Semisupervised Action Recognition with Adaptive Correlation Learning

Fan Wang[1], Zengmin Xu[1,3,4*], Jiakun Chen[1], Ruimin Hu[2]

[1]School of Mathematics and Computing Science,Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Guilin, China.
[2]National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan, China.
[3]Center for Applied Mathematics of Guangxi (GUET) , Guilin, China.
[4]Anview.ai, Guilin, China.

*Corresponding author(s). E-mail(s): xzm@guet.edu.cn;
Contributing authors: wf@mails.guet.edu.cn; jiakunchn@gmail.com;

Abstract

There is a huge amount of video image data in the action recognition domain, and it is unreasonable to use expensive manual annotation. Traditional semi-supervised learning applies graph embedding and label propagation to mine local neighborhood relationships between labeled and unlabeled data. However, graph-based modeling methods have limited effectiveness for unstructured action videos. Recently, Graph Convolutional Networks (GCNs) have been used to exploit local neighborhood relationships of samples (action videos). However, existing GCNs methods struggle to extract discriminative high-level features from fixed graph, and suffer from excessive computational complexity when dealing with large-scale data. To address these issues, we propose a new GCN-based semisupervised method with adaptive feature correlation, which enhances local neighborhood by computing its correlation weights and learns global topology from labeled and unlabeled samples to obtain the optimal graph structures, effectively extracting high-level features. Furthermore, owing to the complexity caused by the inevitable redundant computations of GCNs, we apply linear transformations to the features of neighbor graph nodes, then aggregate adjacent nodes' features for capturing the local neighborhood information. Thus, we mitigate this excess complexity by removing nonlinearity and collapsing weight matrices between consecutive layers, thereby addressing the issue of computational complexity . This linear model is simpler than traditional GCN models and offers superior generalization, robustness, and efficiency. The proposed approach achieves comparable performance on UCF101 using only *0.15* $\times$ N labeled training data. On HMDB51 and

Something-Something V2, our method improves the recognition accuracy by *+1.7%* and *+2%* respectively, using only $0.20 \times N$ labeled training data.

Keywords: Graph convolutional networks, Adaptive correlation learning, Semisupervised learning, Global topology optimization

1 Introduction

Action recognition is becoming increasingly important in various application fields, including video surveillance, action analysis in athletic events, and human-computer interaction. It has attracted significant research interest in computer vision [1–3]. These research methods respectively address fabric defect detection in practical industrial manufacturing, as well as the issues of helmet detection in road surveillance and activating discriminative cues in deep feature maps for image retrieval. In particular, some methods improve image dehazing by integrating partial Siamese frameworks with multiscale dual encoding and decoding information fusion, as well as multi-level feature interaction and non-local information enhanced channel attention mechanisms [4, 5]. Fully supervised learning requires substantial labeled data to execute recognition tasks effectively. However, unlike unlabeled data, labeled data are scarce in the real world. Thus, we focus on how to leverage unlabeled data for exploring feature correlations.

Action recognition based on Convolutional Neural Networks (CNN) has achieved significant results in recent years [6–12]. CNN-based action recognition involves end-to-end model training, extracting various types of features (RGB frames or optical flows) from videos using diverse network architectures [13, 14]. Most CNN-based methods utilize local filters with deep learning features and common parameters to process unstructured data. In contrast, Graph Convolutional Networks can operate on structured data. Nevertheless, They represent an effective variant of CNN for graph-structured data, achieving new node representations through feature aggregation from adjacent nodes [15]. However, GCNs primarily draw inspiration from recent deep learning approaches, potentially inheriting unnecessary complexity and redundant computations. To address these issues, a general solution involves removing nonlinear functions from GCNs, thereby reducing computational complexity [16]. In this study, videos are regarded as a specific case of unstructured data, and adaptive GCN is constructed for enhancing feature modeling.

Although action recognition technology has exhibited significant advancements [17–19], several obstacles remain. Large datasets such as Sports-1M and Kinetics-400 require expensive manual annotation. A fully supervised model cannot achieve high performance with insufficient training samples. In contrast, semisupervised learning techniques can use labeled and unlabeled data. During the model training phase, such techniques typically only employ a small quantity of labeled data and a substantial amount of unlabeled data. However, most semisupervised learning algorithms cannot outperform fully supervised learning techniques. Considering the properties of GCNs, we believe that graph-convoluted features extracted by exploring feature correlations can improve semisupervised recognition performance.

On the other hand, during the manual video sample labeling, video samples with stronger correlations are more likely to have the same behavioral category. Nevertheless, most existing deep learning methods simply consider the impacts between different frames in a single video, but how to explicitly define several videos for correlation purposes remains unclear. Therefore, we focus on the feature correlations and global topology among different videos for high-level feature representations.

However, the above-mentioned methods are constrained to fixed graphs, which limits their application scope. Hence, designing a learnable model for general graph structures is important. We present a novel GCN-based Semisupervised method with Adaptive Correlation (GSAC) to overcome the abovementioned issues. GSAC analyzes the correlations among videos and employs feature aggregation to improve feature representations. We apply a weighted strategy that only considers the influences among correlated samples. After several training rounds, our proposed method can learn optimal graph structures effectively. The contributions of this study are summarized as follows:

- This study is the first RGB-based semisupervised action recognition work to investigate adaptive feature correlation and GCNs. Both local neighborhood information and global topology structure are considered by modeling a potential feature subspace.
- Semisupervised node classification with correlated regularization is employed. We leverage the aggregated features with both labeled and unlabeled data to avoid incorrect adjacency, especially without unreliable pseudo-labels during traditional training process.
- The joint optimization algorithm is analyzed to demonstrate its superiority over other semisupervised approaches. Extensive experiments conducted on three benchmarks—UCF101, HMDB51, and Something-Something V2—validate the comparable performance of the proposed method to others, specifically achieving improvements of +1.7% and +2% in accuracy on HMDB51 and Something-Something V2, respectively. These results indicate that our approach enhances action recognition performance and significantly reduces the dependence on labeled data.

2 Related Work

2.1 SemiSupervised Learning

Semisupervised learning has been extensively used in numerous fields [20–23]. Fully supervised learning is expensive, as a significant volume of training data must be identified. Unlabeled samples can be used to learn data correlations in semisupervised learning, and thus, it is advantageous for vision tasks and in terms of human labor costs. In recent years, many semisupervised learning techniques have been presented for various vision tasks. Luo et al. [24] demonstrated an adaptive semisupervised feature selection technique for identifying video semantic contents. Xu et al. [25] proposed a semisupervised action recognition approach based on discriminant manifold learning, which concurrently modeled the compactness and separability. Si et al. [26] introduced a Adversarial Self-Supervised Learning (ASSL), a novel framework that tightly couples self-supervised learning and the semi-supervised scheme via neighbor relation exploration and adversarial learning. Khan et al. [27] developed a unique semisupervised inception neural network ensemble-based architecture to impute missing labels. Bi et al. [28] constructed a novel human activity recognition paradigm,

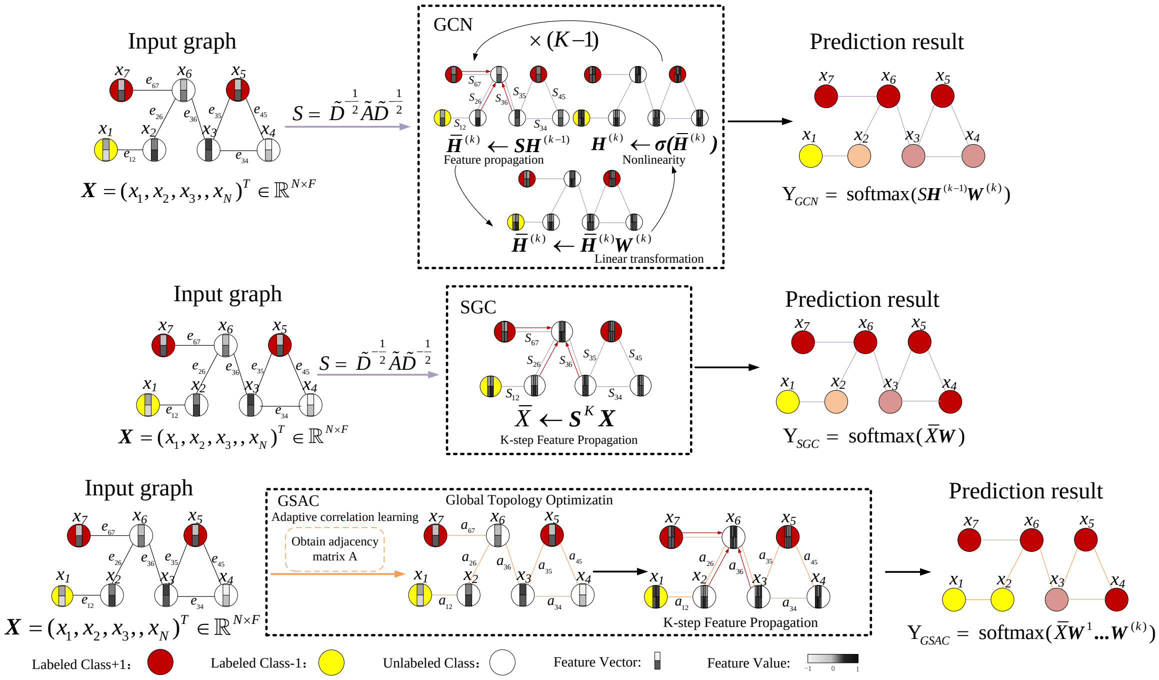


Fig. 1: Comparison with GCN [15], SGC [16] and GSAC. Top row: The GCN sequentially transforms the feature vectors across K layers and then applies a linear classifier to the final representation. Middle row: SGC streamlines the entire process, simplifying it to a single feature propagation step followed by straightforward logistic regression. Bottom row: GSAC obtains better correlations between graph nodes through adaptive correlation learning with simple feature propagation.

that integrated active learning and semisupervised learning into one framework, by actively selecting the most insightful examples for annotation. In response to the above-mentioned studies, we combine semisupervised learning and GCNs to recognize human actions.

2.2 Adaptive Correlation Learning

Zhang et al. [29] proposed an adaptation method that uses knowledge information derived from images to improve video action classification. Veličković et al. [30] created a graph attention network (GAT) by stacking layers; where the nodes can attend to the features of their neighborhoods. The GAT operates on graph-structured data and addresses the drawbacks of the previously developed methods based on graph convolutions or their approximations, whereby the features of adjacent nodes are combined to produce an embedding representation of the core node. Jiang et al. [31] suggested the novel graph learning-convolutional network (GLCN) framework to learn the ideal graph topology adaptively. Han et al. [32] introduced the Point2Node end-to-end graph model to represent a specific point cloud, which can examine the correlations among all network nodes at various levels to aggregate the newly learned properties. Ma et al. [33] introduced a new hashing method called Correlation Filtering Hashing (CFH), which improves fine-grained image retrieval by integrating semantic information with visual features. Ying et al. [34] developed an autoencoder-based adaptive feature fusion approach, utilized softmax normalization for additional learning to extract the features from the convolutional and completely connected layers, then sending them to an autoencoder.

2.3 Graph Convolutional Networks

Previous works have stated that GCNs can analyze a wide range of graph-structured data. Hamilton et al. [35] presented a node embedding method that randomly selects and combines features from the vicinity of nodes. Kipf et al. [15] introduced a hierarchical propagation strategy for neural network models and constructed an effective semisupervised GCN based on graph-structured data. Thakkar et al. [36] developed a portioned GCN, divided a bone graph into four subgraphs with shared joints, and employed this network on the recognition model. Zeng et al. [37] proposed leveraging proposal-proposal interactions with GCNs to localize temporal actions. Manessi et al. [38] designed two cutting-edge network that combine GCNs and long short-term memory networks for learning both long-term dependencies and graph structure information. Sofianos et al. [39] exhibited a space-time-separable GCN (STS-GCN) for posture forecasting. The STS-GCN is the first method to use a GCN for describing human stance dynamics. Qiu et al. [40] learned an effective multistream-based skeleton topology and a semantically guided adaptive GCN for action recognition. Gan et al. [41] illustrated a multigraph fusion technique to create a superior graph and derived a low-dimensional space from the original high-dimensional data for their GCN model.

3 Proposed Approach

This section describes the formulation of the proposed approach. We first introduce the local neighborhood from adjacent edges between connected graph nodes, then develop a feature aggregation method to enhance feature representation of different videos. At the same time, an adaptive correlation learning module is proposed to calculate the correlation weights among samples. Last but not the least, a single-layer graph convolution is constructed by feature propagation, which can derive the optimal global graph structure, thus obtain a graph convolutional model for action recognition. Fig. 1 depicts the differences of GCN, SGC and GSAC.

3.1 Formulation

The original features of the given video are represented as $\boldsymbol{X} = (\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \boldsymbol{x}_3,\ ...,\ \boldsymbol{x}_N)^T \in \mathbb{R}^{N\times F}$, where $\boldsymbol{x}_i \in \mathbb{R}^{F\times 1}$ denotes the feature of the i-th video, N is the number of videos, and F is the feature dimensionality of the video. An undirected graph is represented as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges in the graph, respectively. $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ represents the adjacency matrix of the graph $G = (\mathcal{V}, \mathcal{E})$, where $N = |\mathcal{V}|$ is the number of graph nodes. The degree matrix $\boldsymbol{D} = (\boldsymbol{d}_1,\ \boldsymbol{d}_2,\ ...,\ \boldsymbol{d}_n)$ is a diagonal matrix. In the following description, we consider each video sample to be a graph node and define the edges of the graph by employing the relationships among different videos. Fig. 2 depicts the step-by-step GSAC training process.

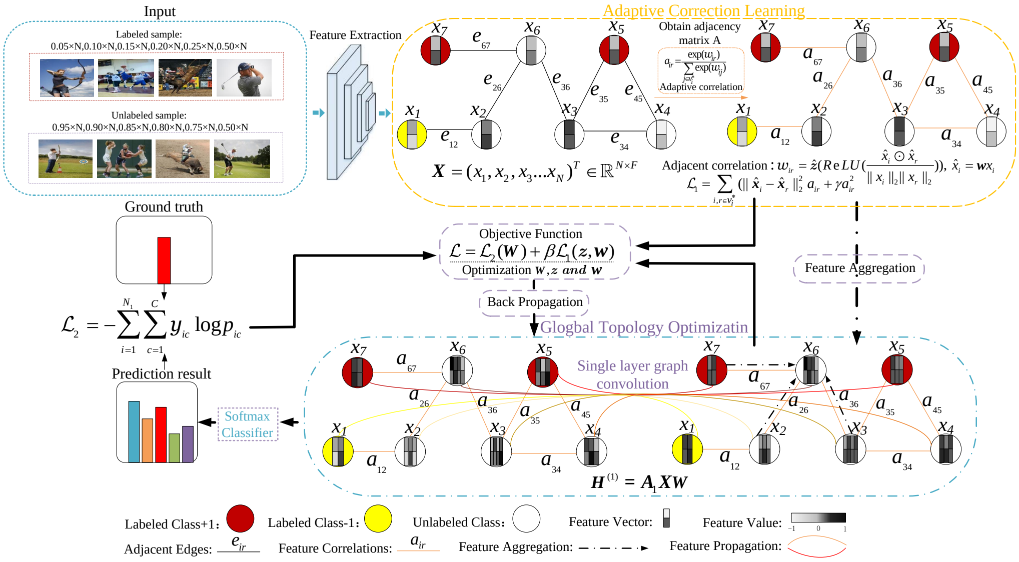

Input
Labeled sample:
0.05×N,0.10×N,0.15×N,0.20×N,0.25×N,0.50×N
Unlabeled sample:
0.95×N,0.90×N,0.85×N,0.80×N,0.75×N,0.50×N
Feature Extraction
Adaptive Correction Learning
$X = (x_1, x_2, x_3 \dots x_N)^T \in \mathbb{R}^{N \times F}$
Obtain adjacency matrix A
$a_{ir} = \frac{\exp(w_{ir})}{\sum_{j \in V_i^*} \exp(w_{ij})}$
Adaptive correlation
Adjacent correlation : $w_{ir} = \hat{z}(ReLU(\frac{\hat{x}_i \odot \hat{x}_r}{\| x_i \|_2 \| x_r \|_2})), \hat{x}_i = wx_i$
$\mathcal{L}_1 = \sum_{i,r \in V_i^*} (\| \hat{x}_i - \hat{x}_r \|_2^2 a_{ir} + \gamma a_{ir}^2$
Ground truth
$\mathcal{L}_2 = -\sum_{i=1}^{N_1} \sum_{c=1}^{C} y_{ic} \log p_{ic}$
Objective Function
$\mathcal{L} = \mathcal{L}_2(W) + \beta \mathcal{L}_1(z, w)$
Optimization w, z and w
Back Propagation
Feature Aggregation
Glogbal Topology Optimizatin
Single layer graph convolution
$H^{(1)} = A_1 XW$
Prediction result
Softmax Classifier
Labeled Class+1:
Labeled Class-1:
Unlabeled Class:
Feature Vector:
Feature Value:
Adjacent Edges: e_{ir}
Feature Correlations: a_{ir}
Feature Aggregation:
Feature Propagation:

Fig. 2: The training process of the proposed semisupervised learning method involves several steps. It starts with a feature extractor and then constructs a K-nearest neighbor graph. Next, an adaptive correlation module is employed to optimize the relationships between different nodes, achieving global topology optimization. A single-layer graph convolution aggregates the features between the nodes (for instance, the features of x_6 are composed of the features from x_7, x_2, x_3, and x_6 itself). Finally, the updated node high-level features are passed through a softmax layer to obtain the prediction results.

In an undirected graph $G = (\mathcal{V}, \mathcal{E})$, if the relationships between each central node and other nodes remain unknown, it can be assumed that there may exist edges between each central node and the others. As the correlation between nodes in most graphs tends to be relatively low, some edge definitions might lead to inappropriate relationships for the central nodes, rendering it ineffective in representing the relationships among nodes properly. In this research, we construct a K-nearest neighbor graph by employing video features [42], which can be formulated as follows:

$$e_{ij} = \begin{cases} 1, & \boldsymbol{x}_j \in \boldsymbol{N}_K(\boldsymbol{x}_i) \text{ or } \boldsymbol{x}_i \in \boldsymbol{N}_K(\boldsymbol{x}_j); \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $\boldsymbol{N}_K(\boldsymbol{x}_i)$ denotes the set of the K-nearest neighbors to $\boldsymbol{x}_i$ in the original feature space. Furthermore, $e_{ij} \in \mathcal{E}$, and when $e_{ij} = 1$, nodes v_i and v_j have connected edges; otherwise, no connected edges are present.

We construct the adjacent correlation of each node to capture local neighborhood information of all samples for better feature embeddings. For any node $v_i \in \mathcal{V}$ in the graph $G = (\mathcal{V}, \mathcal{E})$, the neighborhood features x_r of central node v_i can be aggregated with a weighted scalar w_{ir}, which can be formulated as follows:

$$\boldsymbol{h}_i^* = \sum_{r \in \mathcal{V}_r} w_{ir} \boldsymbol{x}_r,\ \boldsymbol{h}_i^* \in \mathbb{R}^{F \times 1}, \tag{2}$$

where $\mathcal{V}_r$ represents the adjacent set of node v_i, $\boldsymbol{x}_r$ represents the feature of adjacent node v_r.

An adjacent edge between nodes v_i and v_r indicates they are related. Therefore, the weighted scalar w_{ir} can be defined as the correlation between v_i and v_r:

$$w_{ir} = sim(\boldsymbol{x}_i,\ \boldsymbol{x}_r), \tag{3}$$

where w_{ir} can quantify the similarity between different nodes embedding. Once w_{ir} has been determined, those adjacent nodes v_r capturing correlative characteristics for central nodes v_i are beneficial to the feature aggregation.

3.2 Adaptive Correlation Learning

In traditional Graph Convolutional Networks, most static adjacency matrices are unchanged and typically suitable for structured data. However, unstructured video data exhibits changing adjacency relationships between sample points. Therefore, we propose an adaptive correlation learning module to calculate the correlation weights among samples. A specific value for w_{ir} is derived using a learnable module that parameterizes $sim(\boldsymbol{x}_i,\ \boldsymbol{x}_r)$. Consequently, equation (3) is reformulated as

$$w_{ir} = \boldsymbol{z}(\frac{\boldsymbol{x}_i \odot \boldsymbol{x}_r}{||\boldsymbol{x}_i||_2 ||\boldsymbol{x}_r||_2}), \tag{4}$$

where $\boldsymbol{z} \in \mathbb{R}^{1 \times F}$ represents a weighted vector and $\odot$ represents the Hadamard product.

As the dimension of the original feature is relatively high, we transfer the input features to a lower-dimensional space, as follows:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{\mathcal{W}} \boldsymbol{x}_i, \tag{5}$$

where $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{F' \times F} (F' < F)$ represents a learnable linear transformation matrix. This enables the dimension reduction. Thus, equation (4) becomes

$$w_{ir} = \hat{\boldsymbol{z}}(\mathrm{ReLU}(\frac{\hat{\boldsymbol{x}}_i \odot \hat{\boldsymbol{x}}_r}{||\hat{\boldsymbol{x}}_i||_2 ||\hat{\boldsymbol{x}}_r||_2})), \tag{6}$$

where $\hat{\boldsymbol{z}} \in \mathbb{R}^{1 \times F'}$ and ReLU($*$)=max(0, $*$) represents a rectified linear unit activation function that increases feature sparsity and enhances feature-fitting ability.

The adjacent correlation weight w_{ir} between nodes v_i and v_r can be calculated by equation (6). If $w_{ir} > 0$, it is inferred that nodes v_i and v_r are related; otherwise, they are considered unrelated. When using equation (2) for feature aggregation, in order to mitigate the influence of nodes in $\mathcal{V}_i$ that are not related to node v_i, the outcomes of equation (6) that less than 0 will be set to 0. In other words, if the adjacent correlation weight w_{ir} calculated by equation (6) is less than 0, then the features of corresponding nodes v_r will not be utilized in the feature aggregation process. Consequently, this ensures those central nodes v_i only aggregate the features of related nodes, thereby obtaining more effective feature representation.

Let $\mathcal{V}_i^*$ denote a subset of set $\mathcal{V}_i$, where each node in $\mathcal{V}_i^*$ is correlated with the central node v_i. Furthermore, $\boldsymbol{w}_i$ denotes the weighted vector that can be obtained by performing correlation calculations among the central node v_i and the nodes in $\mathcal{V}_i^*$, with each element in $\boldsymbol{w}_i$ obtained by computing w_{ir}, $v_r \in \mathcal{V}_i^*$. For a better comparison of the correlations among different nodes in $\mathcal{V}_i^*$ and the central node v_i, the nonlinear softmax function is used to normalize $\boldsymbol{w}_i$, and a new weighted vector $\boldsymbol{a}_i$ is obtained as follows:

$$\boldsymbol{a}_i = \mathrm{softmax}(\boldsymbol{w}_i). \tag{7}$$

As previously mentioned, the final adaptive correlation weight can be formulated as

$$a_{ir} = \frac{\exp(w_{ir})}{\sum\limits_{j \in \mathcal{V}_i^*} \exp(w_{ij})},\ r \in \mathcal{V}_i^*, \tag{8}$$

where a_{ir} represents an element in $\boldsymbol{a}_i$. As mentioned by Nie et al. [43], if nodes v_i and v_r have a smaller distance between them $||\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_r||_2^2$, a larger correlation weight a_{ir} should be assigned. Therefore, when equation (8) is used to compute the adaptive correlation weight a_{ir}, the optimal correlation weight can be obtained by minimizing the following loss function:

$$\mathcal{L}_1 = \sum_{i,r \in \mathcal{V}_i^*} (||\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_r||_2^2 a_{ir} + \gamma a_{ir}^2), \tag{9}$$

where the second term represents a regularization term and γ represents a regularization hyperparameter. If the regularization term is not included, a trivial solution is obtained; the nearest node of central node v_i is assigned a correlation weight to 1. In contrast, the other nodes are assigned a correlation weight of 0. Therefore, the second term in equation (9) is significant.

3.3 Global Topology Optimization

According to the above-mentioned analysis, the adaptive correlation weight a_{ir} can be incorporated into equation (2), which can be rewritten as follows:

$$\boldsymbol{h}_i = \sum_{r \in \mathcal{V}_i^*} a_{ir} \boldsymbol{x}_r,\ \boldsymbol{h}_i \in \mathbb{R}^{F \times 1}. \tag{10}$$

We introduce a learnable shared linear module $\boldsymbol{W} \in \mathbb{R}^{F \times \hat{F}}$, then obtain a new formulation of the node high-level features to improve the feature representation:

$$\hat{\boldsymbol{h}}_i = \boldsymbol{h}_i^T \boldsymbol{W},\ \hat{\boldsymbol{h}}_i \in \mathbb{R}^{1 \times \hat{F}}. \tag{11}$$

By combining equations (10) and (11), we obtain

$$\hat{\boldsymbol{h}}_i = \sum_{r \in \mathcal{V}_i^*} a_{ir} \boldsymbol{x}_r^T \boldsymbol{W}. \tag{12}$$

Let a_{ir} be the element in the i-th row and r-th column of the adjacency matrix $\boldsymbol{A}$ in an undirected graph $G = (\mathcal{V}, \mathcal{E})$. $\hat{\boldsymbol{H}} = (\hat{\boldsymbol{h}}_1^T,\ \hat{\boldsymbol{h}}_2^T,\ \hat{\boldsymbol{h}}_3^T,\ ...,\ \hat{\boldsymbol{h}}_N^T)^T \in \mathbb{R}^{N \times \hat{F}}$, that is, each row of $\hat{\boldsymbol{H}}$ denotes a video's feature representation. The original features of all videos can be converted to equation (13), and the output features $\hat{\boldsymbol{H}}$ of the first layer in GCNs becomes:

$$\hat{\boldsymbol{H}} = \boldsymbol{A_1XW}^{(1)}. \tag{13}$$

Since original features $\boldsymbol{X}$ only perform feature aggregation in the first layer of GCNs, $\hat{\boldsymbol{H}}$ denotes the input/output aggregated feature in subsequent layers of GCNs. Given a nonlinear activation function $\sigma(\cdot)$, $\hat{\boldsymbol{H}}$ obtains the input feature $\hat{\boldsymbol{H}}^{(1)} = \sigma(\hat{\boldsymbol{H}}) = \sigma(\boldsymbol{A_1XW^{(1)}})$ for the second layer of the model. Based on previous work with GCNs [15], the proposed model is composed of a two-layer GCNs, and the output of each layer is activated by a nonlinear activation function:

$$\begin{aligned} \hat{\boldsymbol{H}}^{(2)} &= \sigma(\boldsymbol{A}_2\hat{\boldsymbol{H}}^{(1)}\boldsymbol{W}^{(2)}) \\ &= \sigma(\boldsymbol{A}_2\sigma(\boldsymbol{A}_1\boldsymbol{X}\boldsymbol{W}^{(1)})\boldsymbol{W}^{(2)}), \end{aligned} \tag{14}$$

A general graph convolutional model can be obtained by stacking equation (13) in multiple layers:

$$\hat{\boldsymbol{H}}^{(k)} = \sigma(\boldsymbol{A}_k\sigma(\hat{\boldsymbol{H}}^{(k-1)})\boldsymbol{W}^{(k)}), \tag{15}$$

where $\boldsymbol{A}_k$ is the adjacency matrix of the k-th layer, which the adaptive correlation learning module can obtain, $\hat{\boldsymbol{H}}^{(k-1)}$ is the input feature matrix of the k-th layer, $\hat{\boldsymbol{H}}^{(k)}$ is the output feature matrix of the k-th layer, $\hat{\boldsymbol{H}}^{(0)} = \boldsymbol{X}$, and $\boldsymbol{W}^{(k)}$ is the learnable weight matrix of the k-th layer.

However, according to GCN simplifying assumption [16], the nonlinearity between GCN layers is not critical. In this study, we hypothesize the main benefit arises from the features aggregation of local adjacent nodes, and **remove the nonlinear transition functions between each layers**, thus the resulting model is linear:

$$\boldsymbol{H}^{(k)} = \boldsymbol{A}_k...\boldsymbol{A}_1\boldsymbol{X}\boldsymbol{W}^{(1)}...\boldsymbol{W}^{(k)}. \tag{16}$$

The weighted parameters can be reformulated into a separate matrix to simplify the notation: $\boldsymbol{W} = \boldsymbol{W}^{(1)}\boldsymbol{W}^{(2)}...\boldsymbol{W}^{(k)}$. Thus, equation (16) can be transformed into

$$\boldsymbol{H}^{(k)} = \boldsymbol{A}_k...\boldsymbol{A}_1\boldsymbol{X}\boldsymbol{W}. \tag{17}$$

Unlike two-layer GCNs in a graph (e.g. a citation network) of nodes classification (e.g. documents) [15], in this study, we achieve our best results with either one-layer or two-layer model. After performing a **single-layer graph convolution** operation, GSAC could learn the high-level features of samples efficiently. Thus, we obtain the probability distribution matrix $\boldsymbol{Y}$, derive the optimal graph structures and weight matrix after several training rounds. Finally, GSAC can extract more representative high-level features by fusing the optimal graph structures and achieve recognition via a softmax classifier. Our single-layer GCN output can be simplified as follows:

$$\boldsymbol{H}^{(1)} = \boldsymbol{A_1XW}. \tag{18}$$

Algorithm 1 Training details of GSAC.

```
Input:
    Training data $\boldsymbol{X} \in \mathbb{R}^{N \times F}$
    Labeled data $\boldsymbol{Y}_l \in \mathbb{R}^{1 \times N_l}$
    Hyperparameters $\alpha$, $p_d$, $\beta$ and $\gamma$
Output:
    Optimal $\boldsymbol{z}_1$, $\mathcal{W}_1$, and $\boldsymbol{W}^{(1)}$

Construct a K-nearest neighbor graph
while not converges do
    Compute $\hat{\boldsymbol{x}}_i^{(1)}$ using (5)
    Compute $w_{ir}^{(1)}$ using (6)
    Compute $a_{ir}^{(1)}$ using (8)
    Compute adjacency matrix $\boldsymbol{A}_1$ using $a_{ir}^{(1)}$      ▷ Adaptive Correlation Learning
    Compute $\boldsymbol{H}^{(1)}$ using (18).                            ▷ Global Topology Optimization
    Compute loss using (21) and update the weights
end while
return $\boldsymbol{z}_1$, $\mathcal{W}_1$, and $\boldsymbol{W}^{(1)}$
```

The feature $\boldsymbol{H}^{(1)}$, derived from the graph convolutional model, serves as the input to the fully connected layer. Subsequently, the output from this layer undergoes normalization via the nonlinear softmax layer, resulting in prediction for videos classification:

$$\begin{cases} \boldsymbol{p}_i = \mathrm{softmax}(FC(\boldsymbol{h}_i^{(1)})), \\ y_i^* = \arg\max_c (p_{ic}), \end{cases} \tag{19}$$

where FC represents the fully connected layer operation, $\boldsymbol{h}_i^{(1)}$ is the vector of the i-th row of $\boldsymbol{H}^{(1)}$, and p_{ic} is the c-th element of $\boldsymbol{p}_i$, which represents the probability value that the video sample i belongs to the c-th category.

We follow the steps below to reduce the cross-entropy loss induced by the labeled data in semi-supervised action recognition tasks:

$$\mathcal{L}_2 = -\sum_{i=1}^{N_l} \sum_{c=1}^{C} y_{ic} \log p_{ic}, \tag{20}$$

where N_l represents the number of labeled data, C denotes the number of categories, y_{ic} is the c-th element of the one-hot representation of the ground truth for video sample i, and p_{ic} is the probability that video sample i belongs to the c-th category.

The overall loss function of our method is a combination of equations (9) and (20), which can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_2 + \beta \mathcal{L}_1, \tag{21}$$

As discussed in the Temporal Pyramid Network [7], a balancing coefficient β ranging from 0.005 to 0.5 is applied to the output of $\mathcal{L}_1$. $\mathcal{L}_1$ is the adaptive correlation loss in the first

Table 1: Ablation study of k-layer graph convolution on three datasets.

Method	HMDB51 $0.50 \times N$ labeled	UCF101 $0.50 \times N$ labeled	Something-Something V2 $0.50 \times N$ labeled
GSAC-nonlinear ($k = 1$)	0.5813	0.8992	**0.6051**
GSAC-nonlinear ($k = 2$)	0.5801	0.8958	0.6029
GSAC-nonlinear ($k = 3$)	0.5791	0.8941	0.6049
GSAC-linear ($k = 1$)	**0.6059**	0.9189	0.5636
GSAC-linear ($k = 2$)	0.6034	**0.9203**	0.5689
GSAC-linear ($k = 3$)	0.5992	0.9105	0.5642

Table 2: Comparison on HMDB51 dataset (average accuracy repeated by 10 times) with respect to $0.05 \times N$, $0.10 \times N$, $0.15 \times N$, $0.20 \times N$, $0.25 \times N$, and $0.50 \times N$ labeled training data.

Method	$0.05 \times N$ labeled	$0.10 \times N$ labeled	$0.15 \times N$ labeled	$0.20 \times N$ labeled	$0.25 \times N$ labeled	$0.50 \times N$ labeled
SVM	0.1095	0.2222	0.4163	0.4688	0.5144	0.5510
MFCU[43]	0.2536	0.3269	0.3921	0.4279	0.4597	0.5467
OGE-SFS[24]	0.1367	0.2318	0.4301	0.4740	0.5213	0.5561
SDMM[25]	0.2740	0.3514	0.4073	0.4502	0.4758	0.5556
GCN[15]	0.3571	0.4473	0.4955	0.5239	0.5346	0.5945
SGC[16]	0.3446	0.4262	0.4921	0.5039	0.5312	0.6034
GAT[30]	0.3482	0.4473	0.5002	0.5164	0.5367	0.5948
SiamMAST[45]	0.3593	**0.4523**	0.4973	0.5279	0.5319	0.5901
PointDMIG[46]	0.3604	0.4421	0.4829	0.5311	0.5375	0.5847
GSAC-nonlinear($k = 1$)	0.3105	0.4031	0.4575	0.5056	0.5359	0.5813
GSAC-linear($k = 1$)	**0.3647**	0.4366	**0.5020**	**0.5405**	**0.5425**	**0.6059**

Table 3: Comparison on Something-Something V2 dataset (average accuracy repeated by 10 times) with respect to $0.05 \times N$, $0.10 \times N$, $0.15 \times N$, $0.20 \times N$, $0.25 \times N$, and $0.50 \times N$ labeled training data.

Method	$0.05 \times N$ labeled	$0.10 \times N$ labeled	$0.15 \times N$ labeled	$0.20 \times N$ labeled	$0.25 \times N$ labeled	$0.50 \times N$ labeled
SVM	0.2531	0.3186	0.4013	0.4851	0.5182	0.5280
MFCU[43]	0.3520	0.4379	0.5055	0.5406	0.5583	0.5959
OGE-SFS[24]	0.2606	0.3317	0.4102	0.4887	0.5190	0.5299
SDMM[25]	0.3384	0.4150	0.4757	0.5007	0.5249	0.5735
GCN[15]	0.4094	0.5105	0.5421	0.5484	0.5633	0.5861
SGC[16]	0.4002	0.5148	0.5419	0.5557	0.5646	0.5865
GAT[30]	0.4003	**0.5077**	0.5439	0.5590	0.5647	0.5922
SiamMAST[45]	0.4089	0.5231	0.5421	0.5521	0.5647	0.5948
PointDMIG[46]	0.4018	0.5218	**0.5479**	0.5563	0.5712	0.6003
GSAC-nonlinear($k = 1$)	0.4035	0.5058	0.5332	**0.5702**	**0.5862**	**0.6051**
GSAC-linear($k = 1$)	**0.4121**	0.5074	0.5376	0.5349	0.5508	0.5636

layer. Our algorithm utilizes several parameters, including weighted vector $\boldsymbol{z}_1$, the transformation matrix $\boldsymbol{\mathcal{W}}_1$ of adaptive correlation learning module, and the learnable weight matrix $\boldsymbol{W}^{(1)}$ of first layer in graph convolution module. Algorithm 1 describes the training procedure of our method in detail.

3.4 Experimental Setup

To evaluate the effectiveness of our approach in video action recognition, we implement several methods to compare with our proposed algorithm: a linear support vector machine (SVM), MFCU [44], OGE-SFS [24], SDMM [25], GCN [15], SGC [16], GAT [30], SiamMAST [45] and PointDMIG [46]. The linear SVM is a fully supervised learning algorithm, whereas MFCU, OGE-SFS, SDMM, GCN, SGC and GAT are semisupervised learning algorithms.

Table 4: Comparison on UCF101 dataset from the THUMOS Challenge (average accuracy repeated by 10 times) with respect to $0.05 \times N$, $0.10 \times N$, $0.15 \times N$, $0.20 \times N$, $0.25 \times N$, and $0.50 \times N$ labeled training data.

Method	$0.05 \times N$ labeled	$0.10 \times N$ labeled	$0.15 \times N$ labeled	$0.20 \times N$ labeled	$0.25 \times N$ labeled	$0.50 \times N$ labeled
SVM	0.2179	0.3801	0.6921	0.7945	0.8178	0.8801
MFCU[43]	0.4481	0.6091	0.6812	0.7521	0.7969	0.8573
OGE-SFS[24]	0.2521	0.3943	0.6021	0.7289	0.8121	0.8759
SDMM[25]	0.4410	0.5802	0.6639	0.7219	0.7692	0.8531
GCN[15]	0.6521	0.7809	0.8452	0.8694	0.8891	0.9073
SGC[16]	0.6219	7638	0.8421	0.8621	0.8729	0.9102
GAT[30]	0.6281	0.7801	0.8203	0.8731	0.8845	0.9017
SiamMAST[45]	**0.6573**	0.7765	0.8421	0.8812	0.8943	0.9125
PointDMIG[46]	0.6439	**0.7843**	**0.8568**	0.8729	0.8979	0.9141
GSAC-nonlinear($\boldsymbol{k = 1}$)	0.5921	0.7451	0.8339	0.8598	0.8721	0.8992
GSAC-linear($\boldsymbol{k = 1}$)	0.6437	0.7829	0.8503	**0.8837**	**0.9010**	**0.9189**

Table 5: Average run times repeated by 10 times (in seconds).

Method	HMDB51 $0.50 \times N$ labeled	UCF101 $0.50 \times N$ labeled	Something-Something V2 $0.50 \times N$ labeled
GCN[15]	16.21s	33.87s	146.21s
SGC[16]	8.34s	15.46s	60.42s
GAT[30]	13.23s	25.21s	111.29
GSAC-nonlinear($\boldsymbol{k = 1}$)	45.23s	178.92s	721.41s
GSAC-nonlinear($\boldsymbol{k = 2}$)	97.35s	399.76s	1650.23s
GSAC-linear($\boldsymbol{k = 1}$)	$\boldsymbol{3.99s}$	$\boldsymbol{10.83s}$	$\boldsymbol{46.46s}$
GSAC-linear($\boldsymbol{k = 2}$)	6.21s	14.54s	63.81s

Due to the limited memory resources, we only use the first split of the training and testing sets. During the training phase, 30 videos, including labeled and unlabeled samples, are randomly selected from each category on three datasets. We denote N as the total number of labeled and unlabeled samples in the training set (N=1530, 3030, and 5220 for HMDB51 [47], UCF101 [48], and Something-Something V2 [49], respectively). We randomly choose n (n=5/10/15/20/25/50) percent of the training set, resulting in $0.05 \times N$, $0.10 \times N$, $0.15 \times N$, $0.20 \times N$, $0.25 \times N$ and $0.50 \times N$ randomly labeled training videos. In contrast, the numbers of remaining unlabeled training samples are $0.95 \times N$, $0.90 \times N$, $0.85 \times N$, $0.80 \times N$, $0.75 \times N$, and $0.50 \times N$, respectively. During the testing phase, we use all test samples in the original dataset, particularly the validation set of Something-Something V2. We repeat the experiment 10 times for a fair comparison since the labeled data in the training set are randomly selected.

To optimize the model, the initial learning rate α is set to 0.002, and dropout with $p_d = 0.6$ is applied to the input of each layer in the network. when calculating the loss function, we set β= 0.2 and γ= 10^3. We follow the settings from the original papers for MFCU, OGE-SFS, and SDMM parameters.

Our experimental hardware platform consists of Silver 4110 CPU, 128GB memory, and NVIDIA GeForce GTX 1080Ti. The software environment includes Ubuntu 18.04.2, Python 3.7 and PyTorch 1.11.0.

3.5 Results Analysis

In Table1, we perform an ablation study on k-layer graph convolution on three datasets. The experimental results show that the model performs best with $k = 1$ or $k = 2$, and there is little improvement in accuracy with $k \geq 3$. This is consistent with findings from most GCN literature (e.g.,GCN [15], which states: ”Best results are obtained with a 2- or 3-layer model”). In subsequent experiments, we compare results for $k = 1$ and $k = 2$ with other models. Therefore, the use of the notation $1 \ldots k$ is a standard expression for GCNs (as demonstrated in SGC [16]). The semisupervised nonlinear GCN is taken as the **baselines**. We present the average accuracy on three datasets in Table 2 to Table 4, where GSAC-linear represents the obtained model after removing the nonlinear activation function.

• Performance on Action Recognition

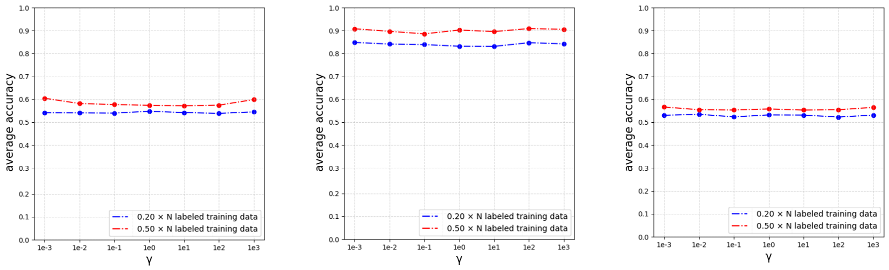
(1) As the proportion of labeled training data increases, all approaches gain improvement. This may be attributed to more labeled training data, enabling all models to learn additional action information, enhancing their recognition performance.

(2) The proposed method is superior to the compared semisupervised methods when using semisupervised learning in the experiments. This may benefit from the correlation weights by adaptive feature correlation and graph convolution to facilitate information exchange among graph nodes, which can provide global topology structures for action recognition. These results indicate that videos can be regarded as graph nodes for constructing an undirected graph. The effectiveness of semisupervised action recognition can be enhanced by considering the correlations among different videos and modeling positively correlated samples via feature aggregation.

(3) We average the accuracy of $0.05 \times N, 0.10 \times N, 0.15 \times N, 0.20 \times N, 0.25 \times N, 0.50 \times N$ cases repeated by 10 times. On the HMDB51 dataset, compared to the latest techniques SiamMAST [45] and PointDMIG [46], our GSAC-linear model improves the average accuracy by 1-2% in most cases. As the HMDB51 dataset involves movie clips or entertainment videos with fewer frames and less screen switching, making video samples easier to represent by aggregated features without nonlinear activation, thus suitable for GSAC-linear to achieve better results.

On Something-Something V2 dataset, compared to the latest techniques SiamMAST [45] and PointDMIG [46], our GSAC-nonlinear model obtains the best results in $0.20 \times N, 0.25 \times N, 0.50 \times N$ cases, and achieves comparable performance when using less labeled training data. Because Something-Something V2 videos contain numerous scene changes and frequent screen switching, significant apparent variations require stronger nonlinear feature fitting capability. More labeled training data can help our model capture the local neighborhood information between adjacent nodes, especially the global topology from labeled and unlabeled samples, which is hard to learn in this dataset. Therefore, our GSAC-nonlinear model performs better than the GSAC-linear model on this dataset.

On UCF101 dataset from the THUMOS Challenge, compared to the latest techniques SiamMAST [45] and PointDMIG [46] our GSAC-linear model achieves the best results when the labeled training data is $0.20 \times N, 0.25 \times N$ and $0.50 \times N$, but slightly underperforms other models in other scenarios. This may account for the simple changes in dataset variation which consists of sports videos with fixed camera positions, and the characteristics of many sports activities are relatively monotonous, thus leading to linear transformations for optimal graph structures without nonlinear activation. Note that the computation speed of GSAC-linear only takes 10.83 seconds to converge, as shown in Table 5. Nevertheless, GSAC-nonlinear model design constraints make it difficult to fit simple data distribution, then some recognition cases seem slightly inferior to other models but still competitive.



(a) HMDB51

(b) UCF101

(c) Something-Something V2

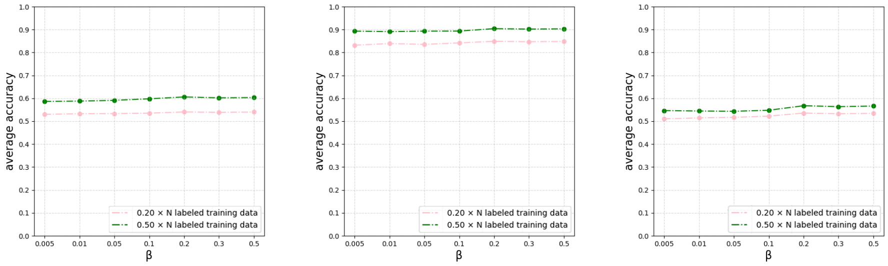
Fig. 3: Sensitivity analysis of the parameter γ on three datasets.

Hence, compared to the latest techniques like SiamMAST [45] and PointDMIG [46], our method demonstrates clear advantages. While SiamMAST effectively integrates multiple features, its fixed RGB frame averaging approach may limit generalization. PointDMIG retains spatial structure information and models long-term spatiotemporal correlations through complex spatiotemporal encoding, but this leads to spatial information loss and excess computation. Our GSAC model addresses these shortcomings with adaptive feature correlation and linear transformations. The GSAC-linear model excels in action videos classification with simple background due to simple unstructured data, while the GSAC-nonlinear model performs better in complex scenarios. GSAC leverages local neighborhood relationships and global topology structures between video nodes, adapting to nearest-neighbor relationships between graph nodes. Therefore, our method surpasses existing global topology modeling and feature extraction techniques, achieving better performance and generalization.

• Computation Speed

Before removing the nonlinear activation functions, each layer calculates complex node relationships and updates features. This results in a computational complexity of $O(n^3)$ due to the multiplication of feature dimensions and adjacency matrices. Nonlinear activation functions typically add to this complexity by performing additional nonlinear operations on each node's features. After removing the nonlinear functions, the model's computations involve only matrix multiplications, reducing the time complexity to $O(n^2)$.

We also set a practical experiment to evaluate the computation speed of related GCN-based methods in Table 5. The comparison results of the model are presented, with or without nonlinear activation function. The best results are highlighted separately in bold. We consider the case of $0.50 \times N$ labeled samples for HMDB51, UCF101, and Something-Something V2, then compute the average run time over the standard splits. Our GSAC-linear achieves the fastest speed because of single-layer graph convolution. Compared with GCN, SGC, GAT, and GSAC-nonlinear, the run time of GSAC-linear gains 4.06×, 2.09×, 3.31×, 24.34× faster on HMDB51, 3.12×, 1.42×, 2.32×, 36.9× faster on UCF101, and 3.14×, 1.30×, 2.39×, 35.51× faster on Something-Something V2, respectively, while our GSAC-linear method achieves the best recognition performance on both HMDB51 and UCF101.



(a) HMDB51 (b) UCF101 (c) Something-Something V2

Fig. 4: Sensitivity analysis of the parameter β on three datasets.

• Parameter Sensitivity Study

Two hyperparameters, i.e., regularization parameter γ and balance parameter β are mainly discussed. To explore how they affect the analysis performance and iteration process in action recognition, we conduct parameter sensitivity experiments.

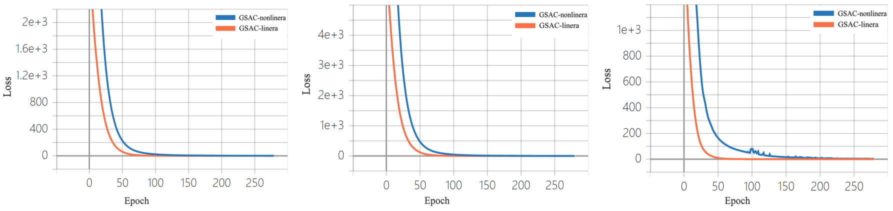
For the parameter γ in equation (9), it serves to control the connection probability. Specifically, it affects similarity measurement as well as the connection probability between samples. When calculating the connection probability between samples, the parameter γ can regulate the connection strength between samples by adjusting the regularization term, thus affecting the final clustering result. The optimal value of γ will be determined adaptively according to the number of neighbors needed. In this paper, γ is used to regulate the correlation between nodes to prevent simply assigning the nearest node of the center node with a correlation weight of 1 and the other nodes with a correlation weight of 0. Using $0.20 \times Nand0.50 \times N$ labeled training data, we set γ in the range of 10^{-3} to 10^{3}. Fig.3 shows that different γ values correspond to slight changes in the iterative process, which indicates that GSAC is robust to the γ parameter's variation.

For the parameter β in equation (21), a larger β implies that a larger proportion of adaptive correlations are considered and vice versa. When $\beta = 0$, adaptive correlations are not included, we show the results of β-parameter sensitivity in Fig.4. It can be observed that, in the case of $0.20 \times Nand0.50 \times N$ labeled training data, as β varies from 0.005 to 0.5, the accuracy correspondingly increases, reaching its peak at β = 0.2. This implies that the model makes good use of unlabeled samples for training, and better adapts to the characteristics of data distribution. A reasonable value of β facilitates the mining of correlations among multiple nodes, further improving the performance of our proposed semi-supervised approach.

When all hyperparameters are chosen within a certain range, e.g., γ ranges from $\{10^{-3} \sim 10^{3}\}$ and β ranges from $\{0.005 \sim 0.5\}$, a stable and high accuracy can be obtained. In other words, there is flexibility in choosing parameter values in order to get the best performance.

• Convergence Study

Finally, we conduct experiments on three datasets and obtain corresponding convergence curves to investigate the proposed technique. The number of labeled training samples is set to $0.50 \times N$ for each dataset, whereas the remaining samples (i.e., $0.50 \times N$) are considered



(a) HMDB51 (b) UCF101 (c) Something-Something V2

Fig. 5: Convergence curves of loss values in (21) when using our algorithm on three datasets.

as unlabeled training data. The results in Fig.5 demonstrate that the loss function value converges after several epochs. We notice that GSAC-nonlinear requires more epochs to achieve convergence since a nonlinear activation function such as ReLU is involved in global topology optimization, especially the video data distribution of Something-Something V2 is more complex than other datasets.

4 Conclusions

Semisupervised action recognition can reduce the high cost of manual annotation. In this study, we consider each video as a node in an undirected graph, explore the feature correlation among different videos, and present a semisupervised learning framework based on adaptive correlation learning and global topology optimization. The adaptive correlation learning module uses local neighborhood information to assign different weights to adjacent nodes with feature aggregation. Graph convolution is employed to provide more representative features for each node, and the aggregated high-level features are fed into the subsequent layers for video action recognition. According to the experimental results, the significant efficiency advantage of our model makes it an ideal choice for handling large-scale data and practical applications. However, the proposed method may be slightly inferior to some models on certain datasets.

In future work, we will continue to specifically optimize the model to enhance its generalization performance while maintaining its high efficiency. We will also consider the dynamic neighborhood structure, which we expect to be able to learn the best neighborhood structure given that the neighbors of each central node may change during model training, thereby leading to better performance.

5 Statements and Declarations

5.1 Funding

This research was supported by the Guangxi Natural Science Foundation (2024GXNSFAA010493), the National Nature Science Foundation of China (61862015), the Science and Technology Project of Guangxi (AD23023002,AD21220114), the Guangxi Key Research and Development Program (AB17195025).

5.2 Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

5.3 Authors contribution statement

Authors Contribution: Conceptualization, Z.X.; methodology, F.W., J.C. and Z.X.; software, F.W. and J.C., Z.X.; validation, F.W. and J.C.; formal analysis, F.W., J.C. and Z.X.; investigation, F.W., J.C. and Z.X.; resources, Z.X. and R.H.; data curation, Z.X.; writing—original draft, F.W. and J.C.; writing—review and editing, F.W. and Z.X.; visualization, F.W. and Z.X.; supervision, Z.X. and R.H.; project administration, Z.X.; funding acquisition, Z.X. All authors have read and agreed to the published version of the manuscript.

5.4 Informed Consent Statement

Not applicable

5.5 Competing Interests

The authors declare no conflict of interest.

5.6 Data availability and access

Data will be made available on reasonable request.

References

[1] Lu, F., Liu, G.: Image retrieval by aggregating deep orientation structure features. International Journal of Machine Learning and Cybernetic, 1–14 (2024) https://doi.org/10.1007/s13042-024-02172-w

[2] Mi, J., Luo, J., Zhao, H.: Improved dense residual network with the coordinate and pixel attention mechanisms for helmet detection. International Journal of Machine Learning and Cybernetics, 1–17 (2024) https://doi.org/10.1007/s13042-024-02205-4

[3] Lin, H., Cai, D., Xu, Z.: Fabric4show: real-time vision system for fabric defect detection and post-processing. Visual Intelligence **2**(1), 13 (2024) https://doi.org/10.1007/s44267-024-00047-w

[4] Sun, H., Li, B., Dan, Z.: Multi-level feature interaction and efficient non-local information enhanced channel attention for image dehazing. Neural Networks **163**, 10–27 (2023) https://doi.org/10.1016/j.neunet.2023.03.017

[5] Sun, H., Lou, z., Ren, D.: Partial siamese with multiscale bi-codec networks for remote sensing image haze removal. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–16 (2023) https://doi.org/10.1109/TGRS.2023.3321307

[6] Chen, Y., Kalantidis, Y., Li, J., Yan, S., Feng, J.: Multi-fiber Networks for Video Recognition. Paper presented at the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018 (2018)

[7] Yang, C., Xu, Y., Shi, J., Dai, B.: Temporal pyramid network for action recognition, (2020). Paper presented at the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),Seattle, WA, USA, 13–19 June 2020

[8] Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient and scalable video understanding on edge devices, vol. 44, pp. 2760–2774 (2020). https://doi.org/10.1109/TPAMI.2020.3029799

[9] Feichtenhofer, C.: X3D: Expanding Architectures for Efficient Video Recognition. Paper presented at the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020 (2020)

[10] Li, Y., Ji, B., Shi, X., Zhang, J., Kang, B., Wang, L.: TEA: Temporal Excitation and Aggregation for Action Recognition. Paper presented at the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020 (2020)

[11] Sudhakaran, S., Escalera, S., Lanz, O.: Gate-Shift Networks for Video Action Recognition. Paper presented at the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020 (2020)

[12] Wang, L., Tong, Z., Ji, B., Wu, G.: TDN: Temporal Difference Networks for Efficient Action Recognition. Paper presented at the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20-25 June 2021 (2021)

[13] Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**, 221–231 (2013) https://doi.org/10.1109/TPAMI.2012.59

[14] Simonyan, K., Zisserman, A.: Two-Stream Convolutional Networks for Action Recognition in Videos. Paper presented at the 2014 Neural Information Processing Systems(NeurIPS), Montréal, Canada, 8–12 December 2014 (2014)

[15] Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. Paper presented at the 2017 International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017 (2017)

[16] Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Handbook of Experimental Pharmacology, vol. 97, pp. 6861–6871. PMLR, Long Beach (2019)

[17] Liu, H., Liu, T., Chen, Y., Zhang, Z., Li, Y.F.: Ehpe: Skeleton cues-based gaussian coordinate encoding for efficient human pose estimation. IEEE Transactions on Multimedia, 1–12 (2022) https://doi.org/10.1109/TMM.2022.3197364

[18] Liu, H., Chen, Y., Zhao, W., Zhang, Z.: Human pose recognition via adaptive distribution encoding for action perception in the self-regulated learning process. Infrared Physics & Technology **114**, 103660 (2021)

[19] Liu, T., H, L., B, Y., Z, Z.: Ldcnet: Limb direction cues-aware network for flexible human pose estimation in industrial behavioral biometrics systems. IEEE Transactions on Industrial Informatics, 1–11 (2023) https://doi.org/10.1109/TII.2023.3266366

[20] Chang, X., Y., Y.: Semisupervised feature analysis by mining correlations among multiple tasks. 2016 IEEE transactions on neural networks and learning systems **28**(10), 2294–2305 (2016) https://doi.org/10.1109/TNNLS.2016.2582746

[21] Gao, Y., J., M., L., Y.A.: Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. 2017 IEEE Transactions on Image Processing **26**(5), 2545–2560 (2017)

[22] Ma, J., J., J., Li Y, .: Feature guided gaussian mixture model with semi-supervised em and local geometric constraint for retinal image registration. Information Sciences **417**, 128–142 (2017)

[23] Ma, J., W., Y., P., L.: A generative adversarial network for infrared and visible image fusion. Information fusion **48**, 11–26 (2019)

[24] Luo, M., Chang, X., Nie, L., Yang, Y., Hauptmann, A.G., Zheng, Q.: An adaptive semisupervised feature analysis for video semantic recognition. IEEE Transactions on Cybernetics **48**(2), 648–660 (2018) https://doi.org/10.1109/TCYB.2017.2647904

[25] Xu, Z., Hu, R., Chen, J., Chen, C., Jiang, J., Li, J., Li, H.: Semisupervised discriminant multimanifold analysis for action recognition. IEEE Transactions on Neural Networks and Learning Systems **30**(10), 2951–2962 (2019) https://doi.org/10.1109/TNNLS.2018.2886008

[26] Si, C., Nie, X., Wang, W., Wang, L., Tan, T., Feng, J.: Adversarial Self-supervised Learning for Semi-supervised 3D Action Recognition. Paper presented at the 2020 European Conference on Computer Vision(ECCV), Glasgow, UK, 23–28 August 2020 (2020)

[27] Khan, H., Liu, H., Liu, C.: Missing label imputation through inception-based semi-supervised ensemble learning. Advances in Computational Intelligence **2**(1), 10 (2022) https://doi.org/10.1007/s43674-021-00015-7

[28] Bi, H., Perello-Nieto, M., Santos-Rodriguez, R.: An active semi-supervised deep learning model for human activity recognition. Journal of Ambient Intelligence and Humanized Computing **14**, 13049–13065 (2023) https://doi.org/10.1007/s12652-022-03768-2

[29] Zhang, J., Han, Y., J., T.: Semi-supervised image-to-video adaptation for video action recognition. IEEE transactions on cybernetics **47**(4), 960–973 (2017) https://doi.org/10.1109/TCYB.2016.2535122

[30] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. Paper presented at the 2018 International Conference on Learning Representations (ICLR), Vancouver Convention Center, Vancouver, Apr 30th through May 3rd 2018 (2018)

[31] Jiang, B., Zhang, Z., Lin, D., Tang, J., Luo, B.: Semi-Supervised Learning With Graph Learning-Convolutional Networks. Paper presented at the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, 15–20 June 2019 (2019)

[32] Han, W., Wen, C. C. Wang, Li, X., Li, Q.: Point2node: Correlation learning of dynamic-node for point cloud feature modeling. Paper presented at the thirty-fourth AAAI Conference on Artificial Intelligence, New York Hilton Midtown, New York, 7–12 February 2020 (2020)

[33] Ma, L., Li, X., Shi, Y., Wu, J., Zhang, Y.: Correlation filtering-based hashing for fine-grained image retrieval. IEEE Signal Processing Letters **27**, 2129–2133 (2020) https://doi.org/10.1109/LSP.2020.3039755

[34] Ying, L., Qian Nan, Z., Fu Ping, W.: Adaptive weights learning in cnn feature fusion for crime scene investigation image classification. Connection Science **33**(3), 719–734 (2021) https://doi.org/10.1080/09540091.2021.1875987

[35] Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Paper presented at 2017 Advances in Neural Information Processing Systems(NIPS), Long Beach, United States, 4–7 December 2017 (2017)

[36] Thakkar, K., Narayanan, P.J.: Part-based graph convolutional network for action recognition. (2018). Paper presented at the 3rd international symposium on the genetics of industrial microorganisms, University of Wisconsin, Madison, 13–19 June 2020

[37] Zeng, R., Huang, W., Gan, C., Tan, M., Rong, Y., Zhao, P., Huang, J.: Graph Convolutional Networks for Temporal Action Localization. Paper presented at 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea, October 27 to November 2 2019 (2019)

[38] Manessi, F., Rozza, A., Manzo, M.: Dynamic graph convolutional networks. Pattern Recognition **97**, 107000 (2020) https://doi.org/10.1016/j.patcog.2019.107000

[39] Sofianos, T., Sampieri, A., Franco, L.: Space-time-separable graph convolutional network for pose forecasting. Paper presented at the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021 (2021)

[40] Qiu ZX, D.W. Zhang HB: Effective skeleton topology and semantics-guided adaptive graph convolution network for action recognition. The Visual Computer, 1–13 (2022) https://doi.org/10.1007/s001090000086

[41] Gan, J., Hu, R., Mo, Y.: Multigraph fusion for dynamic graph convolutional network. IEEE Transactions on Neural Networks and Learning Systems, 1–12 (2022) https://doi.org/10.1109/TNNLS.2022.3172588

[42] Abeywickrama, T., Cheema, M.A., Taniar, D.: K-nearest neighbors on road networks: a journey in experimentation and in-memory implementation, vol. abs/1601.01549 (2016). http://arxiv.org/abs/1601.01549

[43] Nie, F., Zhu, W., Li, X.: Structured graph optimization for unsupervised feature selection. IEEE Transactions on Knowledge and Data Engineering **33**(3), 1210–1222 (2021) https://doi.org/10.1109/TKDE.2019.2937924

[44] Wang, S., Ma, Z., Yang, Y., Li, X., Pang, C., Hauptmann, A.G.: Semi-supervised multiple feature analysis for action recognition. IEEE Transactions on Multimedia **16**(2), 289–298 (2014) https://doi.org/10.1109/TMM.2013.2293060

[45] Xuemin, L., Wei, Q., Reformat, M., Haiquan, Z., Jim.X, C.: Siammast: Siamese motion-aware spatio-temporal network for video action recognition. The Visual Computer **40**(5), 3163–3181 (2023) https://doi.org/10.1007/s00371-023-03018-2

[46] Yao, D., Zhenjie, H., Jiuzhen, L., Kaijun, Y., Xinwen, Z.: Pointdmig: a dynamic motion-informed graph neural network for 3d action recognition. Multimedia Systems **30**(4), 192 (2024) https://doi.org/10.1007/s00530-024-01395-9

[47] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: A large video database for human motion recognition. In: 2011 International Conference on Computer Vision, pp. 2556–2563 (2011)

[48] Soomro, K., Roshan Zamir, A., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. CoRR **abs/1212.0402** (2012)

[49] Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., Memisevic, R.: The “something something” video database for learning and evaluating visual common sense. In: 2017 IEEE International Conference on Computer Vision, pp. 5843–5851 (2017)