Learning to Model Relationships for Zero-Shot Video Classification

Junyu Gao, Tianzhu Zhang, *Member, IEEE*, and Changsheng Xu, *Fellow, IEEE*

Abstract—With the explosive growth of video categories, zero-shot learning (ZSL) in video classification has become a promising research direction in pattern analysis and machine learning. Based on some auxiliary information such as word embeddings and attributes, the key to a robust ZSL method is to transfer the learned knowledge from seen classes to unseen classes, which requires relationship modeling between these concepts (e.g., categories and attributes). However, most existing approaches ignore to model the explicit relationships in an end-to-end manner, resulting in low effectiveness of knowledge transfer. To tackle this problem, we reconsider the video ZSL task as a task-driven message passing process to jointly enjoy several merits including alleviated heterogeneity gap, low domain shift, and robust temporal modeling. Specifically, we propose a prototype-sample GNN (PS-GNN) consisting of a prototype branch and a sample branch to directly and adaptively model all the relationships between category-attribute, category-category, and attribute-attribute. The prototype branch aims to learn robust representations of video categories, which takes as input a set of word-embedding vectors corresponding to the concepts. The sample branch is designed to generate features of a video sample by leveraging its object semantics. With the co-adaption and cooperation between both branches, a unified and robust ZSL framework is achieved. Extensive experiments strongly evidence that PS-GNN obtains favorable performance on five popular video benchmarks consistently.

Index Terms—Zero-shot video classification, graph neural networks, zero-shot learning, deep attention model

1 Introduction

The last decade has seen a growing trend towards intelligent video understanding. With the advances in transmission technologies, digital devices, and display techniques, video is fast becoming one of the most popular media for communication, public security, and entertainment. For example, the world's largest video sharing platform, Youtube, receives more than 300 hours of HD quality videos in every minute.[1] To understand such a large-scale data with rich and complex semantics, effectively classifying various categories such as human actions and events is fundamental, which plays a critical role in many applications such as video searching, advertising, summarization, etc. As a result, many effective works have been proposed and made significant progress [1], [2], [3], [4], [5].

Particularly, with the development of deep learning techniques, recent video classification methods follow a standard supervised framework and require a large number of training samples annotated for each video category to capture both inter and intra-class appearance variations. However, in many realistic scenarios, collecting and annotating sufficient video samples for each category is a bottleneck and consequently, limits the scalability of a fully supervised framework for ever-changing categories. Note that humans have a remarkable ability to recognize new categories without seeing any visual samples. A child would have no problem recognizing the action *playing violin* if she/he has seen violin before and known some related actions such as *playing guitar/piano*. In machine learning, this is known as *zero-shot learning* (ZSL), which provides a promising alternative to supervised learning and draws considerable attention in recent years.

As no labeled samples belonging to the unseen categories are available, the key to solve ZSL problem is to resort to some auxiliary information for describing how an unseen category is semantically related to the seen categories.[2] To guarantee the information is usable, it should also be related to the visual samples in the features space [6]. In ZSL, such information can help represent seen and unseen classes in a high dimensional space, called semantic space, where the knowledge from seen classes can be transferred to unseen classes. Most existing approaches on ZSL typically utilize either attribute [7], [8], [9], [10] or label-embedding [11], [12], [13], [14], [15] as auxiliary information to construct a semantic space. For attribute-based methods, a defined attribute ontology is utilized to describe various properties of the classes. Each class is represented by an attribute vector and termed as a *class prototype*. Although attribute-based methods are easy to

1. http://www.businessofapps.com/data/youtube-statistics/

- *J. Gao and C. Xu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the PengCheng Laboratory, Shenzhen 518066, China. E-mail: {junyu.gao, csxu}@nlpr.ia.ac.cn.*
- *T. Zhang is with the University of Science and Technology of China, Hefei, Anhui 230052, China. E-mail: tzzhang10@gmail.com.*

Manuscript received 16 July 2019; revised 14 Mar. 2020; accepted 23 Mar. 2020. Date of publication 14 Apr. 2020; date of current version 2 Sept. 2021.
(Corresponding author: Changsheng Xu.)
Recommended for acceptance by H. Ling.
Digital Object Identifier no. 10.1109/TPAMI.2020.2985708

2. We use the terms *class*, *category*, and *label* interchangeably.

understand and implement, the definition and annotation of attributes are labor-intensive and highly subjective, especially in video understanding. Therefore, these methods are hard to generalize to arbitrary unseen categories in a practical scenario [15], [16]. Recently, many approaches have shown that object semantics possess great potential in video understanding [17], [18], [19], [20]. Inspired by the strong relationships between objects and video categories, [17], [18], [19] employ objects as attributes for zero-shot video classification and achieve promising accuracy and generalization ability, where pre-trained object classifiers are used to recognize objects in videos. By using objects, the labor-intensive attribute annotation is avoided. However, most of them [17], [18] only consider the category-object relationships by using fixed similarity scores between word embedding vectors, which cannot get benefit from an end-to-end training. Although Wu *et al*. [19] adopt a multi-layer perceptron to learn the relationships between objects and seen categories, the relationships between objects and unseen categories are still represented by implicitly using the fixed similarity scores between seen and unseen categories. For label-embedding based methods, the class prototypes are obtained through the word vectors of class labels. Label-embedding based methods have the advantage of being learned from large-scale corpus with relatively less labor intensive, and the generated semantic information can be easily overlooked by humans [6], [13].

With the semantic information, however, it is still difficult for traditional ZSL methods to alleviate the large discrepancy between a class prototype and its class members [14], [21]. Note that most existing ZSL methods mainly focus on *static images*. When it comes to zero-shot video classification, current approaches suffer from more severe problems. The reasons are as follows. *(1) Relationship modeling*. ZSL method aims to transfer the learned knowledge from seen classes to unseen classes, which requires relationship modeling between these categories. For videos, the learning of relationships is more difficult than it in zero-shot object recognition since videos can cover a wide range of topics such as action, social events, scenes, and objects. Therefore, comprehensive relationship modeling between concepts is urgently required. However, for the current methods in zero-shot video classification, on the one hand, the attribute-based methods [8], [9], [10], [22] only leverage the category-attribute relationships to distinguish novel video categories. On the other hand, label-embedding based methods [13], [14], [15], [23] can only model category-category relationships in an implicit way and hardly get benefit from other information of videos. *(2) Heterogeneity gap [24]*. Since the category representations are typically obtained from either textual corpus (i.e., label-embedding) or human definition (i.e., attributes), there will be a large inconsistency between visual features and the semantic information. This inconsistency, called heterogeneity gap, is more serious in videos due to the multifarious video context with various poses and appearances. As a result, the gap will lead to an information loss such that seen-to-unseen correlation would degrade [24]. *(3) Domain shift*. Video data contains richer semantics and more noises than images, thus there exist a large domain gap between seen and unseen videos. Traditional ZSL methods [8], [13] often learn a projection for visual features from seen classes and directly use it in unseen test samples with no adaptation.

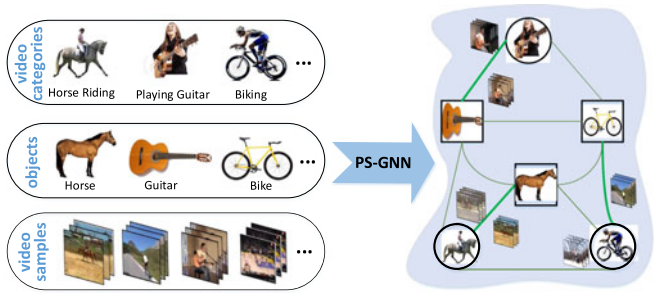


Fig. 1. The motivation of our PS-GNN. We use object semantics as attributes to bridge both video categories and samples. To model the relationships between category-attribute, category-category, and attribute-attribute, we build a concept graph to perform message propagation. Moreover, video representations are learned to be close to their belonging categories. In the right of this figure, the thinker green lines signify stronger relationships.

Obviously, the fixed knowledge transfer from the seen domain is insufficient to recognize novel classes. *(4) Temporal modeling*. Compared with static images, videos contain both spatial and temporal information. In fact, significant advantages of exploiting temporal rather than only spatial information for video understanding have been shown recently [25], [26]. Nevertheless, most existing ZSL methods ignore to leverage the temporal cue in the learning process.

Motivated by the above observations, in this work, we propose a novel framework to directly and adaptively model the relationships between all the concepts[3] for zero-shot video classification, which can jointly enjoy the merits of alleviated heterogeneity gap, low domain shift, and robust temporal modeling. Specifically, as shown in Fig. 1, we also use object semantics as attributes to bridge both video categories and samples. To model *all the relationships* between category-attribute, category-category, and attribute-attribute, we formulate the ZSL task as a learning problem on a concept graph, where nodes are associated with the concepts, and edges are generated by external human-knowledge information (e.g., knowledge graphs) and trainable similarity kernels. To effectively model the dependencies and propagate messages between different nodes, we resort to Graph Neural Networks (GNNs) [27], [28], [29], which have received increasing attention and successfully been applied in various computer vision tasks [2], [30], [31], [32], [33]. Based on the concept graph, we propose a novel framework, named Prototype-Sample GNN (PS-GNN), consisting of a prototype branch and a sample branch, where knowledge learned from seen classes can be effectively transferred to unseen videos in an end-to-end manner. The prototype branch aims to learn prototypes for different video categories, which takes as input a set of word embedding vectors corresponding to the concepts. The sample branch is designed to generate attribute-feature of a video sample by leveraging its object scores and the word-embedding vectors. The whole framework is optimized via a multi-task loss including a classification loss and a prototype center loss, where the video representations are forced to be close to the class prototypes. The architecture of our framework is shown in Fig. 2, and the details will be demonstrated in Section 3. Interestingly, both prototypes and the video samples are represented by the word embedding and

3. Here, the concepts include video categories and attributes (objects).

object semantics in a unified framework, where the multi-task loss is adopted to associate videos and categories together. As a result, the *heterogeneity gap* is alleviated. For different video input, the edge weights in the proposed PS-GNN are learned adaptively with the guidance of universal human-knowledge, which lowers the *domain shift* problem. In addition, to conduct robust *temporal modeling*, we incorporate a temporal-relation attention module into the sample branch to leverage the dynamically changing object scores over time.

To sum up, the main contributions of this work are:

- Starting from the definition, a novel PS-GNN is developed for video ZSL, which directly and adaptively models all the relationships between category-attribute, category-category, and attribute-attribute. The successful attempt signifies that considering relationships with graph-based end-to-end learning is effective, which offers an alternative orientation for ZSL.
- By carefully incorporating the object semantics, multi-task loss function, external knowledge, adaptive graph learning, and temporal-relation attention module, our proposed method can jointly enjoy the merits of alleviated heterogeneity gap, low domain shift, and robust temporal modeling for zero-shot video classification.
- In the PS-GNN framework, the prototype branch and the sample branch can co-adapt and cooperate to achieve robust ZSL in a unified framework. Here, better class prototypes can make the learned video features more discriminative, and better video features can lead to more representative prototypes. Moreover, we conduct extensive experiments on five popular benchmarks, demonstrating the effectiveness of our framework.

It should be noted that a preliminary version of this work was published in AAAI 2019 as an oral presentation [34]. In this work, we further extend our previous paper from both theoretical and empirical aspects. Theoretically, an adaptive graph learning strategy is developed, which aims to perform customized relationship modeling for different video input and thus alleviates the domain shift problem. In addition, a prototype center loss is designed to align the video samples and their corresponding class prototypes, thereby the heterogeneity gap is lowered and the intrinsic structure between videos and categories are preserved. Furthermore, a robust temporal-relation attention module is exploited while the temporal modeling strategy in [34] is the special case of this paper. Empirically, we comprehensively evaluate the performance of PS-GNN by making comparisons with more recent methods on 5 benchmarks, presenting more analysing results and visualization results with different scenarios, and so on. We also extend our method in a few-shot learning setting.

2 Related Work

Zero-shot video classification is a promising research direction and some efforts have been achieved in recent years [8], [17], [18], [19], [23], [24]. In this section, we briefly review four problems related to our work, including zero-shot learning, video classification by deep learning, zero-shot video classification, and graph neural networks.

2.1 Zero-Shot Learning

The goal of Zero-Shot Learning (ZSL) [7] is to generalize existing knowledge to classify new categories without training samples. We refer readers to the excellent surveys [35], [36] on ZSL for more information. Current ZSL approaches generally fall into three categories: attribute-based methods, embedding-based methods, and hybrid methods. Early approaches of ZSL follow an intuitive idea that first learn different attribute classifiers and then recognize a visual pattern by comparing its predicted attributes with descriptions of unseen classes. The pioneering work, Direct Attribute Prediction (DAP) [7] infers the posterior of each attribute, and then the class posteriors are calculated by maximizing a posterior. Whilst the Indirect Attribute Prediction (IAP) [7] computes the attribute posteriors from the class posterior of seen categories. Embedding-based approaches aim to construct a shared embedding space for training samples and their semantic features. The SJE model [37] utilizes several compatibility functions linearly to build a joint embedding space. The ESZSL model [11] adds a Frobenius norm regularizer to learn an embedding space. Xian *et al.* [38] utilize a generative adversarial network to synthesize CNN features conditioned on class-level semantic information. Hybrid methods focus on using the combination of seen categories to help recognize unseen samples. The ConSE model [39] jointly uses the classification probabilities of seen categories to classify unseen objects with a convex method. Recently, Fu *et al.* [40] propose a class prototype graph where a manifold distance is designed to improve the generalization ability. However, it does not jointly model the relationships among categories and attributes in an end-to-end learning fashion. Currently, much of the research focuses on static image recognition. When coming to videos, these methods suffer from more severe challenges. As a result, robust video ZSL methods are required to handle the particular properties of videos. In this paper, we propose to learn comprehensive relationships among categories and attributes in a two-stream graph neural network.

2.2 Video Classification by Deep Learning

With the development of deep learning techniques, video classification has entered a brand new era. Since deep neural networks can naturally get benefit from the end-to-end training, the process of learning classifiers and features can co-adapt and co-operate. One direction of video classification is using traditional Convolutional Neural Networks (CNNs) for model training. The pioneering work of Karpathy *et al.* [41] extend 2D CNNs into video classification with different architectures. Simonyan and Zisserman [42] design a CNN-based two-stream framework that combines spatial and temporal information. Another direction is resorting to 3D CNN architectures for video classification, such as C3D [25] and P3D ResNet [43]. The two-stream Inflated 3D ConvNet (I3D) [5] is proposed to learn seamless spatio-temporal fetures. For RNNs, Srivastava *et al.* [44] adopt an encoder-decoder architecture and use LSTM networks to learn video representations. Moreover, various methods have been proposed in attention-based frameworks. Wang *et al.* [4] propose a non-local operator that computes the response at a position as a weighted sum of all positions. Girdhar *et al.* [45] incorporate an attention module in action recognition, which can be

trained with or without supervision. A recent work [46] involves both a slow pathway and a fast pathway operating at low/high frame rates. Although the above methods achieve significant performance, they require large-scale training data, thus suffer from the scalability problem. In this work, we propose a novel PS-GNN that leverages both relationship learning and object semantics in a unified framework, which achieves favorable zero-shot video classification performance.

2.3 Zero-Shot Video Classification

Recently, zero-shot video classification has drawn considerable attention. One main branch is recognizing human actions in the zero-shot setting. The early work [8] takes a set of human-defined attributes into consideration to describe the intrinsic information of the action in a video. Gan *et al.* [47] treat each video category as a domain, and tackle attribute detection as a multi-source domain generalization process. Although simple and effective, the manually-specified attributes are labor-intensive and subjective to annotate [16]. To address these problems, label embeddings have shown great potential in recent years since only category names are needed. Xu *et al.* [13], [14] utilize word embeddings to build a shared semantic space for mapping both class names and video features. Qin *et al.* [15] utilize error-correcting output codes by considering both intrinsic data structures and category-level semantics. However, only using word embeddings is far from adequate to distinguish different video categories because of the heterogeneity gap. Recently, some methods show that object semantics have great potential for understanding videos. The pioneer work Objects2action [17] constructs a semantic embedding space by using thousands of object categories. With the help of pre-trained object detectors, such a method [18] can even perform zero-shot spatial-temporal action localization. Although simple and effective, these methods only alleviate the heterogeneity gap to some extent since they do not jointly learn the class prototypes and video representations in an end-to-end framework. There are also other alternatives to tackle Zero-Shot Action Recognition (ZSAR). Some approaches use the semantic relationships between categories such as inter-class relationship [48] and pairwise relationship [49]. In addition to action recognition, a number of methods apply ZSL to arbitrary types of videos, such as social events and scenes. Wu *et al.* [19] propose an object-scene semantic fusion network for large-scale zero-shot video classification. Zhang *et al.* [24] design a GAN-based visual data synthesis framework to generate new data and alleviate the heterogeneity gap. Despite the competitive successes in zero-shot video classification, these traditional methods may ignore the comprehensive relationships modeling between concepts (e.g., video categories, and attributes) in an end-to-end fashion. In fact, all these relationships can contribute to the ZSL task either in an explicit or implicit way. We design an end-to-end PS-GNN framework, which can directly and adaptively models all the relationships between category-attribute, category-category, and attribute-attribute.

2.4 Graph Neural Networks (GNNs)

GNNs are designed to utilize deep learning techniques on arbitrary graph-structured data, which is, in fact, natural generalizations of neural networks to non-euclidean spaces. Readers could refer to the excellent surveys [50], [51] for more information. Generally, there are two ways to develop graph neural networks. On the one hand, some methods apply feed-forward networks directly to graph nodes and their neighbors in the spatial domain [30], [52], [53]. On the other hand, a number of well-defined localized operators is considered in the form of spectral analysis with the convolution theorem [27], [28], [54]. By using the localized spectral filters, these approaches can easily find local neighborhoods of nodes with low computational cost [27], [28]. Until now, graph neural networks achieve successful performance in many computer vision tasks. Marino *et al.* [30] design a graph search neural network (GSNN), which can operate on knowledge graphs in an end-to-end framework for image classification. Lee *et al.* [31] utilize a graph gated neural network to consider the relationships between concepts for multi-label image classification. Wang *et al.* [2] propose to interpret videos as space-time region graphs that model similarity relationships and spatial-temporal relationships. For human skeleton-based action recognition, Yan *et al.* [32] design a spatial-temporal graph convolutional network on dynamic skeletons. For person re-identification, Shen *et al.* [55] use graph neural networks to model probe-gallery relationships in an end-to-end manner. Gao *et al.* [33] utilize a graph ConvLSTM method to model the dynamic knowledge evolution for supervised video classification. In this paper, we advance two-stream graph neural networks for zero-shot video classification by taking both relationship modeling and object semantics into consideration.

3 PS-GNN for Video ZSL

Zero-shot video classification, technically, can be interpreted as a process of jointly learning class prototypes and video features such that the videos are close to their belonging class prototypes. As no labeled samples of the unseen categories are available, the key to solving this problem is to model the relationships between seen and unseen categories. Different from existing ZSL approaches which do not consider the explicit relationships with external knowledge priors [13], [16] or only model category-category relationships [56], as shown in Fig. 2, we propose an end-to-end knowledge-based framework PS-GNN consisting of two GNN branches: Prototype branch (P-branch) and Sample branch (S-branch). The P-branch takes the word embeddings of all the concepts as input and generates the prototypes for video categories. The S-branch combines the object scores of a video sample and the corresponding word embeddings of objects to produce the final video features. During the training phase, the class prototypes of seen categories are learned via a supervised process while the unseen class prototypes can be generalized by the graph neural model. At the test phase, the trained PS-GNN is used to find the nearest category prototypes for a test video. The details are shown in the following subsections.

3.1 Problem Formulation

Assume there is a training set including N_s labeled video samples $\mathcal{D}^s = \{V^s, Y^s\}$ with S seen classes Y^s, where each video $\mathbf{v}^s \in V^s$ belongs to a label $y^s \in Y^s$. Similarly, we have

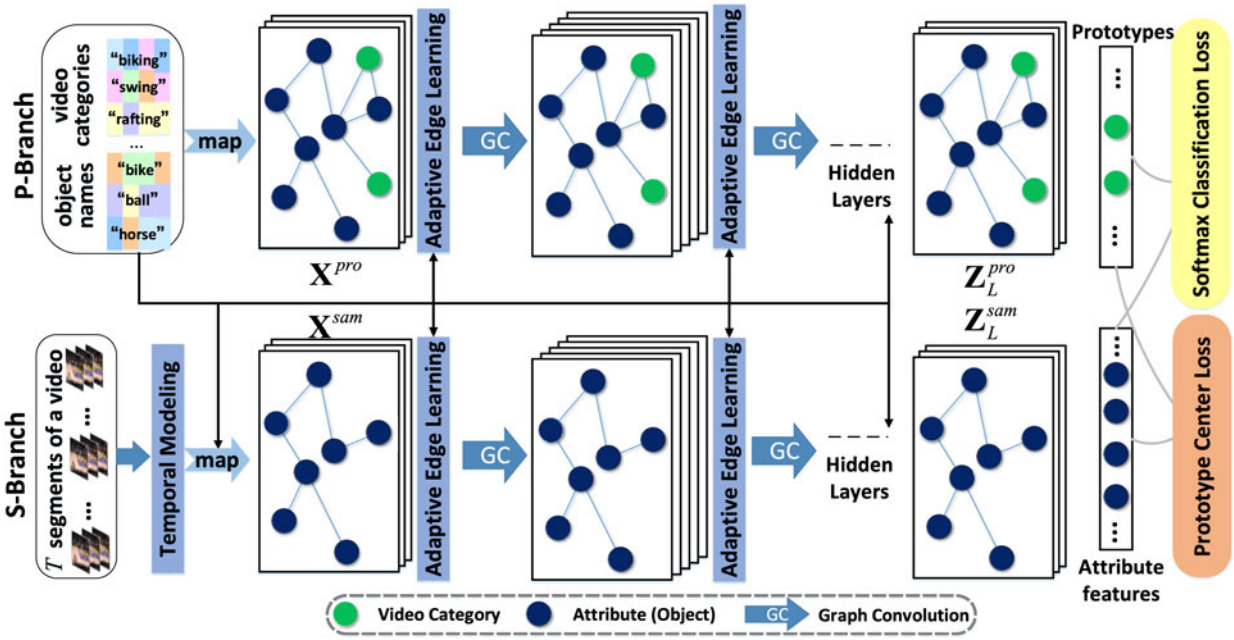


Fig. 2. The pipeline of PS-GNN. Specifically, PS-GNN consists of two branches, i.e., prototype branch (P-branch) and sample branch (S-branch). The P-branch takes as input a set of word vectors of all the concepts and learns the prototypes of video categories. The S-branch generates attribute features of a video sample by leveraging its object scores and the corresponding word vectors. In the training phase, the framework is optimized by a supervised loss and a prototype center loss. With the message propagation in this graphic model, prototypes of unseen categories can be generalized. At the test phase, the nearest prototype can be found for a test video.

a test set $\mathcal{D}^u = \{V^u, Y^u\}$ containing N_u videos from U unseen classes Y^u. Here, $V^s \cup V^u = V, Y^s \cup Y^u = Y$, the labels from sets $\mathcal{D}^s$ and $\mathcal{D}^u$ are disjoint, $Y^s \cap Y^u = \emptyset$. Moreover, we have an object set O with N_o object categories, which serves as a bridge to connect videos and class labels. The goal of the video ZSL task is to learn an objective function $\min \mathcal{L}(V^s, Y, O)$ that can generalize to V^u.

3.2 GNNs: Preliminary

As operators for graph-structured data, GNNs can provide a powerful balance between sample complexity and expressivity. Different from traditional neural networks that only can be applied on regular grids, graph neural networks allow us to calculate the response of a node based on its neighbors defined by the graph structure. In this work, we adopt Graph Convolutional Network (GCN) proposed in [28] as a base building block. To keep this paper self-contained, we briefly introduce GCN as follows: Given an undirected graph with m nodes, a set of edges between nodes, an adjacency matrix $\mathbf{A} \in \mathrm{R}^{m \times m}$, we consider a linear operator of a GCN layer as the multiplication of a graph signal $\mathbf{X} \in \mathrm{R}^{k \times m}$ (the column vector $\mathbf{X}_i \in \mathrm{R}^k$ is the feature representation at the ith node) with a filter $\mathbf{W} \in \mathrm{R}^{k \times c}$

$$\mathbf{Z} = (\mathbf{A} + \mathbf{I})\mathbf{X}^\top \mathbf{W} + \mathbf{B}, \tag{1}$$

where $\mathbf{I}$ is the identity matrix, indicating the self-loops. As a result, the input and output of a GCN layer are $k \times m$ and $c \times m$ matrices, respectively. $\mathbf{B} \in \mathrm{R}^{c \times m}$ is a c dimensional bias for each node.

To achieve higher adaptive ability, we can further improve Eq. (1) by learning adaptive edge representations $\widetilde{\mathbf{A}}$ from the current node features

$$\widetilde{\mathbf{A}}_{ij} = \phi_\theta(\mathbf{X}_i, \mathbf{X}_j), \tag{2}$$

where ϕ is a symmetric function parameterized by θ, e.g., a Multi-Layer Perceptron (MLP). With Eq. (2), we can reformulate the GCN layer by using the learned edge features

$$\mathbf{Z} = (\widetilde{\mathbf{A}} + \mathbf{I})\mathbf{X}^\top \mathbf{W} + \mathbf{B}. \tag{3}$$

Note that a GNN can be built by stacking multiple graph neural layers of the form of Eq. (3), each layer followed by a non-linear activation function (such as leaky ReLU [57]).

3.3 Constructing the Concept Graph

How to organize the video and object categories as a structured graph is not trivial since the relationships between every two concepts are unknown. Recently, Knowledge Graphs (KGs) have been successfully applied in a series of computer vision fields such as multi-label image classification [30], object detection [58], zero-shot image recognition [31], [56], etc. The promising results show that KGs do have a remarkable ability to bridge the knowledge gap in existing methods. As a result, great potential is expected to leverage knowledge graphs for zero-shot video classification. Since we have $S + U + O$ concepts (seen classes, unseen classes, objects) associated with all the videos, we utilize the off-the-shelf KG, ConceptNet 5.5 [59], to construct our concept graph with the same number of nodes corresponding to these concepts. We adopt the terms *concept* and *node* interchangeably hereafter. The graph structure is denoted as an adjacency matrix, $\mathbf{A}$. ConceptNet 5.5 is constructed from many other knowledge bases including WordNet [60], DBpedia [61], *et al.* Similar to previous work [58], we adopt its English subgraph with about 1.5 million nodes for knowledge extraction. We

employ string matching to map the concepts (video categories, objects) to the nodes in ConceptNet. Note that a few concepts have no corresponding nodes due to their rare appearance, we replace the name of these concepts with common words that can be found in ConceptNet. For instance, "skijet" is replaced by "jetski" without losing the substantial semantic information. In addition, we follow [34] to find the edge weights $\mathbf{A}_{ij}$ between two nodes in ConceptNet. Specifically, each edge in ConceptNet is an assertion—indicates a combination of various data sources to produce that edge [59]. While ConceptNet has multiple types of edges, we follow some previous methods [30], [58] to simplify our constructed concept graph as a single matrix (adjacency matrix), which can effectively propagate information between nodes and model the semantic consistency.

3.4 PS-GNN Architecture

3.4.1 Prototype Branch

The prototype branch consists of L GNN layers. The lth layer takes as input the feature representation ($\mathbf{Z}^{pro}_{l-1}$) produced from the previous layer $l-1$ and generates an output feature matrix $\mathbf{Z}^{pro}_{l}$. As shown in Fig. 2, the input to this branch is a $k \times (S+U+O)$ matrix $\mathbf{X}^{pro}$ that is the word vectors of all the concepts. Here, k indicates the dimensionality of the word vectors. The output of the final layer L is a $d_1 \times (S+U+O)$ matrix $\mathbf{Z}^{pro}_{L}$ where d_1 is the dimensionality of the prototypes. In particular, the S seen category prototypes $\mathbf{Z}^{pro}_{L,1:S}$ (corresponding to Y^s) are optimized using the training data. During the training phase, another U unseen category prototypes can be generalized from these seen ones and the learned object prototypes via the GNN. Note that the O object prototypes serve as a bridge between seen and unseen action categories, which will not be explicitly used in the training/inference phase.

3.4.2 Sample Branch

The sample branch, also with L GNN layers, aims to generate the video representations for measuring their similarities with the category prototypes. Note that this branch only considers the object concepts as graph nodes since we can only get the object (attribute) information of a given video beforehand. To obtain object scores of a video, we follow [18] to utilize a GoogLeNet model [62], trained on a 12,988-class shuffle [63]. Similar to [17], the top K most relevant objects for each video category are selected, which results in O objects from the initial 12,988 object classes.

Temporal Modeling. Since temporal modeling plays an important role in video understanding [1], the sample branch employs the temporal information by using an attention module. Particularly, for a video $\mathbf{v}$,[4] we segment it into T equal-length segments $\{\mathbf{v}_t\}_{t=1}^{T}$. The number of segments is fixed for all video samples such that the sequential parallelization can be achieved in our framework. For the frames in segment $\mathbf{v}_t$, we average the output at the softmax layer of the GoogleNet model and obtain the object scores $\mathbf{v}_t \mapsto \bar{\mathbf{v}}^O_t \in \mathrm{R}^{No}$. Based on the object scores, we compute the semantic representation of each video segment using $\bar{\mathbf{v}}^O_t$ and the corresponding word vectors

$$\mathbf{x}^v_t = \frac{1}{\sum_{o \in \mathbf{K}(O,\mathbf{v}_t)} \bar{\mathbf{v}}^O_{t,o}} \sum_{o \in \mathbf{K}(O,\mathbf{v}_t)} \bar{\mathbf{v}}^O_{t,o} \mathbf{e}_o, \tag{4}$$

where $\mathbf{K}(O,\mathbf{v}_t)$ is the top-K scored objects in the video segment $\mathbf{v}_t$. $\mathbf{e}_o$ is the word-embedding vector of the oth object. To comprehensively consider the dynamic information of this video, a temporal-relation attention operator is conducted as follows:

$$\begin{aligned} \mathbf{G}^v_{s,t} &= \frac{\Big(\mathrm{ReLU}(f(\mathbf{x}^v_t)^\top g(\mathbf{x}^v_s))\Big)^2}{\sum_{t=1}^{T}\Big(\mathrm{ReLU}(f(\mathbf{x}^v_t)^\top g(\mathbf{x}^v_s))\Big)^2} \\ \mathbf{r}^v_s &= r\left(\gamma \sum_{t=1}^{T} \mathbf{G}^v_{s,t} h(\mathbf{x}^v_t) + \mathbf{x}^v_s\right), \end{aligned} \tag{5}$$

where $f(\cdot), g(\cdot)$ are two fully-connected layers with the output dimension of d_2. $h(\cdot)$ is a fully-connected layer with the output dimension of k. $\mathbf{G}^v_{s,t}$ is the learned attention weight that denotes the contribution of segment t to the representation of segment s. $r(\cdot)$ is a fully-connected layer with a *tanh* activation function, which produces residual object scores $\mathbf{r}^v_s \in \mathrm{R}^{No}$ for each segment. By combing the residual and initial objects scores ($\mathbf{o}^v_t$ and $\mathbf{r}^v_t$), we can initialize the node features in the sample branch as follows:

$$\mathbf{X}^{sam}_o = \frac{1}{T}\sum_{t=1}^{T} \mathrm{ReLU}\Big(\mathbf{o}^v_{t,o} + \mathbf{r}^v_{t,o}\Big)\mathbf{e}_o, \tag{6}$$

where $\mathbf{X}^{sam}_o$ denotes the initial oth node feature. The ReLU function ensures the refined object scores are positive. The output of this branch is a feature matrix $\mathbf{Z}^{sam}_L \in \mathrm{R}^{O \times d_1}$. d_1 is the same as the dimensionality of the class prototypes produced by the prototype branch.

Discussion. Compared to the initial version in [34], the proposed temporal modeling module has several advantages: (1) Instead of using the raw object scores, we adopt the semantic representation (Eq. (4)) of video segments to calculate the attention weights, which leverages the semantic word embedding in the learning process. (2) The residual learning of object scores can adaptively refine the noises caused by the pre-trained object classifier. (3) The exponential activation in the softmax function is replaced by a squared ReLU operation (Eq. (5)), resulting in sparse attention weights and stable training [64]. In fact, the temporal modeling in [34] is a special case of the proposed method without the semantic representation, residual learning, and square ReLU operation.

3.5 Adaptive Edge Learning

Although we can directly use the weight in a knowledge graph to denote the relationship between two graph nodes in our PS-GNN, the fixed weight might limit the practical scenarios of video classification. For example, the most relevant objects to the video category *birthday party* are *cake* and *candle*. However, some video samples in this category do not have cakes or candles but only include *crowns* and *balloons*. In this situation, a refined graph is more attractive, where the edge weights between the birthday party node and cake/candle nodes are lower than it with the crown/balloon nodes. To effectively propagate messages between

4. For simplicity, we omit the script of $\mathbf{v}$ here.

different nodes in the concept graph, it is desirable to learn adaptive edge features by using the video representations and concept representations. In our work, we consider an MLP stacked after the combination of two nodes. For every different two nodes which are connected in the constructed knowledge graph, the adaptive edge learning is performed

$$\mathbf{R}_{ij} = \phi_{\theta_2}(\hat{\mathbf{x}}_i || \hat{\mathbf{x}}_j), \text{where } \hat{\mathbf{x}}_i = \phi_{\theta_1}(\mathbf{x}^v || \mathbf{e}_i), \tag{7}$$

where $\mathbf{x}^v = \text{MaxPooling}_T([\mathbf{x}_1^v, \ldots, \mathbf{x}_t^v, \ldots, \mathbf{x}_T^v])$ is the fused video representation, MaxPooling_T is a max pooling operator applied with a time range T. $||$ denotes feature concatenation. ϕ_{θ_1} is a fully-connected layer that jointly utilizes the video $\mathbf{x}^v$ and the word embedding of the ith concept to produce a d_3-dimensional feature $\hat{\mathbf{x}}_i$. As a result, $\hat{\mathbf{x}}_i$ can encodes the specific video information and the generic concept semantics to comprehensively learn the adaptive edges. For implementing ϕ_{θ_2}, we utilize a 2-layer MLP with the output dimension of d_4 and 1. The learned weights are then normalized to a probability distribution along each row, by using a ReLU-based softmax like Eq. (5). With the learned edges, we re-write the GNN layer (Eq. (3)) as

$$\mathbf{Z} = (\mathbf{R} + \widetilde{\mathbf{A}} + \mathbf{I})\mathbf{X}^\top \mathbf{W} + \mathbf{B}. \tag{8}$$

As a remark, with the adaptively learned edge weights, the dependency between different nodes can be modeled accordingly. In addition, our GNN layer also follows a residual learning process: we have the knowledge-based adjacency matrix $\widetilde{\mathbf{A}}$ derived from knowledge graphs, which serves as a base graph prior in our framework. Meanwhile, the learned adjacency matrix $\mathbf{R}$ serves as a residual graph refinement for effective information propagation. Note that we share the learned edge weights between both branches because edges only represent relationships between concepts. The relationships should be consistent in the prototype branch and sample branch. To save computational cost, for a given video, we only generate the adjacency matrix once using Eq. (7) and then apply it for all the GNN layers.

3.6 Learning and Inference

Loss Function. For the S seen classes, the cross-entropy loss is evaluated over all the labeled video samples. In addition, to force the video features to be close to their belonging category prototypes, we design a prototype center loss to minimize the intra-class variations. Therefore, the total training loss is

$$\begin{aligned}\mathcal{L} =& \frac{1}{N_S}\sum_{i=1}^{N_S}\sum_{j=1}^{S}\mathbf{y}_i^j\Big[-\gamma \log(\mathbf{p}_i^j) + (1-\gamma)||\hat{\mathbf{z}}_{i,j} - \mathbf{Z}_{L,j}^{pro}||_2\Big] \\ &+ \lambda||\theta||_2^2, \text{ where } \hat{\mathbf{z}}_{i,j} = \frac{1}{|\mathcal{N}(j)|}\sum_{o\in\mathcal{N}(j)}\mathbf{Z}_{L,i,o}^{sam},\end{aligned} \tag{9}$$

where $\mathbf{Z}_{L,j}^{pro}$ denotes the final representation of the jth category prototype, outputted from the last GNN layer of the P-branch. $\mathbf{Z}_{L,i,o}^{sam}$ is the oth object node feature generated by the S-branch (in layer L) with respect to the ith video. $\mathbf{y}_i^j$ is the ground-truth label (0 or 1) of the ith video with respect to the jth seen category. $\mathbf{p}_i^j$ is the predicted score with a softmax operation: $\mathbf{p}_i = \text{softmax}(\mathbf{q}_i)$, where $\mathbf{q}_i^j = -||\hat{\mathbf{z}}_{i,j} - \mathbf{Z}_{L,j}^{pro}||_2$. θ indicates all the learnable weights in our model. γ is a balance factor between the cross entropy loss and the prototype center loss. λ is the coefficient of the weight decay term. $\mathcal{N}(j)$ is the one-hop object neighbors of the jth video categories in the knowledge graph, which means we focus on strongly-related objects for classifying a specific video. Empirically, we find that using neighbors results in faster convergence speed and higher performance than using all the object features. The reason is that, with the message-passing of GNNs, the useful information is propagated and augmented to the category neighbors $\mathcal{N}(i)$ via the optimization of the framework. In addition, only using neighbors can avoid some distraction in video classification.

Inference. During training, we are able to not only produce the prototypes of the S seen categories but also generalize to unseen categories via the PS-GNN model. The proposed model also learns to adaptively generate sample features with regard to different input videos. At test phase, we use the generated object features of the ith test video (from S-branch) to perform nearest neighbor search on the prototypes of unseen categories (from P-branch)

$$y_i := \arg\min_j ||\hat{\mathbf{z}}_{i,j} - \mathbf{Z}_{L,j}^{pro}||_2, j = S+1, \ldots, S+U. \tag{10}$$

Note that the above equation can be easily extended to generalized ZSL setting where videos from seen categories are also involved for testing.

4 Experiments

In this section, we evaluate the performance of the proposed PS-GNN method on two related tasks: zero-shot action recognition (ZSAR) and zero-shot classification of unconstrained videos. For the former task, we utilize three widely-used video datasets: Olympic Sports [65], HMDB51 [66] and UCF101 [67]. For the latter one, we adopt the popular dataset Columbia Consumer Video (CCV) [68] and the large-scale video dataset FCVID [69] for evaluation. Further, numerous ablation experiments are also performed to comprehensively and systematically analyze the proposed method. Specifically, the core code of PS-GNN will be updated at https://github.com/junyuGao/Zero-Shot-Action-Recognition-with-Two-Stream-GCN.

4.1 Implementation Details

For the word embedding method, we follow [18] to employ the skip-gram network trained on the metadata of the images and videos from the YFCC100M dataset [74]. This model produces a 500-dimensional representation for each word. To represent each concept with a fixed length, we simply average all the word vectors [18]. Both branches in our framework are composed of 2 graph neural layers with output channel dimensionality of 1024 and 512. All the layers in our framework, except for special instruction, are applied with the Leaky ReLU activation function. We perform L2-Normalization on the produced prototypes to regularize them into similar magnitudes. For the attention module, the number of segments T is set to 16 by grid search over {8, 16, 32}. The number of selected objects for each video category is set to $K = 100$. d_1, d_2, d_3, and d_4 are set to 512, 256, 512, and 256, respectively. To train our whole model, we use the ADAM [75] optimizer with learning rate 0.0001 and weight

decay 0.0005. The model is trained for 5 epochs with a batch size of 48. We implement our framework by Tensorflow.

4.2 Zero-Shot Action Recognition

4.2.1 Datasets and Splits

We perform experiments on three popular action recognition benchmarks Olympic Sports, HMDB51 and UCF101 containing 783, 6766 and 13320 videos with 16, 51, and 101 categories, respectively. For evaluation, we follow the 50/50 data splits proposed by [14], i.e., videos of 50 percent categories are employed for model learning and the other 50 percent categories are held unseen for testing. We adopt the 50 independent splits generated by [14] and report the average accuracy and standard deviation for comparison.

4.2.2 Zero-Shot Settings

Commonly, there exist two types of zero-shot settings including inductive setting and transductive setting. The former assumes that only the labeled videos from the seen categories are available during training while the latter can utilize the unlabeled samples of the unseen categories. In our experiments, for the transductive setting, we first pick out the top 2000 frequent objects in all videos, then choose the top K most relevant objects for each video class from this object set. Moreover, different from traditional zero-shot settings where the seen categories are not utilized at the test phase, the recent generalized setting adopts both seen and unseen video samples as test data. Following [14], [76], we employ the generalized setting in a transductive way. In this setting, we follow [76] to add an additional bias loss to alleviate the bias towards seen categories.

4.2.3 Compared Methods

Compared Methods. We compare our method with several state-of-the-art methods. (1) Direct/Indirect Attribute Prediction method (DAP/IAP) [7]. DAP/IAP trains a series of SVM classifiers independently for each attribute and utilizes a probabilistic method to map attribute predictions to target classes. (2) Human Actions by Attributes (HAA) model [8], which is implemented by [13]. HAA first trains attribute detection SVMs, then evaluates samples based on their predicted attribute scores and the attribute vectors of target classes. (3) Self-training method with SVM and label Embedding (SVE) [12]. It learns the visual-semantic mapping by support vector regression. (4) Embarrassingly Simple Zero-Shot Learning (ESZSL) [11], which uses the mean square loss on label predictions instead of the common regression loss. (5) Structured Joint Embedding (SJE) [37] uses a triplet hinge loss to enforce relevant labels having higher predicted values than those of non-relevant labels. (6) Unsupervised Domain Adaptation (UDA) [70]. UDA learns a dictionary on auxiliary data and adapts it to the target domain as a constraint on the target dictionary. (7) Convex Semantic Embeddings (ConSE) [39] first trains classifiers for each seen category and uses the combination of seen category embeddings to predict unseen videos. (8) Propagated Semantic Transfer (PST) [72] uses label propagation to adjust the initial predictions of DAP. (9) Multi-Task Embedding (MTE) [13] designs a visual-semantic mapping with generalization properties and a dynamic data re-weighting model to pay different attention to auxiliary data. (10) Objects2Action [17], which also utilizes objects as attributes for ZSAR. In the method, video labels are assigned to an unseen video sample based on a convex combination of label and object similarities. (11) Manifold Regularized ridge regression (MR) [14] employs a manifold regularization for a regressor based on unseen samples. (12) Zero-Shot with Error-Correcting Output Codes (ZSECOC) [15] learns ECOC for video categories from both category-level semantics and intrinsic data structures. (13) Universal Representation (UR) model [16] addresses ZSAR as a generalized multiple-instance learning problem to preserve visual and semantic components in a shared space. (14) Generative Approach (GA) [73] adopts a generative approach to synthesize unseen class data and utilizes unlabelled C3D feature from unseen classes to rectify the biases. (15) Visual Data Synthesis (VDS) [24] leverages both semantic knowledge and visual distribution to synthesize visual features of unseen categories via Generative Adversarial Networks (GANs). (16) Out-of-distribution (OD) detector [71] which uses conditional Wasserstein GAN to synthesize features from unseen action classes.

4.2.4 Quantitative Results

In this section, a comparison of our PS-GNN against other approaches for the tasks of inductive/transductive ZSL and generalized ZSL in action recognition is given.

Inductive Setting. For the inductive setting, as shown in Table 1, our proposed method performs favorably against state-of-the-art methods. In this table, the best results for each dataset are shown in boldface. The standard deviation from the mean is also reported. Compared with the recent approaches UR, VDS, and OD, PS-GNN achieves an absolute gain of $(18.6\%, 10.7\%, 9.2\%)$ on the UCF101 benchmark. For the HMDB51 and Olympic Sports datasets, we also get better or comparable results against the state-of-the-art approaches. Furthermore, several tendencies can be observed from this table. First, most methods achieve higher performance on Olympic Sports dataset than other benchmarks. This is because that the dataset only has 8 classes held unseen for model evaluation. If we perform random guess for labeling unseen videos, the accuracies on the three datasets are 12.5, 4, and 2 percent respectively. Second, most recent approaches are based on semantic label embeddings, which demonstrates the learned embeddings from large language corpus can generate semantic information that is overlooked by humans [6]. As a result, our method also uses word vectors as label embeddings. Third, from the results on UCF101 dataset, we can observe that the object-based methods obtain much better performance than the visual feature-based approaches. The reason is that the video categories of UCF101 are highly related to objects, which verify the effectiveness of using objects as attributes. Fourth, compared with another object-based method, Objects2Action, our proposed method outperforms it by $(6.1\%, 10.3\%)$ on UCF101 and HMDB51, respectively. Note that our PS-GNN and Objects2Action get better accuracy on UCF101 than HMDB51 while other visual feature-based methods (e.g., MTE, and UR) have the opposite results. The reason is that many video categories in HMDB51 are not very sensitive to objects, such as the actions *run*, *walk*, *sit*, and *stand*. As a result, we can further improve the performance of PS-GNN by adding visual features into the

TABLE 1
Zero-Shot Video Classification Performance on Four Benchmarks Compared With State-of-the-Art Methods

Method	Reference	Feature	Label Embedding	ID/TD	Olympic Sports	HMDB51	UCF101	CCV
DAP	CVPR2009 [7]	FV	A	ID	45.4 ± 12.8	N/A	15.9 ± 1.2	N/A
IAP	CVPR2009 [7]	FV	A	ID	42.3 ± 12.5	N/A	16.7 ± 1.1	N/A
HAA	CVPR2011 [8]	FV	A	ID	46.1 ± 12.4	N/A	14.9 ± 0.8	N/A
SVE	ICIP2015 [12]	BoW	W	ID	N/A	13.0 ± 2.7	$10.9 \pm \pm 1.5$	N/A
ESZSL	ICML2015 [11]	FV	W	ID	39.6 ± 9.6	18.5 ± 2.0	15.0 ± 1.3	N/A
SJE	CVPR2015 [37]	FV	W	ID	28.6 ± 4.9	13.3 ± 2.4	9.9 ± 1.4	16.3±3.1
SJE	CVPR2015 [37]	FV	A	ID	47.5 ± 14.8	N/A	12.0 ± 1.2	N/A
UDA	ICCV2015 [70]	FV	A+W	ID	N/A	N/A	14.0 ± 1.8	N/A
UDA	ICCV2015 [70]	FV	A+W	ID	N/A	N/A	14.0 ± 1.8	N/A
ConSE	ICLR2014 [39]	FV	W	ID	36.6 ± 9.0	15.0 ± 2.7	11.6 ± 2.1	20.7 ± 3.1
MTE	ECCV2016 [13]	FV	W	ID	44.3 ± 8.1	19.7 ± 1.6	15.8 ± 1.3	N/A
Objects2Action	ICCV2015 [17]	Ob	W	ID	N/A	15.6	30.3	N/A
ZSECOC	CVPR2017 [15]	FV	W	ID	$\mathbf{59.8 \pm 5.6}$	22.6 ± 1.2	15.1 ± 1.7	N/A
UR	CVPR2018 [16]	FV	W	ID	N/A	24.4 ± 1.6	17.5 ± 1.6	N/A
VDS	IJCAI2018 [24]	FV	W	ID	43.9 ± 7.9	25.3 ± 4.5	25.4 ± 4.1	33.1 ± 5.8
OD	CVPR2019 [71]	I3D	W	ID	50.5 ± 6.9	$\mathbf{30.2 \pm 2.7}$	26.9 ± 2.8	N/A
PS-GNN	Ours	Ob	W	ID	58.8 ± 5.7	25.9 ± 4.1	$\mathbf{36.1 \pm 4.8}$	$\mathbf{36.3 \pm 5.2}$
SVE	ICIP2015 [12]	BoW	W	TD	51.4	22.7	18.7	N/A
UDA	ICCV2015 [70]	FV	A+W	TD	N/A	N/A	14.0 ± 1.8	N/A
PST	NIPS2013 [72]	FV	A	TD	48.6 ± 11.0	N/A	15.3 ± 2.2	N/A
MTE	ECCV2016 [13]	FV	W	TD	56.6 ± 7.7	24.8 ± 2.2	22.9 ± 3.3	N/A
MR	IJCV2017 [14]	FV	W	TD	43.2 ± 8.3	24.1 ± 3.8	22.1 ± 2.5	33.0 ± 4.8
UR	CVPR2018 [16]	FV	W	TD	N/A	28.9 ± 1.2	20.1 ± 1.4	N/A
GA	WACV2018 [73]	C3D	W	TD	41.3 ± 11.4	20.7 ± 3.1	20.3 ± 1.9	N/A
PS-GNN	Ours	Ob	W	TD	$\mathbf{61.8 \pm 6.8}$	$\mathbf{32.6 \pm 2.9}$	$\mathbf{43.0 \pm 4.9}$	$\mathbf{39.8 \pm 4.3}$

*Feature: Fisher Vectors (*FV*), Bag of Words (*BoW*), Object scores(*Ob*), or deep* I3D/C3D *features; Label Embedding: Attribute (*A*) or Word Embeddings(*W*);* ID*: Inductive setting;* TD*: Transductive setting. The average % accuracy ± standard deviation is reported. Note that some methods such as Objects2Action, ZSECOC, UR, GA and OD adopt less than 50 splits for evaluation.*

framework. However, adding visual features is not direct since visual features can not be regarded as a graph node like objects. Further analysis of adding visual features into our framework can be found in Section 4.4.12.

Transductive Setting. As shown in Table 1, compared with the inductive setting, our proposed method achieves better results in the transductive setting due to the use of unlabeled data from unseen categories. Generally, the proposed PS-GNN gets better or comparable results. For the two state-of-the-art methods, MTE and UR, our method outperforms them by $(20.1\%, 7.8\%)$ and $(22.9\%, 3.7\%)$ on UCF101 and HMDB51 datasets, respectively. We also perform comparably on Olympic Sports.

There are another two approaches achieve high performance in recent two years. The Cross-Domain UR (CD-UR) model [16] uses the large-scale action dataset ActivityNet [77] as auxiliary training data. CD-UR obtains the accuracy of 42.5 percent on UCF101, while our PS-GNN achieves comparable performance without using external training videos. Moreover, as shown in Table 1, we outperform its non-cross-domain variant in both inductive and transductive settings. Another object-based method [18] designs a spatial-aware object representation for videos and achieves the accuracy 40.4 percent on UCF101, while our proposed approaches get comparable performance. Note that [18] employs an object detector to leverage the spatial information of objects while our approach does not consider this information. These strategies can also be used to further improve the proposed method.

Generalized ZSAR. Following [73], we perform generalized ZSAR by using 20 percent data from the seen classes for testing and the remaining 80 percent for training. We use three representative baselines for comparison: SJE, ConSE, and GA. Table 2 demonstrates the favorable performance of our PS-GNN model against other competitors, which obtains the accuracy of $(52.9\%, 24.2\%, 35.1\%)$ on the three datasets. The results clearly show that our method has promising generalization ability. Note that the recent OD method obtains better performance on HMDB51, however, OD utilizes the pre-trained feature on Kinetics dataset while our method does not employ external video data. Moreover, the proposed PS-GNN outperforms OD in the inductive zero-shot setting.

4.2.5 Qualitative Analysis

Fig. 3 shows the top-5 scored videos evaluated by our PS-GNN for 4 unseen categories. The videos within the red rectangle represent false-positive samples. We find that the proposed model is able to recognize the actions *kayaking* and *pole vault* since they are highly related to objects. For the action *basketball dunk*, a false positive video from *basketball*

TABLE 2
Results on the Generalized Zero-Shot Setting

Method	Olympic	HMDB51	UCF101
SJE	32.5 ± 6.7	10.5 ± 2.4	8.9 ± 2.2
ConSE	37.6 ± 9.9	15.4 ± 2.8	12.7 ± 2.2
GA	42.2 ± 10.2	20.1 ± 2.1	17.5 ± 2.2
OD	41.8	**26.8**	24.8
PS-GNN(Ours)	$\mathbf{52.9 \pm 6.2}$	24.2 ± 3.3	$\mathbf{35.1 \pm 4.6}$

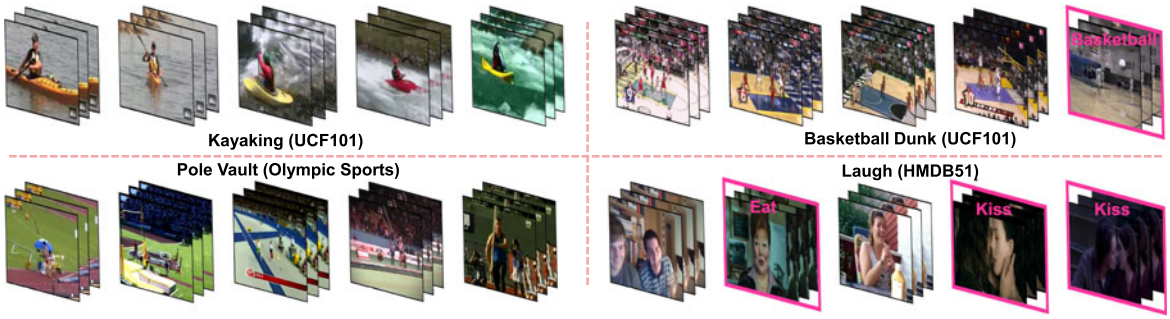


Fig. 3. Top-5 returned video examples for unseen classes on UCF101 (top), Olympic Sports (bottom left) and HMDB51 (bottom right). The videos within the red rectangle represent false-positives.

class is returned because both actions have similar related objects. In addition, some action categories are not very sensitive to objects. For example, videos from *eat* and *kissing* are easily selected as false positives for the action *laugh*. Note that a video category may not only be related to objects, but it could also have relations with other information such as visual/audio features, facial expressions, etc. By adding various information into our framework, the zero-shot performance will be further improved.

4.2.6 Few-Shot Action Recognition

Finally, we extend the proposed framework to the few-shot setting. Here, a few labeled samples of each unseen classes are available during training. The proposed PS-GNN provides a simple way to adapt to this setting by calculating the loss of these new labeled samples (Eq. (9)). We experiment with the few-shot action recognition in a generalized setting, i.e., both seen and unseen (few-shot) classes are adopted in the testing phase. For n-shot experiments, we use the randomly selected $1, 2, 3, 5$ positive videos from the target action categories in 10 independent trials. In this section, to save computational cost and improve task parallelization, the number of selected objects per video category K and the batch size are set to 100 and 48, respectively. The mean accuracies and standard deviations are reported. From Table 3 we can find that the use of a few data samples of the unseen action categories significantly improves the action recognition performance. Note that the state-of-the-art object-based action recognition method [20] achieves $(65.6\%, 38.9\%)$ accuracies on UCF101 and HMDB51 in a fully-supervised way. Compared with [20], our proposed method has comparable performance by using only a few samples from unseen categories.

4.3 ZSL for Unconstrained Videos

In this section, we employ the large-scale Fudan-Columbia Video Dataset (FCVID) [69] and Columbia Consumer Video (CCV) dataset [68] for zero-shot video classification. FCVID contains around 90K web videos annotated manually according to 239 categories. These categories cover a wide range of topics such as procedural events (e.g., "making cake"), objects (e.g., "panda"), social events (e.g., "tailgate party"), scenes (e.g., "beach"), etc. CCV is collected from YouTube, which contains 9,682 videos over 20 semantic categories such as wedding ceremony and graduation. For ZSL on FCVID, we follow [19] to split the video dataset into 160 seen and 79 unseen classes. Moreover, we design another train-test split with 180 seen and 59 unseen classes. The experiments are conducted in an inductive manner. For the CCV dataset, following [14], we adopt the same ZSL setting and split strategy like them in Section 4.2.

TABLE 3
Results on the Few-Shot Action Recognition

Dataset	1-shot	2-shot	3-shot	5-shot
UCF101	45.9 ± 4.1	50.3 ± 3.7	57.6 ± 2.9	61.4 ± 2.5
HMDB51	28.6 ± 5.1	31.8 ± 4.8	32.3 ± 4.0	34.2 ± 3.6
Olympics	58.2 ± 7.6	64.7 ± 8.2	65.9 ± 7.2	68.1 ± 7.3

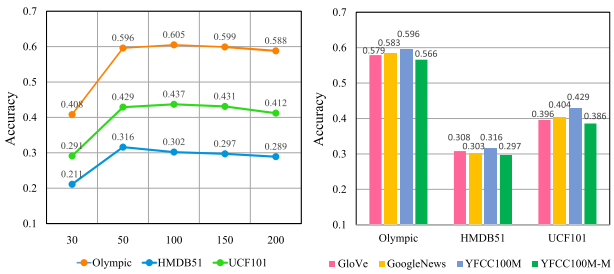
4.3.1 Quantitative Results

*Baselines.*To evaluate the effectiveness of the proposed PS-GNN, in FCVID, we compare our approach with four baselines including DAP, ConSE, Nearest Neighbor (NN) [19], and Object Scene Semantic Fusion (OSF) Network [19]. Here, NN utilizes synthesized class prototypes to infer the label of test videos. OSF designs an object- and scene-based semantic fusion network within a CNN architecture, which uses both object and scene classification networks for generating video representation. For CCV dataset, the adopted baselines are SJE, ConSE, VDS, and MR.

Overall Performance. Our method can also well handle videos from unconstrained categories. On the one hand, the experimental results of FCVID are summarized in Table 4. From this table, we draw several conclusions: (1) All the methods outperform the random guess bound by a large margin, which illustrates the success of zero-shot learning in video classification. (2) Our proposed approach outperforms all the comparison models. Our results improve by 2.0 percent percentage point (or 16.8 percent respectively) over OSF which is also a state-of-the-art method using object semantics. The improvement is largely due to the relationships modeling via

TABLE 4
Zero-Shot Learning Accuracy (%) on FCVID

Train/Test	(160/79)	(180/59)
Chance	1.3	1.7
DAP	9.0	11.0
ConSE	10.6	12.3
NN	8.8	N/A
OSF	11.9	N/A
PS-GNN(Ours)	**13.9**	**16.7**



(a) Effect of K (b) Different Word Embeddings

Fig. 4. Comparison results among (a) different number of selected objects. (b) different types of word embeddings.

the PS-GNN, which is more comprehensive to achieve robust performance. (3) Our approach can achieve favorable zero-shot video classification in both dataset split, indicating that it is robust to capture discriminative information with different scale of datasets. Compared with other state-of-the-art methods, DAP, and ConSE, our method achieves an absolute gain of $(4.9\%, 5.7\%)$ and $(3.3\%, 4.4\%)$ on both splits respectively. On the other hand, we also obtain promising performance on the CCV benchmark. As shown in Table 1, the proposed PS-GNN achieves gains of 9.4, 15.0, and 2.6 percent over three competitors SJE, ConSE and VDS in the inductive setting. We also obtain favorable performance in the transductive setting.

4.4 Ablation Study

In this section, extensive ablation studies are performed to synthetically analyze the proposed PS-GNN model. Specifically, the experiments are conducted on the ZSAR datasets with the transductive setting (20 random splits). We report the mean accuracy in this section.[5]

4.4.1 How Many Objects Should We Use?

The number of the selected objects per video category will influence the size of the built graph. Intuitively, a larger K will result in a more comprehensive graph since more objects are considered to describe a video category. However, a too-large K leads to high computational burden and may bring noises to the constructed graph. As shown in Fig. 4a, a moderate value of K obtains the best performance. Moreover, we observe that UCF101 and Olympic Sports require more objects while we have the best performance when $K = 50$ on HMDB51. The reason is that some categories of HMDB51 are not object-sensitive.

4.4.2 Different Types of Word Embedding

Different types of word embeddings will lead to different representations of video instances and categories. To analyze the sensitivity of the PS-GNN to different types of word embedding methods, we test the performance on another two Word2Vec methods: GloVe [78] and GoogleNews [79]. Fig. 4b shows the comparison between the proposed method and the two baselines. Generally, the word vectors trained on YFCC100M [74] dataset obtain higher accuracy on these datasets. This is because using visual metadata is more suitable for video recognition tasks than training on Wikipedia or GoogleNews data [17]. To further evaluate the generalization ability of our PS-GNN, we build a modified YFCC100M dataset, named YFCC100M-M, for comparison. For YFCC100M-M, we use the name of test classes (except the words appeared in the training classes) as the keywords to exclude the metadata of YFCC100M that contains these words. The results in Fig. 4b demonstrate that our proposed method still achieves favorable performance by using the modified version. Overall, the performance gaps between different types of word embeddings are not significant, which demonstrates that our framework is not sensitive to word representations and has promising generalization ability.

5. In this section, to save computational cost and improve task parallelization, the number of selected objects per video category K and the batch size are set to 100 and 48, respectively.

TABLE 5
Ablation Study of Temporal Modeling

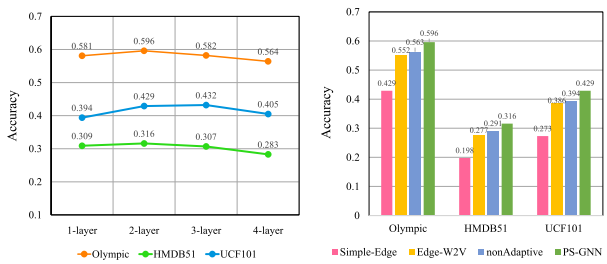
	Olympic	HMDB51	UCF101
PS-GNN-w/o Attention	56.9	26.5	39.5
Attention-w/o resLearning	58.1	27.4	41.0
PS-GNN	59.6	31.6	42.9

4.4.3 Effect of Temporal Modeling

We consider the interaction among different video segments via a temporal-relation attention model with residual learning. To show its effectiveness, we design two baselines PS-GNN-w/o Attention and Attention-w/o resLearning. Here, PS-GNN-w/o Attention removes the proposed attention module and simply averages the object scores of all segments. The baseline Attention-w/o resLearning uses Eq. (5) to directly output the refined object scores without using the residual learning in Eq. (6). Table 5 demonstrates that our proposed method consistently outperforms the baselines, which verifies the effectiveness of the attention design.

4.4.4 Effect of Network Depth

We explore the sensitivity of the network depth in our proposed framework. The results of using different numbers of layers on the three datasets are shown in Fig. 5a. Specifically, we design a 1-layer model with output dimensions of 512, and a 3-layer model which has output channel dimensions of 1024, 512 and 512. And we set the channel dimensions to 1024, 1024, 512, and 512 for the 4-layer variant. Theoretically, a deeper network will enhance the message propagation between nodes thus improve the model robustness. However, we observe that adding more layers does

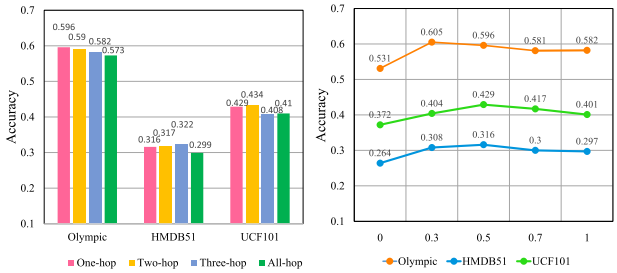


(a) Effect of Network Depth (b) Edge Learning Baselines

Fig. 5. Comparison performance among (a) different number of GNN layers. (b) different baselines of edge representation strategies.

TABLE 6
Comparison Results of Different Model Design

	Olympic	HMDB51	UCF101
OS-GNN	41.2	19.1	26.8
PS-GNN-noUnseen	51.7	26.9	36.4
PS-GNN-w/o C-Loss	58.2	29.7	40.1
CLSWGAN [38]	48.3	29.7	26.6
PS-GNN+Visual	62.7	34.9	43.3
PS-GNN	59.6	31.6	42.9



(a) Effect of i-hop neighbors (b) Effect of γ

Fig. 6. Comparison results among (a) different hops of considered neighbors. (b) different balance factors between the cross entropy loss and the prototype center loss.

not boost the performance of our framework. One potential reason might be that the number of training data is not sufficient (from ~ 700 videos in Olympic Sports to $\sim 10,000$ videos in UCF101) to train a very deep model. In addition, stacking many layers can bring the mixing problem [80], i.e., the features of vertices will converge to the same value

4.4.5 Effect of Knowledge Graph

We use a knowledge graph to initialize the edge weights between nodes. To verify its necessity, we design two baselines: *Simple-Edge* and *Edge-W2V*. The former one abandons the knowledge graph and only employs the weights generated by the MLP (Eq. (7)) for every two nodes. The latter one utilizes Word2Vec embedding to incorporate knowledge into our framework. Specifically, we replace the weights obtained from the knowledge graph with the cosine similarities between corresponding word vectors [33]. Fig. 5b shows the effectiveness of the edge initialization design. The proposed PS-GNN outperforms the baseline Simple-Edge by $(16.7\%, 11.8\%, 15.6\%)$ on the three benchmarks. We also observe that our approach using knowledge graphs achieves better performance, beating the Word2Vec based baseline by up to 4.4 percent with respect to accuracy. The results show the effectiveness of the knowledge graphs. Similar results have also been verified in [33], [59].

4.4.6 Contribution of Adaptive Edge Learning

In Eq. (7), we learn the adaptive edge weights for modeling the dependency between different nodes. The edge leaning procedure plays an important role to generate residual graph refinement in Eq. (8). To explore the contribution of this strategy, we perform PS-GNN on the ZSAR datasets without using adaptive edge learning, i.e., the edges are fixed using knowledge graphs. Fig. 5b exhibits the effectiveness of the adaptive edge learning strategy. Our PS-GNN outperforms the baseline *nonAdaptive* by $(3.3\%, 2.5\%, 3.5\%)$ on the three dataset. With the residual learning process for refining the graph, it is easier to find the appropriate edge weights thus leading to higher model stability and better performance.

4.4.7 Is the Two-Stream Architecture Redundant?

In our two-stream framework, we jointly model the relationships between category-attribute, category-category, and attribute-attribute in an end-to-end manner. In order to explore the importance of this architecture, we design a baseline One-Stream GNN (OS-GNN) which only uses one branch to model the relationships between all concepts. The input to OS-GNN is the word vectors of all the video categories and the weighted attribute features of each video sample generated from Eq. (6). From Table 6 we can observe that the two-stream architecture can significantly improve the zero-shot learning performance. In particular, the proposed PS-GNN outperforms OS-GNN by $(18.3\%, 12.5\%, 16.1\%)$ on the three datasets. Without the co-adaptation and co-operation between the two branches, it is difficult to learn meaningful prototypes and attribute features in a task-driven way, which will result in training dilemma and performance degradation.

4.4.8 Impact of the Prototype Center Loss

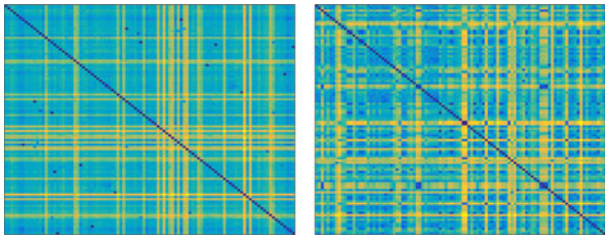
To study the performance of the proposed distance loss, we design a variant, named *PS-GNN-w/o C-Loss*, to illustrate the effectiveness of this loss in Eq. (9). In the proposed PS-GNN, the prototype center loss is able to force the video representations to be close to the class prototypes, which minimizes the intra-class variations and improves the robustness of our framework. As depicted in Table 6, the proposed PS-GNN obtains an absolute gain of $(1.4\%, 1.9\%, 2.8\%)$ compared with the baseline on the three datasets.

4.4.9 Model Training Without Unseen Class

Towards a more practical application scenario, we add a baseline PS-GNN-noUnseen that does not use those unseen class embeddings during training. However, directly removing the unseen categories from our framework results in a severe problem: the prototypes of unseen categories cannot be generalized during the training of our PS-GNN. To enable the PS-GNN-noUnseen in the zero-shot setting, we calculate the prototype of a video category by averaging its one-hop object neighbors. Therefore, at the test phase, we can generalize the prototypes of unseen categories by using the leaned attribute representations from the prototype-branch. As shown in Table 6, without using the unseen categories during training, the baseline gets inferior results compared with the PS-GNN $(7.9\%, 4.7\%, 6.5\%$ decrease on the three datasets). In fact, only using attribute (object) representations as category prototypes ignores the intrinsic relationships between video categories.

4.4.10 Considering Long-Hop Neighborhoods

We conduct an evaluation for analyzing the effect of using long-hop neighborhoods. For the i-hop neighborhoods, we adopt a decay factor $1/i^2$ to represent their importance. As shown in Fig. 6a, two-hop, three-hop, and all-hop neighborhoods are considered for evaluation, respectively. Here,



(a) Word Vector (b) PS-GNN Prototypes

Fig. 7. Distance matrices created using Word2Vec label embeddings and learned prototypes on UCF101. Brighter colors depict larger values.

all-hop denotes that all the objects are considered for video recognition. We can observe that considering two-hop neighbors achieves favorable performance (0.5 and 0.1 percent improvements on UCF101 and HMDB51 datasets, 0.6 percent deterioration on Olympic Sports dataset). However, longer-hop (> 3) will result in a slight performance degradation. This is because utilizing long-hop neighbors may bring some distractions.

4.4.11 Effect of γ

We investigate the contribution of both the cross entropy loss and prototype center loss. Fig. 6b shows the effect of the balance factor γ. We can find that for the three datasets, the accuracy is higher when the cross entropy loss and the prototype center loss are cooperated, which indicates the effectiveness of our multi-task loss. In addition, the cross entropy loss plays a more important role than the prototype center loss since it considers all the video categories.

4.4.12 Combination With Visual Features

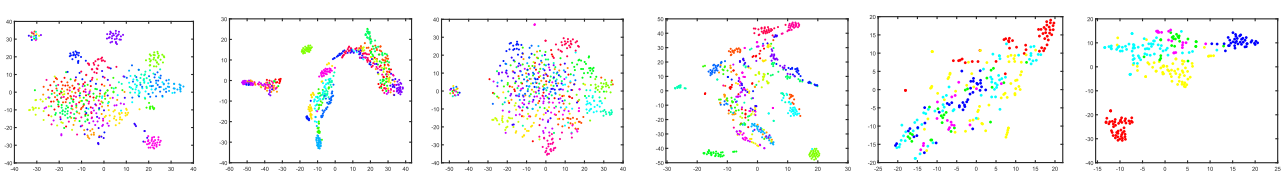
An intuitive idea to boost the performance is to combine visual features with the object information in our framework. However, it is not direct to map visual features such as FV or deep I3D features to graph nodes like object names used in our paper. From this point of view, we design a baseline PS-GNN+Visual, which combines the proposed method with a visual feature-based ZSL model, CLSWGAN [38]. Here, we follow [71] to implement CLSWGAN using I3D features, which achieves the results of $(48.3\%, 29.7\%, 26.6\%)$. We then use a balance factor of 0.7 to combine the prediction scores of both methods linearly. As shown in Table 6, compared with the proposed PS-GNN method, PS-GNN+Visual obtains an absolute gain of 3.1, 3.3, and 0.4 percent on the three datasets, respectively. The results show that our framework can be boosted together with visual features.

4.4.13 Visualization Analysis

To study whether the proposed PS-GNN can learn meaningful category prototypes and video features, we perform visualizations to show some characteristics of our method. Fig. 7 plots the euclidean distance matrices among categories for both word embeddings and learned prototypes on UCF101 dataset. It can be seen that the relationships from word embeddings and prototypes distribute very differently, which demonstrates the proposed method can dig relationships among concepts deeply rather than only using implicit word vectors. Generally, the distances among prototypes are larger than them of word embeddings, which shows the promising discriminative ability of our PS-GNN. To verify the effectiveness of the learned representations, t-SNE [81] visualization is performed for video samples on a random split. we randomly select 600 test videos from 20 unseen categories (UCF101 and HMDB51) and use all the test data from Olympic Sports for visualization. Here, we average all the object node features from the final layer to get the GNN representation. Fig. 8b, 8d, and 8f depict the distribution of the generated features with different colors. We can observe that the learned features are robust since most video samples in the same video category are distributed in a tight area. In addition, we also visualize the averaged initial object features (Eq. (4)) for theses unseen videos. As shown in Fig. 8a, 8c, and 8e, the initial video features are much less discriminative than the finally learned ones, which demonstrates our model can capture high-level representations in the zero-shot setting.

5 Conclusion

In this paper, we have developed a GNN-based zero-shot video classification method, i.e., Prototype-Sample GNN, which can directly and adaptively model all the relationships between concepts in an end-to-end manner. By carefully designing a two-stream architecture with a prototype branch and a sample branch, the proposed method recasts the ZSL task into a neural message-passing process in the concept domain. In our framework, both branches can co-adapt and cooperate to achieve robust ZSL performance. To generate more informative representations for ZSL, we introduce object semantics, a multi-task loss function, knowledge graphs, and adaptive learning into our framework. Overall, our PS-GNN is able to enjoy the merits of alleviated heterogeneity gap, low domain shift, and robust temporal modeling. The comprehensive evaluation has been performed by comparing our approach with state-of-the-arts on five popular benchmarks. The effectiveness is evidenced by its favorable performance compared with others.



(a) UCF-Ini (b) UCF-GNN (c) HMDB-Ini (d) HMDB-GNN (e) Olympic-Ini (f) Olympic-GNN

Fig. 8. The t-SNE visualization of video features over a random split from UCF101, HMDB51, and Olympic datasets. Specifically, (a), (c) and (e) indicate the average representation of the initial object features (Eq. (4)); (b), (d), (f) corresponds to the averaged representation of the learned object-feature. For UCF101 and HMDB51 datasets, we randomly select 600 test videos from 20 unseen categories. For the Olympic Sports dataset, we use all the test videos for visualization.

Future work may include exploring richer information to learn prototype/video features and designing more generalized strategies for our framework. For the former problem, other knowledge information can be employed such as the types of edges. Besides, it will be interesting to incorporate visual features into the graph neural networks by considering the relationships between video samples. We will also test other types of methods for relationship modeling in zero-shot video classification, such as other types of graph networks and graph embedding methods. For the second problem, we have explored the adaptive edge learning in the graph. However, it is hard to handle the dynamic graph learning issue, i.e., the numbers of graph nodes are different for various videos. This is also an open problem in the field of graph learning. In addition, motivated by the favorable performance of our framework, we intend to apply this approach to other tasks, such as multi-label zero-shot learning and zero-shot video retrieval.

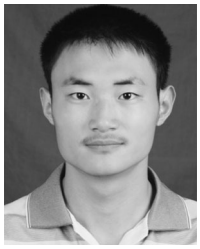
ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grants 2018AAA0102200, in part by the National Natural Science Foundation of China under Grants 61720106006, 61721004, 61832002, 61532009, U1705262, U1836220, and 61702511, in part by the Key Research Program of Frontier Sciences, CAS, under Grant QYZDJSSWJSC039, and in part by the Research Program of National Laboratory of Pattern Recognition under Grant Z-2018007.

REFERENCES

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.
[2] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 413–431.
[3] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6450–6459.
[4] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.
[5] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4724–4733.
[6] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, 2019, Art. no. 13.
[7] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 951–958.
[8] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3337–3344.
[9] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Learning multimodal latent attributes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 303–316, Feb. 2014.
[10] Y. Long and L. Shao, "Learning to recognise unseen classes by a few similes," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 636–644.
[11] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 2152–2161.
[12] X. Xu, T. Hospedales, and S. Gong, "Semantic embedding space for zero-shot action recognition," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 63–67.
[13] X. Xu, T. M. Hospedales, and S. Gong, "Multi-task zero-shot action recognition with prioritised data augmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 343–359.
[14] X. Xu, T. Hospedales, and S. Gong, "Transductive zero-shot action recognition by word-vector embedding," *Int. J. Comput. Vis.*, vol. 123, no. 3, pp. 309–333, 2017.
[15] J. Qin *et al.*, "Zero-shot action recognition with error-correcting output codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1042–1051.
[16] Y. Zhu, Y. Long, Y. Guan, S. Newsam, and L. Shao, "Towards universal representation for unseen action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9436–9445.
[17] M. Jain, J. C. van Gemert, T. Mensink, and C. G. Snoek, "Objects2action: Classifying and localizing actions without any video example," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4588–4596.
[18] P. Mettes and C. G. Snoek, "Spatial-aware object embeddings for zero-shot localization and classification of actions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4453–4462.
[19] Z. Wu, Y. Fu, Y.-G. Jiang, and L. Sigal, "Harnessing object and scene semantics for large-scale video understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3112–3121.
[20] M. Jain, J. C. Van Gemert , and C. G. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 46–55.
[21] Y. Tian, Y. Kong, Q. Ruan, G. An, and Y. Fu, "Aligned dynamic-preserving embedding for zero-shot action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2019.2908487.
[22] C. Gan, T. Yang, and B. Gong, "Learning attributes equals multi-source domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 87–97.
[23] I. Alexiou, T. Xiang, and S. Gong, "Exploring synonyms as context in zero-shot action recognition," in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 4190–4194.
[24] C. Zhang and Y. Peng, "Visual data synthesis via GAN for zero-shot video classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 1128–1134.
[25] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.
[26] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4694–4702.
[27] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2016.
[29] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[30] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 20–28.
[31] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. F. Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1576–1585.
[32] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018.
[33] J. Gao, T. Zhang, and C. Xu, "Watch, think and attend: End-to-end video classification via dynamic knowledge evolution modeling," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 690–699.
[34] J. Gao, T. Zhang, and C. Xu, "I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs," in *Proc. AAAI Conf. Artif. Intell.*, 2019.
[35] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2251–2265, Sep. 2019.
[36] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, 2019, Art. no. 13.

[37] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2927–2936.
[38] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5542–5551.
[39] M. Norouzi *et al.*, "Zero-shot learning by convex combination of semantic embeddings," in *Proc. Int. Conf. Learn. Representations*, 2014.
[40] Z. Fu, T. Xiang, E. Kodirov, and S. Gong, "Zero-shot learning on semantic class prototype graph," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 2009–2022, Aug. 2018.
[41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1725–1732.
[42] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
[43] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with Pseudo-3D residual networks," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 5534–5542.
[44] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 843–852.
[45] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 34–45.
[46] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 6202–6211.
[47] C. Gan, T. Yang, and B. Gong, "Learning attributes equals multi-source domain generalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 87–97.
[48] C. Gan, M. Lin, Y. Yang, Y. Zhuang, and A. G. Hauptmann, "Exploring semantic interclass relationships (SIR) for zero-shot action recognition," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 3769–3775.
[49] C. Gan, M. Lin, Y. Yang, G. de Melo, and A. G. Hauptmann, "Concepts not alone: Exploring pairwise relationships for zero-shot video activity recognition," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3487–3493.
[50] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," 2019, *arXiv:1901.00596*.
[51] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," 2018, *arXiv: 1812.08434*.
[52] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[53] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow, "Gated graph sequence neural networks," in *Proc. Int. Conf. Learn. Representations*, 2016.
[54] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun , "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*.
[55] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang, "Person re-identification with deep similarity-guided graph neural network," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 508–526.
[56] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6857–6866.
[57] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.
[58] Y. Fang, K. Kuan, J. Lin, C. Tan, and V. Chandrasekhar, "Object detection meets knowledge graphs," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1661–1667.
[59] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4444–4451.
[60] F. Bond and R. Foster, "Linking and extending an open multilingual wordnet," in *Proc. 51st Annu. Meet. Assoc. Comput. Linguistics*, 2013, pp. 1352–1362.
[61] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a web of open data," in *The Semantic Web*. Berlin, Germany: Springer, 2007, pp. 722–735.
[62] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[63] P. Mettes, D. C. Koelma, and C. G. Snoek, "The ImageNet shuffle: Reorganized pre-training for video event detection," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2016, pp. 175–182.
[64] Z. Yang *et al.*, "GLoMo: Unsupervisedly learned relational graphs as transferable representations," in *Proc. Neural Inf. Process. Syst.*, 2018.
[65] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 392–405.
[66] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2556–2563.
[67] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
[68] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, "Consumer video understanding: A benchmark database and an evaluation of human and machine performance," in *Proc. 1st ACM Int. Conf. Multimedia Retrieval*, 2011, Art. no. 29.
[69] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 352–364, Feb. 2018.
[70] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2452–2460.
[71] D. Mandal *et al.*, "Out-of-distribution detection for generalized zero-shot action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9977–9985.
[72] M. Rohrbach *et al.*, "Recognizing fine-grained and composite activities using hand-centric features and script data," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 346–373, 2016.
[73] A. Mishra, V. K. Verma, M. S. K. Reddy, S. Arulkumar, P. Rai, and A. Mittal, "A generative approach to zero-shot and few-shot action recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 372–380.
[74] B. Thomee *et al.*, "YFCC100M: The new data in multimedia research," *Commun. ACM*, vol. 59, no. 2, pp. 64–73, 2016.
[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[76] J. Song, C. Shen, Y. Yang, Y. Liu, and M. Song, "Transductive unbiased embedding for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1024–1033.
[77] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 961–970.
[78] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
[79] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[80] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3538–3545.
[81] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov., pp. 2579–2605, 2008.



Junyu Gao received the bachelor's degree in computer science from Xi'an JiaoTong University, Xi'an, Shaanxi, China, in 2015. He is currently working toward the PhD degree in the Multimedia Computing Group, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include multimedia analysis and computer vision, especially deep learning, multimedia computing, and video understanding.

Tianzhu Zhang (Member, IEEE) received the bachelor's degree in communications and information technology from the Beijing Institute of Technology, Beijing, China, in 2006, and the PhD degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2011. Currently, he is a professor with the University of Science and Technology of China (USTC). His current research interests include computer vision and multimedia, especially action recognition, object classification, object tracking, and social event analysis.

Changsheng Xu (Fellow, IEEE) is currently a distinguished professor with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include multimedia content analysis/indexing/retrieval, pattern recognition, and computer vision. He has hold 40 granted/pending patents and published more than 300 refereed research papers in these areas. He has served as associate editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair, and TPC member for more than 20 IEEE and ACM prestigious multimedia journals, conferences and workshops, including the *IEEE Transactions on Multimedia*, the *ACM Transactions on Multimedia Computing, Communications, and Applications* and ACM Multimedia conference. He is IAPR fellow and ACM distinguished scientist.