the probability distribution from the training set with the help of prior knowledge. Gordon et al. [23] develop a general framework for few-shot learning approximate probabilistic inference for prediction. Lake et al. [24] proposed a computational model that captures human learning abilities for a wide range of simple visual concepts. 3) Metric-based methods [25] learn a similarity model to find the most similar class for the target sample among a small set. Matching Network [25] calculates the matching score of the query and the support images to obtain the image label. Prototype Network [26] obtains classification results by computing distances to prototype representations of each class. Relation Net [27] utilizes a deep distance metric to map the sample to an identical feature space and outputs a similarity score within episodes, each designed to simulate the few-shot setting. However, few-shot video classification requires joint consideration of relations in both cross-video frame and inter-video domains, which still needs further exploration.

C. Graph Neural Networks

Graph Neural Network (GNN) is introduced as a solution to process data in non-Euclidean space, which is challenging for traditional Convolutional Neural Networks (CNNs). Gori et al. [28] first introduced the concept of GNN, which can construct graph structure data. Graph neural networks can be categorized into spectral-based methods and non-spectral-based methods. Spectral-based approaches draw inspiration from classical CNNs and utilize multiple spectral convolutional layers operating in the Fourier domain. Henaff et al. [29] propose to apply an input smoothing kernel and use the corresponding interpolated weights as the filter parameters for graph spectral convolutions. Defferrard et al. [30] mainly focuses on learning graph representations in the spectral domain, which are represented by the regularized Laplacian matrix. Non-spectral-based methods generalize through aggregations of graph signals within the node neighborhood, passing messages to the next layer without explicitly defining the convolution operation. Schlichtkrull et al. [31] introduced relational graph convolutional networks and apply them to handle link prediction and entity classification tasks. Kim et al. [32] further propose a novel edge-labeling GNN, which learns to predict the edge-labels rather than the node-labels on the graph by iteratively updating the edge-labels. Most of the aforementioned methods are based on the instance-level relation between examples and the model. To enhance the discriminative ability and accurately select the representative video content while refining video relations, Hu et al. [6] propose a dual-pool graph neural network. Wang et al. [33] represented video frames as graph nodes and obtained edges using attention mechanisms, proposing an attention graph neural network. While the method learns multistage representations, it has a limitation in effectively accumulating matched frame-level features across different videos. Therefore, we propose hierarchical GNNs to jointly consider relations across both cross-video and inter-video domains.

III. Methods

In this section, we will introduce the proposed Dual-Hierarchy Graph Neural Network model, the framework of which is shown in Fig. 2. It can be divided into three parts: video frame feature extraction, the cross-video frame matching graph module, and the video relation graph module. In the following paragraphs, we first show the formulation of the few-shot video classification and then sequentially introduce the architecture detail of the cross-video frame matching graph and video relation graph modules. Finally, we give the details of the loss function for optimizing the model in an end-to-end fashion.

A. Problem Formulation

Few-shot video classification tasks aim to train a classification model that can generalize well in cases where only a few samples are available. Given training data D_{train}, each few-shot video classification task τ consists of a support set S and a query set Q. Specifically, in an N-way, K-shot problem, the support set contains N classes with K samples for each class.

TABLE I
SYMBOL DESCRIPTION

symbol	comment
S	support set
Q	query set
T	number of query samples
L	number of frames
$g(\cdot)$	feature extractor
$pos(\cdot)$	position encoding
$G_{frame} = (A_f, V_f)$	cross-video frame matching graph
$G_{video} = (A_v, V_v)$	video relation graph
$\hat{F}^s$	support video feature from G_{frame}
$\hat{F}^q$	query video feature from G_{frame}
f_{avg}	average feature function
ϕ	Conv-BN-LeakRelu blocks
v_s	video-level feature of the support video
v_q	video-level feature of the query video
$\wedge$	union set

Similar to prior works [6], [7], [34], we adopt episodic training [20], [25] for standard few-shot settings. In each episode, we consider an N-way, K-shot classification problem. Let $S = \{(x_i, y_i)\}_{i=1}^{N\times K}$ be the support set, and $Q = \{(x_i, y_i)\}_{i=N\times K+1}^{N\times K+T}$ be the query set, consisting of input data-label pairs. Here, T is the number of query samples, and $y_i \in \{C_1, ..., C_N\}$ represents the class labels. In the training stage, data labels are provided for both support set S and query set Q. Given testing data D_{test}, the goal is to determine the class label each query video belongs to. It is important to note that the labels of the training set and testing set are mutually exclusive.

B. Preliminaries: Graph Neural Networks

We adopt GNNs to learn the edges between the nodes of a specific graph and present a graph G as (A, V), where $A \in (0, 1)^{L\times L}$ is the adjacency matrix, and $V \in \mathbb{R}^{L\times D}$ is the node feature. To learn node representation, we update our GNNs iteratively via the following equation:

$$H^{l+1} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^l W^l\right) \tag{1}$$

where I is the identity matrix, $\widetilde{A} = I + A$ is equivalent to adding a self-loops to the original adjacency matrix A, $\widetilde{D}$ is the diagonal degree matrix of $\widetilde{A}$, and W^l is the trainable matrix for feature representation. H^l is the node embeddings computed after l updatings of the GNNs, i.e., $H^l = V$. The final output node embeddings are represented as $H^{l+1} \in \mathbb{R}^{L\times D}$, where L is the number of nodes in the graph, and D represents the dimension of the node embeddings. For simplicity, we use the above GNN operator to keep this paper self-contained.

C. Cross-Video Frame Matching Graph Module

Our key insight is to build a graph network to explicitly retain the most discriminative video content and accumulate

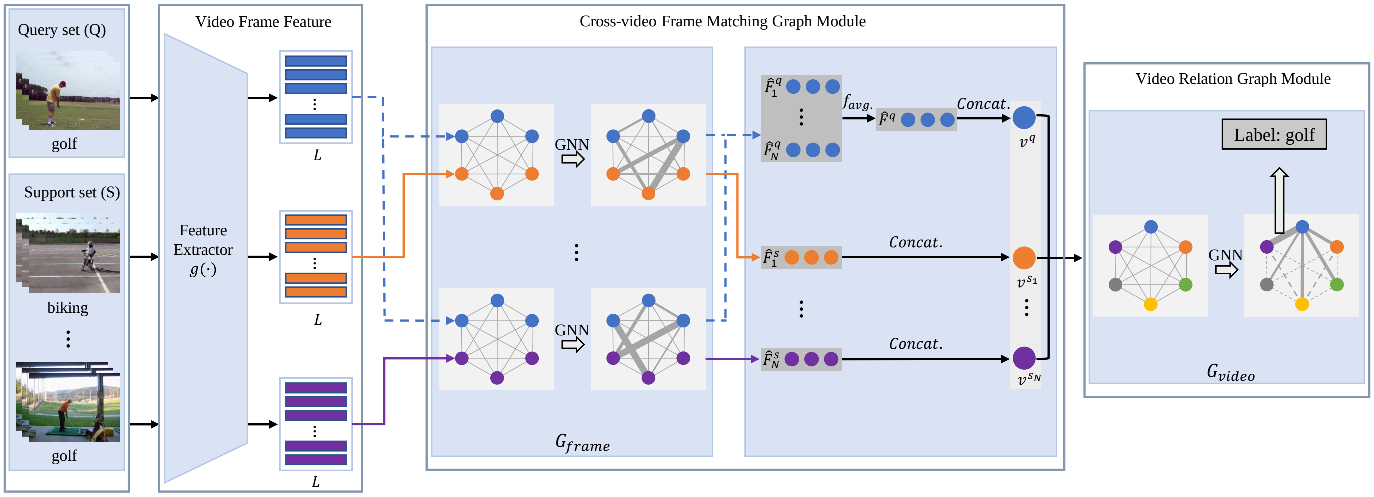

Fig. 2. Illustration of the framework on a 5-way 1-shot problem: First, the feature extractor $g(\cdot)$ is used to extract the per-frame features of the support and query videos. The frame-level features of each query video and support video are represented as nodes to construct a cross-video frame matching graph G_{frame}, where each edge represents the relationship between nodes. Then, by applying the feature averaging function f_{avg} and the feature concatenation operation $Concat$ to the nodes of the cross-video frame matching graph, we obtain the video-level features v^s and v^q. These features serve as nodes to construct a video relation graph G_{video}, where edges with darker colors imply stronger relevancy. The video relation graph is responsible for learning the positive relation between query and support videos, and the class label is finally obtained by predicting the edge.

matching frame-level features between different videos, where matched frame nodes within the query and support videos are tightly associated.

To achieve the goal, in the first hierarchy of the graph neural network, we propose a cross-video frame matching graph to relate frame features across videos, as illustrated in Fig. 2. First of all, for each video, we follow the sampling strategy of TSN [16], which divides the video into L segments and uniformly samples from each segment. This allows each video to be represented by a fixed length. Given an input sequence $X^i = \{x_1^i, ..., x_L^i\}$, we extract the sequence features with a feature extractor. Hence, each video is encoded into an $L \times D$-dimensional vector matrix. Let $X^q = \{x_1^q, ..., x_L^q\}$, $X^s = \{x_1^s, ..., x_L^s\}$ represent a query video and a support video with L uniformly sampled frames, respectively. We use position encoding of video frames to maintain temporal relationships and define the query and support representation as follows:

$$\begin{aligned} F^q &= g(X^q) + pos(X^q) = \{\hat{x}_1^q, \hat{x}_2^q, ..., \hat{x}_L^q\} \\ F^s &= g(X^s) + pos(X^s) = \{\hat{x}_1^s, \hat{x}_2^s, ..., \hat{x}_L^s\} \end{aligned} \tag{2}$$

Where $g(\cdot)$ represents a ResNet-50 [35] network, which encodes the input sequence into D-dimensional features. The function $pos(\cdot)$ denotes the positional encoding of frame indices.

Then, we construct a cross-video frame matching graph $G_{frame} = (A_f, V_f)$, where the edge $A_f \in (0,1)^{2L \times 2L}$ reflects the matched scores between the central and adjacent nodes. $V_f = (F^q \wedge F^s) \in \mathbb{R}^{2L \times D}$ is the feature matrix of frame sequences sampled from a query-support video pair. Actually, we build a series of the cross-video frame matching graphs, each of which corresponds to a query-support video pair.

We learn edge A_f as a trainable function, specifically by using a multilayer perceptron ϕ stacked over $abs(\cdot)$, as shown in Eq. 3. $abs(\cdot)$ is an absolute difference between the features of two nodes in the graph. ϕ contains two Conv-BN-LeakRelu blocks with the parameter set θ_f and a softmax layer.

$$A_f = SoftMax\left(\phi(abs(V_f)) + I\right) \tag{3}$$

Given node feature matrix V_f and the edge $A_f \in \mathbb{R}^{2L \times 2L}$, we utilize Eq. 1 to comprehensively accumulate frame-level features of the query video with related frame-level features from the support video. After that, we represent the video feature by aggregating individual node descriptors within a query or support video. Specifically, for the support video representation, we concatenate the frame-level node features, while for the query video, we first average corresponding node features from different query-support video pair graphs and then concatenate them. The aggregation process can be described by the following equation:

$$\begin{aligned} v^s &= Concat(\hat{F}^s) \\ v^q &= Concat\left(f_{avg}(\hat{F}_1^q, \hat{F}_2^q, ..., \hat{F}_N^q)\right) \end{aligned} \tag{4}$$

where $\hat{F}^s$ and $\hat{F}_i^q$ represent the output feature representations of G_{frame}. v^s and v^q denote the aggregated representations of the support and query videos, respectively. f_{avg} is a feature averaging function, $Concat$ represents the feature concatenation operation, N is the N-way. Through the proposed cross-video frame matching graph, we accumulate frame-level features in the query video with related frame-level features in the support video. The final remained video features can participate in the subsequent construction of the video relation graph.

D. Video Relation Graph Module

In the second hierarchy of the graph neural network, we consider the relations between videos in the graph model to achieve the few-shot video classification task. As illustrated in Fig. 2, we further construct the videos in the support set and the query set into a video relation graph G_{video}:

$$G_{video} = (A_v, V_v) \tag{5}$$

where $V_v = \{v_1^s, ..., v_N^s, ..., v_1^q, ..., v_T^q\}$ represents the video-level node feature matrix, and $A_v \in (0,1)^{(N+T)\times(N+T)}$ is the edges, representing the similarity between the video-level nodes. Similar to Eq. 3, the edges are defined as follows:

$$A_v = SoftMax\left(\phi(abs(V_v)) + I\right) \tag{6}$$

We utilize Eq. 1 to update node information through graph convolution to relate query videos with matched support videos. Therefore, from the definition of edge in Eq. 6, we can observe that the value of the edge reflects the degree of matching between two distinct videos.

E. Model learning

Within our framework, the edge-label prediction can be obtained from the final edge feature, i.e., $y^{q,s} \in A_v$. Here, $y^{q,s}$ can be considered as the class probability. Therefore, the label of query videos can be acquired by feeding the corresponding edges of the model into the softmax function:

$$\hat{y}^{q,s} = SoftMax\left(y^{q,s}\right) \tag{7}$$

Given the ground-truth labels y, our framework is learned end-to-end using the standard cross-entropy (CE) loss on the class probabilities $\hat{y}^{q,s}$, which is given by the following loss function:

$$Loss = CE(y, \hat{y}^{q,s}) \tag{8}$$

IV. Experiments

In this section, we evaluate the performance of the proposed model on few-shot video classification task with three public datasets, i.e., Kinetics, UCF101, and HMDB51. We describe the details of the experimental settings and compare our model with a wide range of baselines. In addition, we conduct detailed ablation studies to analyze the properties of the DHGNN.

A. Datasets

Kinetics dataset [9]: For the Kinetics dataset, we adopt the same split as [5], where we sample 64 classes for meta-training, 12 classes for validation, and 24 classes for meta-testing. The selected classes for each split do not overlap with each other.

UCF101 [10]: UCF101 contains 101 action categories with 13320 videos. The categories can be classified into five types (Body motion, Human-human interactions, Human-object interactions, Playing musical instruments, and Sports). In this work, we construct a few-shot dataset following the same rule as [5].

HMDB51 [8]: HMDB51 dataset consists of 6766 video clips from 51 action categories, with each category containing at least 101 clips. In our evaluation, we followed the split proposed by Zhang et al. [36] to assess the performance of our method.

B. Baselines.

Specifically, we evaluate our method on the standard 5-way 1-shot and 5-way 5-shot classification tasks, reporting average results over 10000 tasks randomly selected from the test sets. We compare our method against state-of-the-art few-shot classification methods, including DPGNN [6], TARN [18], ProtoNet [26], ARN [36], OTAM [7], CMN [5], TA2N [37], HCL [38], and TRX [39].

C. Implementation Details

We evaluate our model by conducting 5-way 1-shot and 5-way 5-shot experiments. For each few-shot video classification task, we randomly sample N classes, and each class has K examples, while an additional T unlabeled example belonging to one of the N classes is used for testing. In this setting, each episode contains $N \times K + T$ examples.

For all datasets, we uniformly sample 16 frames (each node contains two frames, a total of 8 nodes) of each video and follow the basic image preprocessing procedure. Following [37], the extracted frames are first resized to 256×256, and a random horizontal flip is applied. During training, a random crop with a size of 224×224 is also applied. To ensure a fair comparison with previous methods, we use the ImageNet pre-trained ResNet-50 as the feature extractor, resulting in 2048-dimensional feature vectors obtained before the last layer of ResNet-50. Our method is implemented using PyTorch, and we utilize the SGD optimizer with a learning rate of 0.01, training for 25,000 tasks. At iteration 3000, the learning rate is decreased by 0.1. We average gradients and perform backpropagation once every 16 iterations.

As shown in Fig. 2, our model consists of three parts: video frame feature extraction, the cross-video frame matching graph module, and the video relation graph module. We utilize the output from the feature extractor as the nodes for the cross-video frame matching graph. The number of nodes in the graph is 16, and the GNN applied to the graph has three layers with output dimensions of 1024, 512, and 256. Each query video will construct a cross-video frame matching graph with the support video. We obtain the video-level features representation of 2048 dimensions by concatenating 8 frame-level features, which are used as a node for video relation graph construction. The GNN for the video relation graph consists of two layers, and the output dimensions of these layers are 1024 and 512, respectively.

D. Performance Comparison

Tab. II shows comparison results of our proposed method and other approaches on the three standard datasets. Our approach demonstrates competitiveness with state-of-the-art methods on the Kinetics dataset, using the same evaluation metrics. Out of the compared methods, ARN [36] shows the lowest performance. DPGNN [6] achieves an accuracy of 80.7% in the 5-shot setting. The results indicate the capability of graph neural networks. The recent works of TA2N [37], HCL [38], and TRX [39] achieve comparable classification accuracies of 85.8%, 85.8%, and 85.9%, respectively. Compared

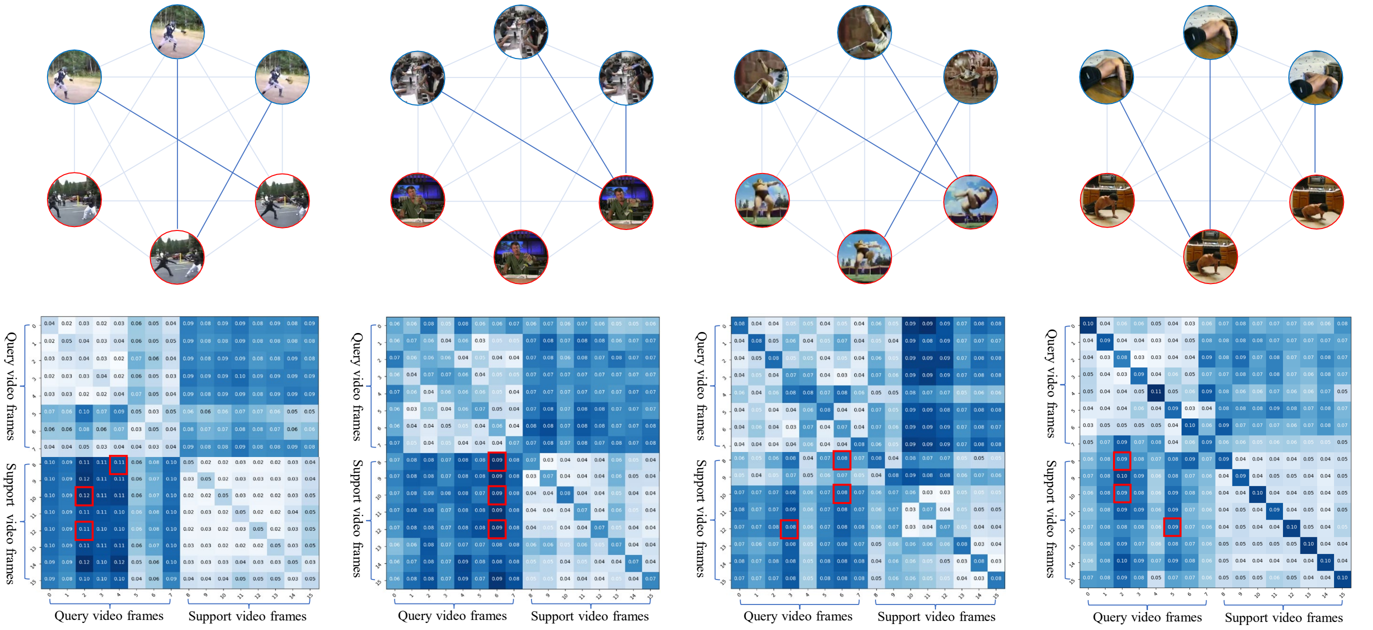

Fig. 3. Examples for cross-video frame matching graph module. We find the best-matched frames of the query video (blue) to the support video (red). Nodes of semantically best-matched frames are connected with a darker edge. The matching score matrix below reflects the matching degree of the query video frame related to the support video frame, where the best matching occurs at different temporal positions.

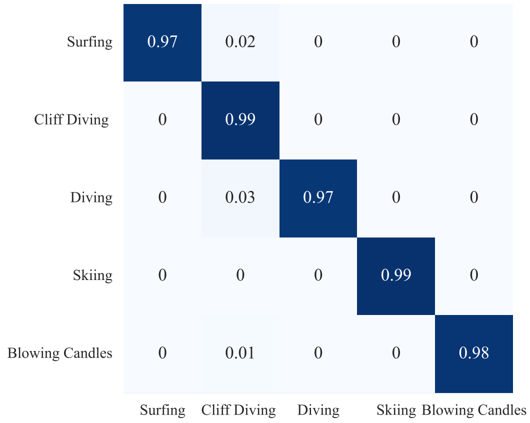

Fig. 4. The visualization of edge prediction. The darker the cell is, the higher confidence it represents. The left axis stands for the query video, and the bottom axis stands for the support video.

TABLE II
PERFORMANCE COMPARISONS BETWEEN OUR PROPOSED APPROACH AND OTHER METHODS UNDER STANDARD 5-WAY 1-SHOT AND 5-WAY 5-SHOT SETTINGS

Method	*Backbone*	*Kinetics*		*UCF101*		*HMDB51*	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ARN [36]	ResNet-50	45.5	60.6	66.3	83.1	45.5	60.6
CMN [5]	ResNet-50	60.0	78.9	-	-	-	-
ProtoNet [26]	ResNet-50	64.5	77.9	74.0	89.6	54.2	68.4
TARN [18]	ResNet-50	64.8	78.5	-	-	-	-
DPGNN [6]	ResNet-50	66.8	80.7	67.7	84.7	-	-
OTAM [7]	ResNet-50	73.0	85.8	79.9	88.9	54.5	66.1
TA2N [37]	ResNet-50	72.8	85.8	81.9	95.1	59.7	73.9
HCL [38]	ResNet-50	**73.7**	85.8	82.5	93.9	59.1	76.3
TRX [39]	ResNet-50	-	85.9	-	**96.1**	-	75.6
Ours:DHGNN	ResNet-50	72.5	**88.3**	**83.6**	95.2	55.8	**76.7**
TRX [39]	ViT	-	90.6	-	96.9	-	79.7
Ours:DHGNN	ViT	**80.8**	**91.1**	**92.2**	**97.2**	**67.9**	**82.3**

with them, our method significantly improves the baselines in the 5-shot task. Especially compared with the graph-based methods, the accuracy of the 5-shot setting can achieve 88.3%, which exceeds DPGNN by 7.6%. These results validate the robust representation capabilities of our dual-hierarchical strategy.

On the HMDB51 dataset, the pros and cons of different methods are roughly the same as the Kinetics. In the 5-shot setting, our proposed method improves the accuracy by 1.1%. However, there is a noticeable decrease in the 1-shot setting. This shows that one-shot classification is still a challenging task. On UCF101, it exceeds the state-of-the-arts on 1-shot settings. UCF101 is an easy dataset when used as a few-shot video classification task, where matching is less important. To further validate our contributions, we substitute the ResNet-50 backbone with ViT [40]. Despite the use of this more robust backbone, our method still achieves impressive performance gains compared to TRX, indicating the strong generality of our method. Overall, the proposed method is superior to the other existing graph-based methods and particularly benefits from the presence of a number of videos in the support set (i.e., few-shot rather than one-shot).

Fig. 3 shows the matching results of the cross-video frame matching graph. We show query video frames (blue) with their corresponding support video frames (red), and the edge indicates the matching score between the video frames. For example, the query video frame in the first example (node 1) gets a higher matching score (The first red rectangle) when matched to the support video frames (node 3). We select three nodes and three edges (red rectangular boxes in the matching score matrix) to highlight each case. The figure shows that the query video frames match different support video frames.

Our proposed model can make discrimination among confusable classes. As shown in Fig. 5, for the commonly easily confused pairs of classes, such as 'Skiing' vs. 'Surfing', 'Diving' vs. 'Cliff Diving', our DHGNN model can predict correct classification results with high accuracy. But for highly similar actions, such as 'kick' and 'kick ball', the classification ability of the model still needs to be improved. Additionally, we visualize the edge of the video graph, which is used for predictions of five query videos, as shown in Fig. 4. The heatmap shows that our model makes the right predictions for five query samples. Notably, our model not only contributes to predicting more accurately but also enlarges differences between videos of different classes by making video-level features more discriminative, which makes the prediction heatmap clean and clear.

E. Ablation Experiments

To study the contribution of DHGNN, we conduct a series of detailed ablation studies. Unless otherwise stated, we adopt the ResNet-50 model as the default setting for comparative experiments.

TABLE III
ABLATION EXPERIMENTS CROSS-VIDEO FRAME MATCHING GRAPH MODULE

Method	*Kinetics*		*UCF101*		*HMDB51*	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
w/o cross-video matching	61.6	80.6	67.0	91.9	47.7	67.7
w cross-video matching	**72.5**	**88.3**	**83.6**	**95.2**	**55.8**	**76.7**

First, we conduct ablation studies on the cross-video frame matching graph module to validate its importance. We carry out experiments by removing the cross-video frame matching graph module from our model. We show the ablation performance of the three datasets in Tab. III. The results are reported in the 5-way 1-shot and 5-way 5-shot settings.

On the Kinetics and HMDB51 datasets, the cross-video frame matching graph module outperforms the no-cross-video frame matching graph module in the 5-way 1-shot setting, achieving improvements of 10.9% and 8.1% respectively. From the 1-shot setting to the 5-shot setting, the classification accuracy is further boosted by 7.7% and 9%. Similarly, on the UCF101 dataset, the cross-video frame matching graph module demonstrates improvements of approximately 16.6% in the 1-shot setting and 3.3% in the 5-shot setting. These experimental results validate the effectiveness of the cross-video frame matching graph module in enhancing performance across the three benchmark datasets.

As presented in Tab. IV, we employ cosine similarity along with our proposed edge calculation method to classify query videos from different categories in the Kinetics dataset. The reported results are based on the 1-shot setting for the 5-way classification task. Our method outperforms using cosine similarity alone for edge calculation, leading to significantly improved classification accuracy. These findings highlight the effectiveness of our edge calculation method in enhancing performance.

TABLE IV
THE IMPACT OF DIFFERENT EDGE COMPUTING APPROACHES ON ACCURACY.

Method	*UCF101*	*Kinetics*	*HMDB51*
Cosine Similarity	62.6	53.9	41.2
DHGNN(Ours)	**83.6**	**72.5**	**55.8**

TABLE V
COMPARISON OF MODEL PARAMETERS, MEMORY USAGE, AND COMPUTATIONAL TIME ACROSS DIFFERENT METHODS

Method	*params(M)*	*memory(M)*	*time@100(s)*
ProtoNet*	23.51	534.28	≥ 23
OTAM*	23.51	534.28	≥ 27
TA2N [37]	39.86	612.52	≥ 34
DHGNN(Ours)	80.71	802.12	≥ 52

To comprehensively understand the parameter size, maximum memory consumption for one 1-shot inference, and computational time required by DHGNN and other methods to complete 100 tasks, we conducted a thorough analysis, as presented in Table V. Notably, the "*" notation indicates data derived from our replication results. Among these methods, ProtoNet extracts video features using CNN, then obtains class prototypes by averaging the features and calculates prototype similarity. OTAM employs the Dynamic Time Warping algorithm to obtain similarity scores. As they do not introduce additional modules, their parameters, inference time, and memory consumption are lower compared to our method. TA2N introduces lightweight modules like the Temporal Transform Module and Action Coordinate Module, resulting in a notable increase in resource consumption. Although our method also introduces additional modules, leading to increased parameters, inference time, and memory usage, it brings substantial performance improvement. For instance, in the 5-shot classification task on the HMDB51 dataset, our method outperforms ProtoNet, OTAM, and TA2N by 8.3%, 10.6%, and 2.8%, respectively.

We compared the performance of different network depths in the cross-video frame matching graph and the video relation graph to explore the classification accuracies. The summarized results are shown in Fig. 6. In our analysis, we focused on the 5-way 1-shot setting as an example and kept the depth of the video relation graph as one while varying the depth of the cross-video frame matching graph, and vice versa. We experimented with different network depths by adjusting the number of graph neural network (GNN) layers. The results indicate that the performance improves as the depth of the

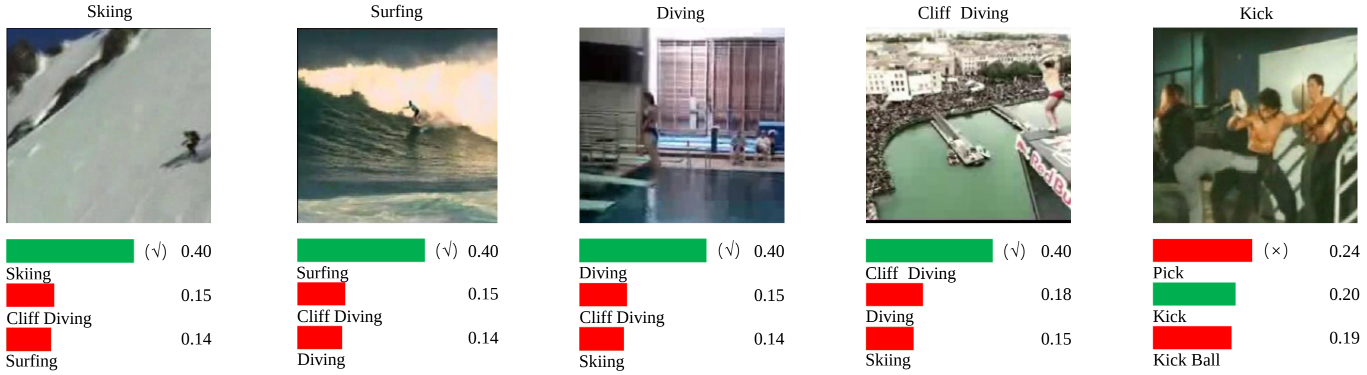

Fig. 5. Classification accuracy on the 5-way 1-shot setting. The label above the video indicates the ground truth label, and the bars below show model predictions sorted in decreasing accuracies. Green and red distinguish correct ($\surd$) and incorrect ($\times$) predictions, respectively.

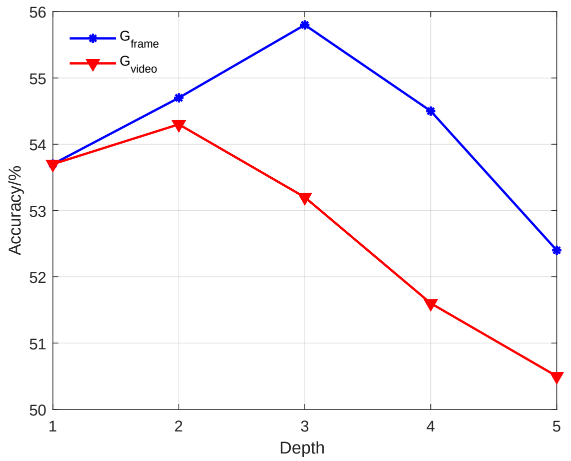

Fig. 6. Effect of different depth of cross-video frame matching graph G_{frame} and video relation graph G_{video} in terms of classification accuracy on the HMDB51 dataset.

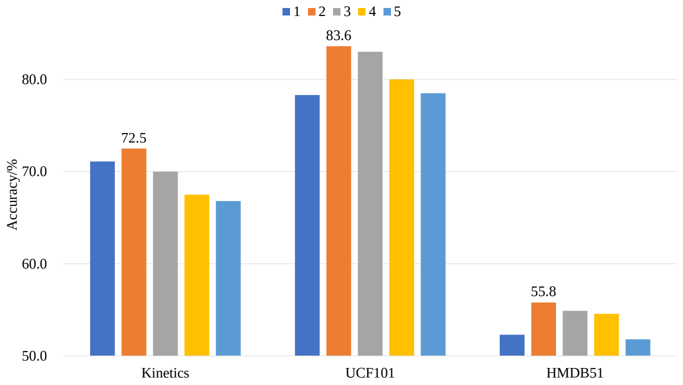

Fig. 7. Performance comparison of different frame numbers of the node in the cross-video frame matching graph module.

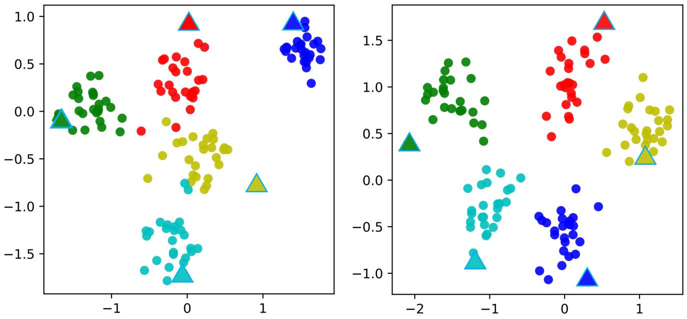

Fig. 8. t-SNE feature embedding of query videos (triangles) and support prototypes (circles) on HMDB51 and UFC101. Left/Right: HMDB51/UFC101

cross-video frame matching graph increases, up to a depth of 3. However, in the video relation graph, we observed that the highest accuracy is achieved when the graph network has a depth of 2. Deeper networks tend to suffer from overfitting and feature smoothing, which hampers further improvement in accuracy.

Furthermore, we analyze the effect of different frame numbers of each node on the classification accuracy. The results are illustrated in Fig. 7. We conduct ablation experiments in the 5-way 1-shot setting and only change the frame number of each node in the cross-video frame matching graph module while keeping the rest of the model as same. We find that the optimal number of frames is 2, which has the best classification accuracies on all three datasets. When too many frames are input, the model might be overwhelmed by a surplus of redundant information, impeding its ability to precisely recognize and classify. Thus, appropriately selecting and extracting key video frames is crucial to enhance the model's classification accuracy.

We further visualize the feature embedding of the query and the prototype of support videos by the t-SNE [41] method, as shown in Fig. 8. We can observe that each cluster is close to its respective prototype, which proves that our model can learn a consistent feature representation for query and support videos from the same class.

V. Conclusion

In this paper, we propose a novel Dual-Hierarchy Graph Neural Network model to address the few-shot video classification problem. The model consists of three parts: video frame feature extraction, the cross-video frame matching graph module, and the video relation graph module. During the feature extraction stage, the feature extractor captures video frame features from the input video sequence. In the first hierarchy of the graph neural network, the cross-video frame matching graph module establishes associations between related frames across different videos. This module accumulates information within the matched frame nodes to obtain robust frame-level features. In the second hierarchy of the graph neural network, we aggregate frame-level representations by concatenating them to obtain video-level features, which are used as nodes to construct a video relation graph. The video relation graph module dynamically learns positive relations between video pairs.

Through extensive experiments on three public datasets (HMDB51, Kinetics, and UCF101), our model achieves competitive results compared to state-of-the-art methods. Furthermore, the ablation study reveals that our model's success in few-shot video classification is primarily attributed to its explicit consideration of relations among video frames from both query and support videos.

References

[1] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[2] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International conference on learning representations*, 2017.

[3] J. Cao, Z. Yao, L. Yu, and B. W.-K. Ling, "Wpe: Weighted prototype estimation for few-shot learning," *Image and Vision Computing*, p. 104757, 2023.

[4] S. Wang, R. Ma, T. Wu, and Y. Cao, "P3dc-shot: Prior-driven discrete data calibration for nearest-neighbor few-shot classification," *Image and Vision Computing*, vol. 136, p. 104736, 2023.

[5] L. Zhu and Y. Yang, "Compound memory networks for few-shot video classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 751–766, 2018.

[6] Y. Hu, J. Gao, and C. Xu, "Learning dual-pooling graph neural networks for few-shot video classification," *IEEE Transactions on Multimedia*, vol. 23, pp. 4285–4296, 2021.

[7] K. Cao, J. Ji, Z. Cao, C.-Y. Chang, and J. C. Niebles, "Few-shot video classification via temporal alignment," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10615–10624, 2020.

[8] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in *2011 International Conference on Computer Vision*, pp. 2556–2563, 2011.

[9] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017.

[10] K. Soomro, A. R. Zamir, and M. Shah, "A dataset of 101 human action classes from videos in the wild," *Center for Research in Computer Vision*, vol. 2, no. 11, 2012.

[11] Y.-G. Jiang, Z. Wu, J. Tang, Z. Li, X. Xue, and S.-F. Chang, "Modeling multimodal clues in a hybrid deep learning framework for video classification," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3137–3147, 2018.

[12] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *BMVC 2008-19th British Machine Vision Conference*, pp. 275–1, British Machine Vision Association, 2008.

[13] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, pp. 3551–3558, 2013.

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

[15] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6816–6826, 2021.

[16] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*, pp. 20–36, Springer, 2016.

[17] S. Chen, X. Wang, Y. Tang, X. Chen, Z. Wu, and Y.-G. Jiang, "Aggregating frame-level features for large-scale video classification," *arXiv preprint arXiv:1707.00803*, 2017.

[18] M. Bishay, G. Zoumpourlis, and I. Patras, "Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition," *In British Machine Vision Conference*, 2019.

[19] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*, pp. 1842–1850, PMLR, 2016.

[20] T. Munkhdalai and H. Yu, "Meta networks," in *International conference on machine learning*, pp. 2554–2563, PMLR, 2017.

[21] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," *arXiv preprint arXiv:1606.03126*, 2016.

[22] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[23] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E. Turner, "Meta-learning probabilistic inference for prediction," *arXiv preprint arXiv:1805.09921*, 2018.

[24] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[25] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016.

[26] X. Zhu, A. Toisoul, J.-M. Perez-Rua, L. Zhang, B. Martinez, and T. Xiang, "Few-shot action recognition with prototype-centered attentive learning," *arXiv preprint arXiv:2101.08085*, 2021.

[27] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

[28] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, pp. 729–734, IEEE, 2005.

[29] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[30] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[31] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pp. 593–607, Springer, 2018.

[32] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11–20, 2019.

[33] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9236–9245, 2019.

[34] L. Yang, L. Li, Z. Zhang, X. Zhou, E. Zhou, and Y. Liu, "Dpgn: Distribution propagation graph network for few-shot learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13387–13396, 2020.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision–ECCV 2016: 14th European Conference,*

Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, pp. 630–645, Springer, 2016.

[36] H. Zhang, L. Zhang, X. Qi, H. Li, P. H. Torr, and P. Koniusz, “Few-shot action recognition with permutation-invariant attention,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pp. 525–542, Springer, 2020.

[37] S. Li, H. Liu, R. Qian, Y. Li, J. See, M. Fei, X. Yu, and W. Lin, “Ta2n: Two-stage action alignment network for few-shot action recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 1404–1411, 2022.

[38] S. Zheng, S. Chen, and Q. Jin, “Few-shot action recognition with hierarchical matching and contrastive learning,” in *European Conference on Computer Vision (ECCV)*, pp. 297–313, Springer, 2022.

[39] T. Perrett, A. Masullo, T. Burghardt, M. Mirmehdi, and D. Damen, “Temporal-relational crosstransformers for few-shot action recognition,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 475–484, 2021.

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[41] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.