Condition-Adaptive Graph Convolution Learning for Skeleton-Based Gait Recognition

Xiaohu Huang, Xinggang Wang, *Member, IEEE*, Zhidianqiu Jin, Bo Yang, Botao He, Bin Feng, and Wenyu Liu, *Senior Member, IEEE*,

Abstract—Graph convolutional networks have been widely applied in skeleton-based gait recognition. A key challenge in this task is to distinguish the individual walking styles of different subjects across various views. Existing state-of-the-art methods employ uniform convolutions to extract features from diverse sequences and ignore the effects of viewpoint changes. To overcome these limitations, we propose a condition-adaptive graph (CAG) convolution network that can dynamically adapt to the specific attributes of each skeleton sequence and the corresponding view angle. In contrast to using fixed weights for all joints and sequences, we introduce a joint-specific filter learning (JSFL) module in the CAG method, which produces sequenceadaptive filters at the joint level. The adaptive filters capture fine-grained patterns that are unique to each joint, enabling the extraction of diverse spatial-temporal information about body parts. Additionally, we design a view-adaptive topology learning (VATL) module that generates adaptive graph topologies. These graph topologies are used to correlate the joints adaptively according to the specific view conditions. Thus, CAG can simultaneously adjust to various walking styles and viewpoints. Experiments on the two most widely used datasets (i.e., CASIA-B and OU-MVLP) show that CAG surpasses all previous skeletonbased methods. Moreover, the recognition performance can be enhanced by simply combining CAG with appearance-based methods, demonstrating the ability of CAG to provide useful complementary information. The source code will be available at https://github.com/OliverHxh/CAG

Index Terms—Skeleton-based Gait Recognition, Graph Convolution, Adaptive Feature Learning.

I. INTRODUCTION

AIT recognition is an important biometric technology with various applications ranging from case detection to human-robot interaction. The main idea is to identify a person by his/her distinctive walking style. The gait recognition methods can be classified into two categories; appearance-based [1]–[5] and model-based [6]–[11]. Since the appearance-based methods are more sensitive to the appearance variations, model-based methods have gained more attention recently.

Xiaohu Huang, Xinggang Wang, Zhidianqiu Jin, Bin Feng and Wenyu Liu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. Bo Yang and Botao He are with Wuhan FiberHome Digital Technology Co., Ltd.

- X. Huang(e-mail:huangxiaohu@hust.edu.cn).
- X. Wang(e-mail:xgwang@hust.edu.cn).
- $Z. \ Jin (e\text{-}mail\text{:}jzdq@hust.edu.cn).$
- B. Yang(email: byang@fhzz.com.cn)
- B. He(e-mail:hebotao@fhzz.com.cn)
- B. Feng (e-mail: fengbin@hust.edu.cn). Corresponding author.
- W. Liu(e-mail: liuwy@hust.edu.cn).

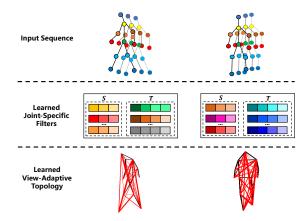


Fig. 1: An overview of the proposed idea. First row: CAG takes gait-skeleton sequences as inputs. Second row: CAG automatically generates joint-specific filters for each sequence in both the S (spatial) and T (temporal) domains; thus, it can capture personalized walking styles and extract fine-grained patterns. Third row: CAG dynamically learns a view-adaptive topology for each sequence to handle customized gait characteristics across different camera views (Best viewed in color).

Among the various model-based methods, the skeleton representation is the most popular because it can be easily used to extract features, and it is consistent with the human body structure.

Graph convolutional networks (GCNs) have been widely applied to achieve impressive results in skeleton-based gait recognition [7], [8], [12]–[15] since they can model inherent correlations between joints. These methods have used standard weight-sharing convolutions to extract the spatial and temporal features of each joint from the sequences. However, personalized gait characteristics exhibit complex patterns of joints. Thus, such uniform feature extraction can only capture a general human walking style but cannot adapt to individual walking styles. Considering that gait is a fine-grained motion pattern, it is essential to distinguish a personalized walking style among different individuals. Therefore, the fixed filters used in the above-mentioned methods limit the flexible and robust modeling ability.

Moreover, the used 2D skeleton structure shows different characteristics for different camera views, making the corresponding topological correlations diverse. Also, current gait methods [7], [8], [12] employ predefined graph topologies, which may not be suitable for all views. Some action recogni-

tion methods [16]–[18] proposed adaptive graphs by learning the correlations between joints dynamically. However, such adaptive graphs were not designed to fit cross-view scenarios. Therefore, current skeleton-based methods do not offer explicit solutions to viewpoint variations. As a consequence, the recognition performance is hindered.

To tackle the above issues, we propose a novel GCN for skeleton-based gait recognition, called condition-adaptive graph (CAG) convolution network. The main idea of CAG is to adapt graph convolution learning to suit the variations in personalized walking styles and viewing conditions. As shown in Fig. 1, in CAG, filters are automatically learned to capture personalized walking characteristics using a joint-specific filter learning (JSFL) module, and graphs that can handle viewpoint variations are generated using a view-adaptive topology learning (VATL) module. These dynamic filters are used to extract fine-grained spatial-temporal patterns of each joint, and the graphs are employed to correlate the joints adaptively, based on the specific viewing condition. Therefore, the JSFL and VATL modules can be seamlessly integrated together in the proposed network.

Specifically, the JSFL module produces filters by encoding joint-level features across the entire sequence. In particular, since different joints correspond to different body parts with various patterns, the network can exploit joint-level feature mining to obtain fine-grained information. The JSFL module constructs two branches, which correspond to model spatial configuration and temporal motion, respectively. This architecture enables the spatial and temporal filters to learn separately. Considering the computational efficiency, all filters are learned in a depth-wise manner.

The VATL module generates view-adaptive topologies by using prior-view knowledge. Specifically, VATL transforms the general **fixed view-invariant** topology into a set of **learnable view-related** topologies. It then constructs the view-adaptive topology with the following three components: (1) A topology, which is the most appropriate for the sequence view. (2) A topology, which is a weighted summation of all learnable view-related topologies, enhances its robustness by utilizing the intrinsic correlation of different views. (3) A fixed topology, which represents prior knowledge about the human body structure and has been shown effective in human action recognition [17], [19], [20]. In this way, the learned view-adaptive topology considers specific viewing conditions and incorporates general knowledge about the human body structure.

In summary, the main contributions of this paper include the following three aspects:

- A JSFL module that dynamically generates joint-specific filters tailored to sequence characteristics. In this way, graph convolutions can adapt to personalized walking styles and detailed spatial-temporal patterns can be extracted from each sequence.
- (2) A VATL module that generates a view-adaptive topology, based on specific viewing conditions in each sequence. In this way, graph convolutions can handle the view variations.

(3) A condition-adaptive graph (CAG) convolutional network is proposed by integrating the JSFL and VATL modules. Extensive experiments conducted on CASIA-B [21] and OU-MVLP [11] datasets demonstrate the state-of-the-art performance of CAG. By combining CAG with appearance-based methods, the recognition performance can be effectively improved.

II. RELATED RESEARCH WORK

A. Gait Recognition

Currently, two categories of mainstream gait recognition methods are available; the **appearance-based** and the **model-based** methods. The **appearance-based** approaches obtain silhouettes as inputs, which rely on abundant shape information to model spatial-temporal features.

Some of the representative appearance-based methods are disentanglement-based, set-based, part-based, and 3D convolutional neural networks (CNNs)-based. The disentanglement-based methods [22]–[24] aimed to disentangle the original walking features into identity-relevant features and identity-irrelevant features, which avoided the negative effects of confounding variables. The set-based approaches [1], [25] regarded a gait sequence as an unordered set, which processed each frame independently, and did not explicitly model temporal relations. Further, the part-based methods [2], [3], [26], [27] proposed to extract features of different parts individually for fine-grained feature extraction, and applied temporal motion modeling in different scales. 3D CNN-based methods [4], [5], [28], [29] stacked layers of 3D convolutions to capture spatial-temporal patterns in multiple scales.

The **model-based** approaches methods model the human structure and body movement by designing simulated models [30], [31] or using skeletons [6]–[8], [11] as inputs. Recently, due to the successful development of pose estimation methods [32]–[34], the skeleton-based methods have prevailed.

The PoseGait [6], CNN-pose [11], and pose-based temporal–spatial network (PTSN) [35] methods used a skeleton sequence as a 2D matrix and employ 2D CNNs or LSTMs to model gait features. These methods did not consider the topological connections of the skeletons. The JointsGait [8], Mao et.al. [12], GaitGraph [7], MSGG [13], CycleGait [14], Gait-D [36] adopted GCN-based architectures from skeleton-based action recognition [19], [37]. Recently, a transformer-based method [38] adopted transformer blocks to model the spatial and temporal correlations in a self-attention manner. Furthermore, the ModelGait [9] and Li et.al. [10] methods used a human mesh-recovery (HMR) [39] network to extract and use both shape and pose features.

The proposed CAG belongs to the **skeleton-based** methods and utilizes a GCN-based network architecture.

B. GCNs for Skeleton Modeling

In recent years, numerous GCNs have been adopted to model spatial-temporal features in skeleton-based video analysis domains, especially in skeleton-based action recognition. Most current GCNs follow the pipeline design of ST-GCN [19]. For skeleton-based methods using GCNs (MSGG [13],

CycleGait [14], GAITTAKE [15], Gait-D [36], GaitGraph [7], and JointsGait [8]), they process different sequences with the same network parameters in GCNs, therefore limiting the model capacity to extract sample-specific characteristics. On the contrary, the proposed JSFL module learns various filters for different sequences and joints, which benefit extracting personalized walking features. Besides, MSGG [13] and GAITTAKE [15] adopt temporal attention approaches, which improve temporal aggregation flexibility. However, this adaptive manner is limited in the temporal domain and only used for feature aggregation, which does not play the main role in feature extraction, while our JSFL module is applied in both spatial and temporal domains, and used for feature extraction in the GCN backbone. A Transformer-based method (Gait-TR [38]) uses Transformer blocks to dynamically learn spatial gait patterns, but its temporal learning parameters are still shared for different samples, which is not flexible.

Some dynamic GCNs [16]–[18] were proposed to learn joint correlations dynamically in order to relax the fixed topology constraints and enrich the global context. However, these methods were not designed to extract fine-grained features, and their graphs were not generated to relate explicitly to viewing conditions, which is crucial for gait recognition. In contrast, the proposed VATL module employs learning of adaptive topologies, explicitly based on viewing conditions.

C. Adaptive Mechanisms

Data-dependent mechanisms have achieved great success in computer vision, which adjust feature extractions to capture instance-specific properties. SE-Net [40] connected the relations among different channels to adaptively attend to the most important ones. Self-attention methods [41]–[43] utilized QKV-based techniques to effectively construct the global context. Further, inspired by the attention ideas, the methods reported in [44]–[46] generated dynamic weights to combine a set of filters in order to promote the network representation capacity. Recently, lightweight networks [47]–[50] produced convolutional filters on-the-fly, which adaptively fit the customized features.

For appearance-based methods (GaitPart [2] and Meta-Gait [51]) using attention mechanisms, their approaches are just supplements to the uniform feature extraction of their backbones. In contrast, the proposed JSFL module achieves dynamic feature extraction by generating adaptive convolutional filters, which no longer require an attention mechanism. Besides, the part-level feature learning in appearance-based methods is achieved by a manual partition, where the part semantics are not well aligned. In contrast, JSFL can obtain better-aligned parts from the skeleton inputs.

Previously, a few appearance-based gait approaches (MGAN [52], GaitGAN [53], Chai et.al. [54], and Vi-GaitGL [55]) have studied the topic of learning view-invariant gait features. MGAN [52] and GaitGAN [53], and Makihara et.al [56] transform gait energy images (GEIs), period energy images (PEIs) or silhouettes from arbitrary views into a targeted view, which however is not feasible for skeleton-based gait recognition. Chai et.al. [54] and Vi-GaitGL [55] propose to

learn view-specific embedding or projection parameters for the fully-connected layers. In contrast, the proposed viewadaptive topology learning (VATL) aims to generate viewadaptive topologies for GCNs.

III. PROPOSED METHOD

In this section, we initially review the preliminary concepts of GCN. Then, we describe the proposed network architecture and provide details of the proposed modules.

A. Preliminary Concepts

Notations. A human skeleton is denoted as a topology graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the vertex set \mathcal{V} denotes body joints and the edge set \mathcal{E} denotes bones. The vertex set is represented as $\mathcal{V} = \{v_1, v_2, ..., v_N\}$, where N denotes the number of vertices. The edge set \mathcal{E} is formulated as an adjacent matrix $A \in \mathbb{R}^{N \times N}$, where each element $a_{i,j}$ is defined as the connection strength between vertices v_i and v_j . We formulate a skeleton sequence of T frames as $I \in \mathbb{R}^{T \times N \times C_{in}}$, where C_{in} denotes the input-channel dimension. For each input skeleton, we apply a batch normalization layer to normalize the joint coordinates before feeding them into the network. The features of the vertex set of T frames are formulated as $X \in \mathbb{R}^{T \times N \times C}$, where C denotes the channel dimension of each vertex.

Graph Convolution. Let $X_S \in \mathbb{R}^{T \times N \times C'}$ be the output features after performing spatial configuration extraction, where C' denotes the output-channel dimension. In this way, the general graph convolution described in [19] follows the formulation given below:

$$X_{S} = \sum_{k=1}^{K_{S}} A^{k} X W_{S}^{k}, \tag{1}$$

where K_S denotes the kernel size of the spatial domain (e.g., 3 in ST-GCN [19]), and $W_S \in \mathbb{R}^{K_S \times C \times C'}$ denotes the feature transformation filter in the spatial domain. The adjacent matrix $A^k \in \mathbb{R}^{N \times N}$ enables GCN to aggregate the information about vertices in a spatial context, which captures the human architecture configuration.

After the application of spatial configuration extraction, a kernel size of K_T temporal convolution is employed by W_T to model the temporal dynamics; the output X_T with a temporal dimension T' is obtained as follows:

$$X_T = Conv(X_S, W_T), (2)$$

where $W_T \in \mathbb{R}^{K_T \times C' \times C'}$, and $X_T \in \mathbb{R}^{T' \times N \times C'}$.

B. Network Architecture

The two-branch architecture of CAG is illustrated in Fig. 2, where a human skeleton sequence I is taken as an input. In the first branch, based on the viewing conditions and sequence characteristics, VATL dynamically constructs a view-adaptive topology G_{VA} . The constructed G_{VA} is utilized to guide a joint aggregation of graph convolutions in the second branch. Also, a cross-entropy loss L_{CE}^{view} is employed to supervise view-related feature learning.

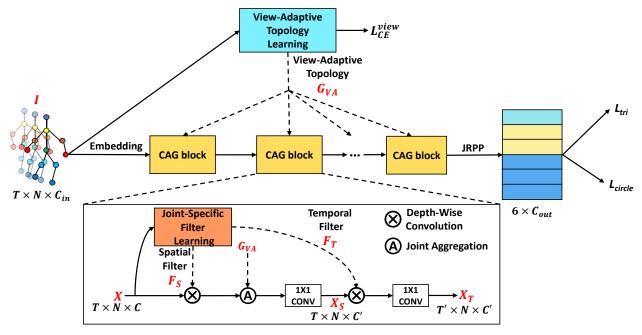


Fig. 2: Illustration of CAG. A gait skeleton sequence of T frames is taken as an input. The input sequence is passed through two branches; the first branch employs the VATL module to produce a view-adaptive topology for adaptation to camera viewpoints; the second branch extracts customized features using a series of CAG blocks. In each block, the adaptive joint-specific filters are dynamically generated by the JSFL module. Joints relationship pyramid mapping (JRPP) [8] is used to map the gait skeleton features into 6 scales. C_{in} , C_{out} , L_{CE} , L_{tri} and L_{circle} denote the input-channel number, the output-channel number, the cross-entropy loss, the triplet loss [57] and the circle loss [58], respectively. See text for more details.

In the second branch, coarse skeleton features are initially extracted using a lightweight embedding module, i.e., an ordinary GCN block, which is described in Eq. (1). Then, CAG blocks are stacked to refine the features and extract customized clues. In each block, the JSFL module is used to automatically generate filters (F_S and F_T) in a depth-wise manner. In particular, F_S combined with G_{VA} from VATL is used for spatial configuration extraction, and F_T is used for temporal modeling. Two 1×1 convolutions are employed to fuse the cross-channel information.

Consequently, the GCN operations in Eq. (1) and Eq. (2) can be rewritten as follows:

$$X_S = Conv_{1\times 1}(\sum_{k=1}^{K_S} G_{VA}^k(X \otimes F_S^k), W_1),$$

$$X_T = Conv_{1\times 1}(X_S \otimes F_T, W_2),$$
(3)

where \otimes denotes the depth-wise convolution, $G^k_{VA} \in \mathbb{R}^{N \times N}$, $F^k_S \in \mathbb{R}^{N \times C}$, $F_T \in \mathbb{R}^{K_T \times N \times C'}$, $W_1 \in \mathbb{R}^{C \times C'}$ and $W_2 \in \mathbb{R}^{C' \times C'}$.

After processing the CAG blocks, a JRPP [8] module is used to map the gait features into 6 scales based on the joint relationship of human architecture. Finally, a triplet loss [57] L_{tri} and a circle loss [58] L_{circle} are applied on the output features to perform training supervision. The overall loss function is summarized as follows:

$$L = \lambda_1 L_{tri} + \lambda_2 L_{circle} + \lambda_3 L_{CE}^{view}, \tag{4}$$

where λ_1 , λ_2 and λ_3 are the hyperparameters to balance the respective loss functions.

Inspired by the two-stream works [20], [59], where fusing joint and bone features enable networks to recognize human activities, we employ two separate streams for individually extracting joint and bone features; the objective is to combine the merits of both. The pipeline of the bone-feature stream is the same as that of the joint-feature stream, except that its input is the subtraction of coordinates in adjacent joints. The joint-feature stream can be regarded as the first-order information, and the bone-feature stream can be regarded as the second-order information. For simplicity, we only illustrate the joint-feature stream in Fig. 2. Features from the two streams are concatenated with a size of $12 \times C_{out}$ and used as an output.

C. Joint-Specific Filter Learning

Since different body parts typically exhibit different amounts of variation and degrees of freedom due to the articulated structure of the skeleton, the JSFL module is used to describe the individual spatial-temporal characteristics flexibly in different gait sequences by generating customized filters. As shown in Fig. 3, two separate branches corresponding to spatial and temporal filter generations are utilized to extract the spatial configurations and capture the temporal dynamics, respectively. Particularly, the filters are generated in a depthwise manner to increase efficiency.

The gait features $X \in \mathbb{R}^{T \times N \times C}$ are used as an input and a temporal adaptive pooling is applied to obtain a temporal downsampled output $X_P \in \mathbb{R}^{T_P \times N \times C}$, where T_P denotes the pooled size. For both spatial and temporal branches, we initially utilize temporal convolutions to learn the contextual

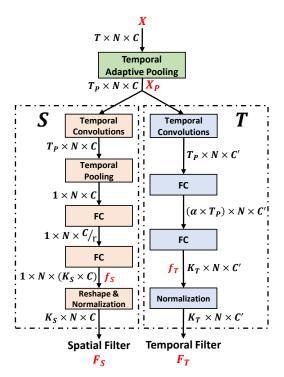


Fig. 3: JSFL constructs two separate branches to generate S (spatial) and T (temporal) joint-specific filters, respectively.

information at each joint. Then, we apply a temporal pooling (TP) operation to the spatial branch to aggregate the temporal global context. Next, two cascaded fully-connected layers with a batch normalization layer and a rectified linear unit (ReLU) activation function are used to construct the cross-channel communications and produce filter f_s with the expected size $1 \times N \times (K_S \times C)$. Subsequently, we reshape the filter into a size $K_S \times N \times C$, and adopt batch normalization to avoid filter parameters being extremely large or small. In summary, the operations in the spatial branch can be formulated as follows:

$$f_S = \mathcal{F}(\mathcal{F}(TP(TC(X_P)), W_3), W_4),$$

$$F_S = BN(Reshape(f_s)),$$
(5)

where TC, TP, \mathcal{F} and BN denote temporal convolution, temporal pooling, fully-connected layer, and batch normalization respectively. $W_3 \in \mathbb{R}^{C \times \frac{C}{r}}$ reduces the channel dimension by the ratio r and $W_4 \in \mathbb{R}^{\frac{C}{r} \times C}$ recovers the channel dimension.

Different from the spatial branch, the temporal branch is used to describe motion characteristics, which does not include TP for maintaining the temporal structure, and uses two cascaded fully-connected layers with a ReLU activation function along the temporal dimension. The objective is to effectively exploit rich temporal relations in different moments in order to explore motion properties. Next, a normalization operation is applied to ensure parameter distribution stability. In summary, the operations in the temporal branch can be formulated as follows:

$$f_T = \mathcal{F}(\mathcal{F}(TC(X_P), W_5), W_6),$$

$$F_T = BN(f_T),$$
(6)

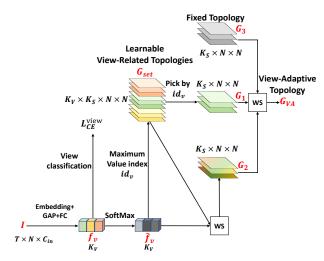


Fig. 4: VATL defines a set of learnable view-related topologies and employs prior-view knowledge to learn the view-adaptive topology. GAP, FC, and WS denote global average pooling, fully-connected layer, and weighted summation respectively.

where $W_5 \in \mathbb{R}^{T_P \times (\alpha \times T_P)}$ inflates the temporal dimension by the ratio α and $W_6 \in \mathbb{R}^{(\alpha \times T_P) \times K_T}$ reduces it to a defined size K_T .

Discussion. Some appearance-based gait methods [2], [3], [5] use part-based approaches to model the local features. These methods are similar to JSFL to some extent. We compare JSFL with these part-based approaches. Their differences are summarized as follows: a) The part-based methods extract features of different sequences using uniform convolutions, whereas JSFL extracts features for each sequence adaptively. b) The part-based methods use non-shared convolutions, whose parameter usage is times larger than that of the vanilla convolutions. However, JSFL, which saves approximately half of the parameters shown in the first and second rows of Table VI, is more efficient than the vanilla convolution. c) The part-based methods mostly obtain the parts using a manual partition, where the part semantics are not well aligned. In contrast, JSFL can obtain the well-aligned parts from the skeleton inputs.

D. View-Adaptive Topology Learning

The VATL module utilizes the intrinsic view information in each sequence to learn a view-adaptive topology. As shown in Fig. 4, the original gait sequence I is obtained as an input. First, we apply an embedding module to extract view-related features, and then use GAP to aggregate the global view information. Next, a fully-connected layer is employed to obtain the view-classification vector $f_v \in \mathbb{R}^{K_V}$, where K_V denotes the number of views (e.g., 11 in CAISA-B [21] and 14 in OU-MVLP [11]). Here, the cross-entropy loss on f_v , which produces a loss L_{CE}^{view} , is used to supervise feature learning. This ensures view prediction ability. Subsequently, a SoftMax function is employed to produce a value-normalized vector $\tilde{f}_v \in \mathbb{R}^{K_V}$.

A set of learnable topologies $G_{set} = \{G_V^1, G_V^2, ..., G_V^{K_V}\}$ is defined to enhance the view-adaptive capacity, where $G_V^i \in$

 $\mathcal{R}^{K_S \times N \times N}$ denotes the corresponding topology obtained from the *i*-th view. To generate the view-adaptive topology G_{VA} , we first obtain the index of the maximum value in f_v , which is formulated as:

$$id_v = \arg\max \widetilde{f_v},\tag{7}$$

where id_v indicates the view that the sequence most possibly encounters. Then, the corresponding topology is selected from G_{set} by id_v as follows:

$$G_1 = G_{set}[id_v], (8)$$

where G_1 reflects the particular properties that the corresponding view possesses. On the test sets of CASIA-B and OU-MVLP datasets, we achieve top-1 view-classification accuracies of 98.5% and 98.7%, respectively, which are quite reliable. Therefore, for each sequence, given the predicted view-classification result, we can accurately select the adaptive topology for the corresponding view. Considering that the topology is initialized as learnable parameters, it can be updated to adapt to the view characteristics through the backward propagation technique. However, G_1 is not sufficient to represent all types of intra-variation existing in this view; thus, we introduce a supplementation. Considering that the data distribution in \tilde{f}_v reveals the sequence characteristics to some extent, we consider the data as linear weights to combine the topologies in G_{set} . This can be formulated as follows:

$$G_2 = \sum_{i=1}^{K_V} \tilde{f}_v^i G_V^i, \tag{9}$$

Also, we use a fixed topology G_3 , which is an ordinary graph in Eq. (1) to extract the general feature representation. Finally, we fuse G_1 , G_2 and G_3 with coefficients g_1 , g_2 and g_3 to obtain G_{VA} :

$$G_{VA} = q_1 G_1 + q_2 G_2 + q_3 G_3. (10)$$

 G_{VA} is used to dynamically connect body joints in the graph convolutions, which not only correlate joints in the nearby locations but also incorporate joint information in the long range. The dynamic complex connections can effectively enhance the adaptation ability in cross-view scenarios.

IV. EXPERIMENTS

A. Datasets

CASIA-B. The CASIA-B [21] dataset contains 124 walking subjects, and each subject includes 110 sequences obtained from 11 camera views. For each view, each subject has 10 sequences of 3 walking conditions, i.e., 6 sequences of normal (NM) walking, 2 sequences of walking with a bag (BG), and 2 sequences of walking with a coat (CL). The training and testing settings followed the protocols reported in [60]. The sequences of the first 74 subjects were used for training, and the sequences of the remaining 50 subjects were used for testing. Specifically, in the testing phase, the NM sequences were used as gallery sets, and the BG and CL sequences were used as probe sets. The skeleton data were extracted by HRNet [32] and OpenPose [33]. Unless otherwise stated, the HRNet data were used on CASIA-B.

OU-MVLP. The OU-MVLP [11] dataset contains 10307 subjects, and each subject includes 28 sequences obtained from 14 camera views. For each view, each subject has 2 sequences (index '01' and index '02'). The training and testing followed the protocols reported in [11]. The sequences of the first 5153 subjects were used for training, and the sequences of the remaining 5154 subjects were used for testing. Specifically, in the testing phase, the sequences with index '01' were used as gallery sets, and the sequences with index '02' were used as probe sets. This dataset provides skeleton data estimated by AlphaPose [34] and OpenPose [33]. In this paper, we used skeleton data extracted by AlphaPose [34].

B. Implementation Details

Hyperparameters. The detailed hyperparameters applied to CASIA-B and OU-MVLP datasets are listed in Table II.

Training Details. 1) The batch size in the training phase was set to (p, k), where p denotes the number of subjects and k denotes the number of sequences for each subject. The batch sizes used in CASIA-B and OU-MVLP were (8, 16) and (32, 12), respectively. 2) The margins in the triplet loss and circle loss were set to 0.2 and 0.5 respectively. 3) Since the data amount of OU-MVLP is twenty times larger than that of CASIA-B, the number of output channels in the embedding module was set as follows: 5-layer stacked CAG blocks and fully-connected layers to 64/128, 128/256, 128/256, 256/512, 256/512, and 256/512 in CASIA-B and OU-MVLP, respectively. 4) In total, 500 epochs were trained by Adam optimizer. The initial learning rate was set to 1e-4 for the VATL module and to 1e-3 for the remaining parameters. The learning rates were iteratively scaled by the step LR decay with a ratio of 0.1 at 255, 355, and 455 epochs. Also, a warmup strategy for the first 5 epochs was adopted to achieve increased stability in the training process.

C. Comparison with State-of-the-art Methods

CASIA-B. A performance comparison between the proposed method and skeleton-based methods for 3 walking conditions and 11 views on CASIA-B [21] is presented in Table I. The following three main observations can be made: (1) CAG achieves the best performance in all three walking conditions, which proves strong feature representation ability. (2) CAG is robust with various pose estimation methods and achieves the best performance for both HRNet and OpenPose.

Since skeleton-based action recognition is a similar application to skeleton-based gait recognition, some state-of-the-art action recognition methods are compared with CAG using the CASIA-B dataset (Table IV). For a fair comparison, all methods adopted the two-stream architecture. As shown in Table IV, CAG outperforms all these methods, which proves the superiority of gait-specific designs in CAG.

OU-MVLP. A performance comparison of the proposed CAG method against skeleton-based methods using AlphaPose data in the OU-MVLP dataset is shown in Table III. CAG outperforms the current approaches marginally for all camera views, which further demonstrates its feature representation generality in a large-scale dataset and robustness in cross-view scenarios.

TABLE I: Comparison of the proposed CAG method with skeleton-based methods using the CASIA-B dataset in term of the averaged rank-1 accuracies (%), excluding identical-view cases. * stands for using 5 modalities as inputs as the same as Gait-TR [38].

	Gallery NM	Pose	1					0 - 18	80°					Ava
	Probe	_	0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	Avg
	PoseGait [6]	3D Pose	55.3	69.6	73.9	75.0	68.0	68.2	71.1	72.9	76.1	70.4	55.4	68.7
	JointsGait [8]	OpenPose	68.1	73.6	77.9	76.4	77.5	79.1	78.4	76.0	69.5	71.9	70.1	74.4
	GaitGraph [7]	HRNet	85.3	88.5	91.0	92.5	87.2	86.5	88.4	89.2	87.9	85.9	81.9	87.7
	ModelGait [9] (pose_CNN)	HMR	87.1	88.3	93.8	95.4	92.1	92.8	90.5	90.7	88.5	92.4	91.7	91.2
NM	Li. et.al. [10] (pose)	HMR	-	-	-	-	-	-	-	-	-	-	-	93.1
	MSGG [13]	HRNet	88.8	92.6	84.2	94.0	93.0	93.9	92.3	94.5	94.4	94.9	90.9	93.0
	CycleGait [14]	HRNet	92.3	93.2	92.9	93.9	91.9	94.1	94.3	93.3	92.8	91.1	91.1	92.8
	Gait-D [36]	HRNet	87.7	92.5	93.6	95.7	93.3	92.4	92.8	93.4	90.6	88.6	87.3	91.6
	Gait-TR [38]	HRNet	95.7	96.4	97.9	97.0	96.9	95.5	95.1	96.1	96.6	96.0	92.4	96.0
	CAG (proposed)	OpenPose	90.5	91.8	94.1	94.3	94.3	92.3	93.2	92.1	93.1	91.2	87.7	92.2
	CAG (proposed)	HRNet	94.2	96.3	96.8	96.2	96.2	96.0	94.8	96.8	96.4	96.5	93.0	95.7
	CAG (proposed)*	HRNet	96.3	96.5	97.8	97.3	97.2	96.4	95.1	97.2	96.6	96.7	93.5	96.4
	PoseGait [6]	3D Pose	35.3	47.2	52.4	46.9	45.5	43.9	46.1	48.1	49.4	43.6	31.1	44.5
	JointsGait [8]	OpenPose	54.3	59.1	60.6	59.7	63.0	65.7	62.4	59.0	58.1	58.6	50.1	59.1
	GaitGraph [7]	HRNet	75.8	76.7	75.9	76.1	71.4	73.9	78.0	74.7	75.4	75.4	69.2	74.8
	ModelGait [9] (pose_CNN)	HMR	86.8	81.2	84.6	86.8	84.9	83.0	83.9	82.8	82.1	84.0	83.2	83.9
BG	Li. et.al. [10] (pose)	HMR	-	-	-	-	-	-	-	-	-	-	-	88.0
	MSGG [13]	HRNet	77.9	81.3	81.7	80.2	78.2	73.8	76.5	77.0	78.6	80.5	73.0	78.1
	CycleGait [14]	HRNet	87.3	85.5	85.0	84.1	82.3	82.9	84.6	82.7	81.7	85.6	82.4	84.0
	Gait-D [36]	HRNet	78.2	80.1	79.3	80.2	78.4	77.6	80.4	78.6	79.1	80.2	76.5	79.0
	Gait-TR [38]	HRNet	90.9	92.4	91.4	93.2	91.9	90.2	91.4	93.9	93.9	92.7	82.9	91.3
	CAG (proposed)	OpenPose	81.5	86.9	88.8	86.8	85.6	84.1	85.9	86.6	85.1	83.3	75.0	84.5
	4 1 /	HRNet	87.5	90.0	91.0	91.1	87.8	88.1	89.2	91.8	90.6	91.6	87.1	89.6
	CAG (proposed)*	HRNet	90.5	92.0	91.9	92.8	91.0	90.4	91.7	93.9	93.6	92.8	89.1	91.8
	PoseGait [6]	3D Pose	24.3	29.7	41.3	38.8	38.2	38.5	41.6	44.9	42.2	33.4	22.5	36.0
	JointsGait [8]	OpenPose	48.1	46.9	49.6	50.5	51.0	52.3	49.0	46.0	48.7	53.6	52.0	49.8
	GaitGraph [7]	HRNet	69.6	66.1	68.8	67.2	64.5	62.0	69.5	65.6	65.7	66.1	64.3	66.3
	ModelGait [9](pose_CNN)	HMR	63.0	62.4	66.3	65.2	61.9	58.2	58.3	59.1	56.8	55.4	55.6	60.2
CL	Li. et.al. [10] (pose)	HMR	-	-	-	-	-	-	-	-	-	-	-	64.3
	MSGG [13]	HRNet	62.2	67.4	66.2	70.2	68.8	66.2	67.4	96.2	71.1	73.4	69.7	68.3
	CycleGait [14]	HRNet	78.6	76.8	79.2	80.5	78.0	77.6	81.2	77.1	76.5	82.4	77.7	78.7
	Gait-D [36]	HRNet	73.2	71.7	75.4	73.2	74.6	72.3	74.1	70.5	69.4	71.2	66.7	72.0
	Gait-TR [38]	HRNet	86.7	88.2	88.4	89.7	91.1	90.7	93.2	93.8	93.2	91.2	83.6	90.0
	CAG (proposed)	OpenPose	67.1	73.1	80.1	77.6	80.6	79.4	78.8	75.1	78.3	73.4	69.5	75.7
	u 1 ,	HRNet	84.1	87.3	88.6	90.3	89.9	89.6	91.0	90.6	89.6	89.3	86.9	88.6
	CAG (proposed)*	HRNet	86.3	88.8	89.3	91.0	91.1	91.2	93.0	92.8	92.6	91.0	89.0	90.6

TABLE II: Hyperparameter settings applied to CASIA-B / OU-MVLP datasets.

T	N	K_S	K_T	K_V	T_P	r
60/32	17/18	3/3	9/9	11/14	15/15	8/8
α	g_1	g_2	g_3	λ_1	λ_2	λ_3
2/2	$\frac{1}{2}/\frac{1}{2}$	$\frac{1}{2}/\frac{1}{2}$	1/1	0.9/0.9	0.1/0.1	0.1/0.1

D. Comparison with Appearance-based Methods

In this section, we compare CAG with appearance-based methods using the CASIA-B dataset in terms of rank-1 accuracy, computational cost, and inference speed in Table V. The computational cost and inference speed are measured using a 100-frame sequence for all models. The advantages of CAG include the following two main aspects: 1) CAG achieves higher performances in the CL condition than appearance-based methods, which indicates its stronger robustness against clothing variations. 2) CAG is more computationally efficient during training and inference phases.

TABLE III: Comparison of the proposed CAG method with skeleton-based methods using the OU-MVLP dataset; averaged rank-1 accuracies (%), excluding identical-view cases; * indicates that the results are produced by the authors.

Probe	Ga	llery All 14 view	/S
	CNN-Pose [6]	GaitGraph [7]*	CAG (ours)
0°	12.3	24.4	45.4
15°	22.7	36.8	61.2
30°	29.3	40.3	64.7
45°	31.5	42.5	67.6
60°	30.5	41.9	67.0
75°	24.7	38.9	63.5
90°	18.1	33.5	57.7
180°	8.7	21.3	39.9
195°	12.3	24.6	48.3
210°	15.5	21.7	44.0
225°	23.5	34.0	61.0
240°	23.3	33.5	60.8
255°	18.3	30.4	57.1
270°	15.2	27.1	52.1
Avg	20.4	30.4	56.4

TABLE IV: Comparison of the proposed CAG method with skeleton-based action recognition methods using the CASIA-B dataset.

Model	Rank-1 Accuracy						
Model	NM	BG	CL	Avg			
2S-AGCN [20]	92.7	80.8	79.3	84.2			
MSG3D [59]	92.0	81.4	80.1	84.5			
CTR-GCN [61]	92.3	80.6	76.7	83.2			
CAG	95.7	89.6	88.6	91.3			

TABLE V: Comparison of the proposed CAG method with appearance-based methods using the CASIA-B dataset.

Model	Rank-	1 Accur	acy(%)	GFLOPs	Inference
	NM	BG	CL	GILOIS	Time (ms)
GaitSet [1]	95.0	87.2	70.4	21.4	2.4
GaitPart [2]	96.2	91.5	78.7	21.4	4.9
CAG	95.7	89.6	88.6	2.1	1.5

E. Ablation Study

Effectiveness of the proposed modules. The effectiveness of the proposed modules is presented in Table VI. The baseline refers to replacing the CAG blocks with ordinary blocks in Eq. (1) and using only the fixed topology for graph convolutions. The following main observations can be made: 1) Comparing the first three experiments, the proposed JSFL and VATL modules both contribute to recognition performance in all three conditions, which confirms the robustness and effectiveness of the dynamic filter learning and view-adaptive topology learning, respectively. 2) In the fourth experiment when applying JSFL and VATL modules together, not only further improves the recognition performance, but also requires fewer parameters and FLOPs than the baseline, which demonstrates the mutual promotion of each module and the efficiency of the proposed design. 3) Combining the joint and bone streams, the best accuracy is achieved, which confirms the complementary properties provided by the joint and bone

Ablation experiments were also conducted using the OU-MVLP dataset (Table VII), where each module still works well in a large-scale dataset.

Investigation of JSFL impact. The JSFL impact was investigated by conducting experiments on the CASIA-B dataset to study the effects of adaptive filter learning and joint-level feature exploration individually. As shown in Table VIII, the second experiment uses the proposed JSFL, the third experiment generates non-adaptive joint-specific filters, and

TABLE VI: Effectiveness and complexity of the proposed modules in terms of averaged rank-1 accuracy (%) using the CASIA-B dataset.

Model	R		Accurac	у	Params	FLOPs
Wiodei	NM	BG	CL	Avg	(M)	(G)
Baseline	92.5	83.1	83.3	86.3	2.05	0.68
Baseline w/JSFL	93.9	87.2	87.2	89.5	1.07	0.30
Baseline w/VATL	94.0	85.9	87.3	89.1	2.09	0.72
CAG (joint)	94.9	87.8	88.1	90.3	1.17	0.38
CAG (joint+bone)	95.7	89.6	88.6	91.3	2.34	0.75

TABLE VII: Effectiveness of the proposed modules in terms of averaged rank-1 accuracy (%) using the OU-MVLP dataset.

Model	Baseline	Baseline w/JSFL	Baseline w/VATL	CAG (joint)	CAG (joint+bone)
Rank-1 Accuracy (%)	26.2	39.2	38.7	45.2	56.4

TABLE VIII: JSFL impact on the CASIA-B dataset in terms of averaged rank-1 accuracy (%).

Model	Rank-1 Accuracy						
Wiodei	NM	BG	CL	Avg			
Baseline	92.5	83.1	83.3	86.3			
Baseline w/JSFL	93.9	87.2	87.5	89.5			
Baseline w/JSFL (non-adaptive)	91.6	75.5	78.6	81.9			
Baseline w/JSFL (global)	88.0	76.8	77.6	80.8			

the fourth experiment is equipped with adaptive filters learned at the global level. Therefore, JSFL (non-adaptive) refers to applying non-shared depth-wise convolutions to different joints, which is joint-specific but non-adaptive. JSFL (global) refers to learning only one group of weights for all joints, which is adaptive but not joint-specific. An interesting finding is that by taking the fine-grained or adaptive mechanism only, the performance is degraded, which indicates that by applying the joint-specific mining or adaptive learning mechanism only, both high model capacity and computational efficiency is hardly achieved unless used together such as in JSFL.

TABLE IX: Effectiveness of VATL topologies G_1 , G_2 and G_3 on the CASIA-B dataset in terms of averaged rank-1 accuracy (%).

Т	opolog	у	Rank-1 Accuracy					
G_1	G_2	G_3	NM	BG	CL	Avg		
$\overline{\hspace{1em}}$			93.8	85.7	87.3	88.8		
	√		93.7	84.7	87.0	88.5		
		√	92.5	83.1	83.3	86.3		
$\overline{\hspace{1em}}$	√		93.9	85.3	87.8	89.0		
$\overline{\hspace{1em}}$		√	93.8	85.8	87.3	89.0		
	√	√	93.8	84.7	87.1	88.5		
$\overline{\hspace{1em}}$	√	✓	94.0	85.9	87.3	89.1		

Impact of VATL topologies. The effectiveness of VATL topologies was investigated by conducting ablation experiments using the CASIA-B dataset. In Table IX, the following main observations can be made: 1) When applying only one topology, the adaptive learned G_1 and G_2 achieve better performance than the fixed G_3 , which confirms the superiority of the proposed view-adaptive learning. 2) By combing them, better performance is achieved, which demonstrates that the topologies were designed at complementary levels. Thus, using all three topologies, the best performance can be achieved.

Combining CAG with appearance-based methods. The mutual promotion of CAG and appearance-based methods was investigated by integrating CAG with two appearance-based methods. In Table X, for a fair comparison, we follow the ensemble settings as in GAITTAKE [15] and BiFusion [13]. The results show that the combined network achieves better performances in all three scenarios, which demonstrates the

complementary properties of motion modeling obtained from CAG and appearance learning obtained from appearance-based methods. And the best ensemble results also indicate that the proposed CAG is a better option to assist the appearance-based methods and make gait recognition more powerful.

Comparison of a view-specific embedding method [55] and VATL. As shown in Tab. XI, we compare the proposed VATL module with a view-embedding method [55], which aims to obtain transformed view-invariant features. We can see that the proposed view-adaptive topologies are more effective in dealing with view variations.

F. HyperParameter Configurations

VATL Configurations. The impact of different settings of coefficients g_1 , g_2 , and g_3 in VATL on the performance is presented in Table XII. It is observed that a balanced coefficient combination achieves better performance than an unbalanced one. Considering the first four experiments, the fourth model (90.3%) outperforms the other three models when the sum of the adaptive topology coefficients (g_1 and g_2) equals the fixed topology coefficient (g_3). This result indicates the importance of balancing adaptive learning and general representation learning. The last two experiments also conform to the law that the sixth model (balanced) outperforms the fifth model (unbalanced). Finally, g_1 , g_2 and g_3 were set to $\frac{1}{2}$, $\frac{1}{2}$, and 1, respectively, to achieve the best performance.

JSFL Configurations. The impact of different settings of temporal kernel size K_T , temporal inflation ratio α , and channel reduction ratio r in JSFL on the performance is presented in Table XIII. The following observations can be made: 1) By comparing the first three experiments ($K_T = 3$, 5, and 9, respectively), it is observed that the performance increases as the value of K_T increases, which indicates that large temporal receptive fields benefit from modeling rich temporal clues in gait modeling. Thus, K_T was set to 9 to achieve the best performance. 2) By comparing the fourth ($\alpha = 2$) and the fifth ($\alpha = 4$) experiments, it is observed that the proposed model with $\alpha = 2$ achieves better performance and lower complexity than that with $\alpha = 4$. Thus, $\alpha = 2$ was set in the proposed network. 3) By comparing the last three experiments (r = 4, 8, and 16), it is observed that the proposed model with r = 4 achieves the best average accuracy but it is much more complex than those with r = 8 and r = 16. The model with r = 16 exhibits the worst performance because a large

TABLE X: Performance of CAG combined with appearance-based methods in terms of averaged rank-1 accuracy (%) using the CASIA-B dataset.

Method	Appearance -based Method	Ensemble Method	NM	BG	CL	Avg
GaitSet [1]	-	-	95.0	87.2	70.4	84.2
GaitPart [2]	-	-	96.2	91.5	78.7	88.8
GaitGL [5]	-	-	97.4	94.5	83.6	91.8
GAITTAKE [15]	3DCNN	Concat	98.0	97.5	92.2	95.9
BiFusion [13]	GaitPart	FCs	98.7	96.0	92.1	95.6
Ours	3D CNN	Concat	98.6	97.9	93.0	96.5
Ours	GaitPart	FCs	98.8	97.3	93.2	96.4

TABLE XI: Comparison of the proposed VATL module with a view-specific embedding method [55] using the CASIA-B dataset in terms of averaged rank-1 accuracy (%).

Method	NM	BG	CL	Avg
Baseline	92.5	83.1	83.3	86.3
Baseline w/view-specific embedding [55]	93.0	83.2	83.3	86.5
Baseline w/VATL	94.0	87.8	87.3	89.1

TABLE XII: Impact of settings of coefficients g_1 (G_1), g_2 (G_2) and g_3 (G_3) in VATL on the performance in terms of averaged rank-1 accuracy (%) using the CASIA-B dataset [21].

g_1	g_2	g_3	NM	BG	CL	Avg
1	1	1	94.5	87.3	87.3	89.7
1	1	$\frac{4}{5}$	94.3	87.1	87.2	89.5
1	1	$\frac{1}{2}$	94.2	87.2	87.1	89.5
$\frac{1}{2}$	$\frac{1}{2}$	1	94.9	87.8	88.1	90.3
$\frac{1}{4}$	$\frac{1}{4}$	1	94.3	87.0	87.6	89.6
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	94.4	88.3	87.5	90.1

channel reduction ratio degrades the representation ability. Finally, considering both the performance and complexity, we set r = 8 in our network.

Weights in the loss function. The impact of weights used in the loss function on the performance is presented in Table XIV. The following observations can be made: 1) Comparing the results in the first and second rows, based on the triplet loss, using the view-classification loss could further improve performance. 2) Comparing the results in the second and third rows, based on the view-classification loss, using triplet loss achieves better performance than using circle loss; when both are used, better performance can be achieved as shown in the fourth row. 3) By increasing the weight of triplet loss (λ_1) and decreasing the weight of circle loss (λ_2) simultaneously, the performances can be further improved. 4) By decreasing the weight of cross entropy loss (λ_3) used in view classification, the recognition performance can be improved, which indicates that decreasing λ_3 helps the model focus on modeling identityrelated features.

G. Visualization

Learned filters. The distribution of the learned filters in JSFL can be visualized using a violin plot, as shown in Fig. 5. The following observations can be made: 1) Fig. 5(a) and Fig. 5(b) represent the data distribution of the learned filters from the same person but different sequences, i.e., the two sequences are captured under different camera viewpoints and walking conditions. Therefore, there exist intra-subject variations in the two sequences. To adapt to the variations, the learned filters for the two sequences have slight differences. 2) By comparing the learned filters of two different subjects '102' and '103' (Fig. 5(c) and Fig. 5(d)), it is observed that the filters are learned diversely, which is mainly due to the personalized walking styles of different subjects.

With the help of visualization, we can realize the adaptation ability of JSFL to cope with customized characteristics in different sequences, which offers a possible scheme to model complex gait patterns.

TABLE XIII: Impact of settings of K_T , α and r in JSFL on the performance in terms of averaged rank-1 accuracy (%) and complexity using the CASIA-B [21] dataset.

K_T	α	r	NM	BG	CL	Avg	param (M)	FLOPs (G)
3	2	8	94.1	87.0	87.4	89.5	1.14	0.37
5	2	8	94.5	87.2	87.8	89.8	1.14	0.38
9	2	8	94.9	87.8	88.1	90.3	1.14	0.38
9	2	8	94.9	87.8	88.1	90.3	1.14	0.38
9	4	8	94.9	87.6	88.0	90.2	1.15	0.39
9	2	4	94.8	88.0	88.1	90.3	1.70	0.51
9	2	8	94.9	87.8	88.1	90.3	1.14	0.38
9	2	16	94.2	86.7	87.4	89.4	0.86	0.31

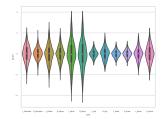
TABLE XIV: Impact of the loss function settings λ_1 (L_{tri}), λ_2 (L_{circle}) and λ_3 (L_{CE}^{view}) on the performance in terms of averaged rank-1 accuracy (%) using the CASIA-B [21] dataset.

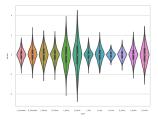
λ_1	λ_2	λ_3	NM	BG	CL	Avg
1	0	0	92.4	86.5	86.1	88.3
1	0	1	92.9	87.7	87.3	89.3
0	1	1	91.5	84.2	84.7	86.8
1	1	1	93.8	87.6	87.3	89.5
1	1	0.1	94.1	87.6	87.6	89.8
0.5	0.5	0.1	94.0	87.5	87.5	89.7
0.3	0.7	0.1	93.7	86.4	86.7	88.9
0.7	0.3	0.1	94.5	87.6	87.9	90.0
0.9	0.1	0.1	94.9	87.8	88.1	90.3

View-adaptive topologies. In Fig. 6, the topology correlations in view-related topology set G_{set} in the CASIA-B and OU-MVLP datasets are visualized. The following two interesting findings that adhere to human intuitions are summarized as 1) The main diagonal line of the two heatmaps indicates that the view-adjacent topologies generally exhibit stronger correlations than the view-distant topologies, indicating that sequences in adjacent views possess similar spatial-temporal characteristics. 2) The anti-diagonal lines of the two heatmaps indicate that the mirror-view topologies exhibit relatively strong correlations (e.g., 0° and 180° , 18° and 162° in the CAISA-B dataset; 0° and 180° , 60° and 240° in the OU-MVLP dataset), indicating that sequences in mirror views have related features. Consequently, the high response regions on the heatmaps resemble an 'X' format.

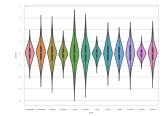
V. CONCLUSION

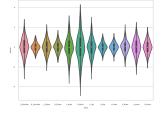
In this paper, a CAG convolutional network for skeleton-based gait recognition was proposed. For each sequence, CAG automatically produces dynamic joint-specific filters to describe personalized fine-grained walking features and learns a view-adaptive topology to fit customized properties under various view conditions. In this way, CAG achieves great adaptation ability in complex scenarios, using the two most popular datasets (i.e., CASIA-B and OU-MVLP), demonstrating its superiority over previous methods. Furthermore, the potential of integrating skeleton-based and appearance-based methods was investigated. Further investigation on combining these two methods will be conducted in future work.





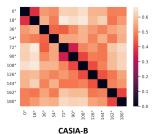
(a) A sequence obtained from subject(b) A different sequence obtained '100'. from subject '100'.





(c) Sequence obtained from subject(d) Sequence obtained from subject '102'. '103'.

Fig. 5: Visualized statistics of the learned filters in the CASIA-B dataset using a violin plot, which plots the parameter distribution of body joints on the arms and legs in the last CAG block. The area of the violin plot denotes the data range, and the bandwidth represents the probability density of data for different values. In each plot, the black box and the range of the slim black line represent the interquartile range and the 95% confidence interval respectively. Best viewed with zoom in.



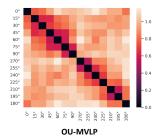


Fig. 6: Topology correlations in G_{set} of the joint stream in CASIA-B and OU-MVLP datasets. The correlation is measured by the mean square error. Darker color indicates a stronger correlation. Best viewed in color.

ACKNOWLEDGEMENT

This research was supported by the NSFC (grant no.61773176).

REFERENCES

 H. Chao, Y. He, J. Zhang, and J. Feng, "Gaitset: Regarding gait as a set for cross-view gait recognition," AAAI, vol. 33, pp. 8126–8133, 2019.
 1, 2, 8, 9

- [2] C. Fan, Y. Peng, C. Cao, X. Liu, S. Hou, J. Chi, Y. Huang, Q. Li, and Z. He, "Gaitpart: Temporal part-based model for gait recognition," CVPR, pp. 14225–14233, 2020. 1, 2, 3, 5, 8, 9
- [3] X. Huang, D. Zhu, H. Wang, X. Wang, B. Yang, B. He, W. Liu, and B. Feng, "Context-sensitive temporal feature learning for gait recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12909–12918. 1, 2, 5
- [4] Z. Huang, D. Xue, X. Shen, X. Tian, H. Li, J. Huang, and X.-S. Hua, "3d local convolutional neural networks for gait recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14920–14929. 1, 2
- [5] B. Lin, S. Zhang, and X. Yu, "Gait recognition via effective global-local feature representation and local temporal aggregation," in *Proceedings* of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14648–14656. 1, 2, 5, 9
- [6] R. Liao, S. Yu, W. An, and Y. Huang, "A model-based gait recognition method with body pose and human prior knowledge," *Pattern Recogni*tion, vol. 98, p. 107069, 2020. 1, 2, 7
- [7] T. Teepe, A. Khan, J. Gilg, F. Herzog, S. Hörmann, and G. Rigoll, "GaitGraph: Graph convolutional network for skeleton-based gait recognition," in 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 2314–2318. 1, 2, 3, 7
- [8] N. Li, X. Zhao, and C. Ma, "Jointsgait: A model-based gait recognition method based on gait graph convolutional networks and joints relationship pyramid mapping," arXiv preprint arXiv:2005.08625, 2020. 1, 2, 3, 4, 7
- [9] X. Li, Y. Makihara, C. Xu, Y. Yagi, S. Yu, and M. Ren, "End-to-end model-based gait recognition," in *Proceedings of the Asian conference* on computer vision, 2020. 1, 2, 7
- [10] X. Li, Y. Makihara, C. Xu, and Y. Yagi, "End-to-end model-based gait recognition using synchronized multi-view pose constraint," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4106–4115. 1, 2, 7
- [11] W. An, S. Yu, Y. Makihara, X. Wu, C. Xu, Y. Yu, R. Liao, and Y. Yagi, "Performance evaluation of model-based gait on multi-view very large population database with pose sequences," *IEEE Transactions* on *Biometrics, Behavior, and Identity Science*, vol. 2, no. 4, pp. 421–430, 2020. 1, 2, 5, 6
- [12] M. Mao and Y. Song, "Gait recognition based on 3d skeleton data and graph convolutional network," in 2020 IEEE International Joint Conference on Biometrics (IJCB). IEEE, 2020, pp. 1–8. 1, 2
- [13] Y. Peng, K. Ma, Y. Zhang, and Z. He, "Learning rich features for gait recognition by integrating skeletons and silhouettes," *arXiv preprint arXiv:2110.13408*, 2021. 1, 2, 3, 7, 8, 9
- [14] N. Li and X. Zhao, "A strong and robust skeleton-based gait recognition method with gait periodicity priors," *IEEE Transactions on Multimedia*, 2022. 1, 2, 3, 7
- [15] H.-M. Hsu, Y. Wang, C.-Y. Yang, J.-N. Hwang, H. L. U. Thuc, and K.-J. Kim, "Gaittake: Gait recognition by temporal attention and keypoint-guided embedding," in 2022 IEEE International Conference on Image Processing (ICIP). IEEE, 2022, pp. 2546–2550. 1, 3, 8, 9
- [16] X. Zhang, C. Xu, and D. Tao, "Context aware graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14333–14342. 2, 3
- [17] F. Ye, S. Pu, Q. Zhong, C. Li, D. Xie, and H. Tang, "Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 55–63. 2, 3
- [18] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling gcn with dropgraph module for skeleton-based action recognition," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16. Springer, 2020, pp. 536–553. 2, 3
- [19] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI* conference on artificial intelligence, 2018. 2, 3
- [20] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 12 026–12 035. 2, 4, 8
- [21] S. Yu, D. Tan, and T. Tan, "A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition," *ICPR*, vol. 4, pp. 441–444, 2006. 2, 5, 6, 9, 10
- [22] X. Li, Y. Makihara, C. Xu, Y. Yagi, and M. Ren, "Gait recognition via semi-supervised disentangled representation learning to identity and covariate features," CVPR, pp. 13309–13319, 2020. 2

- [23] Z. Zhang, L. Tran, X. Yin, Y. Atoum, X. Liu, J. Wan, and N. Wang, "Gait recognition via disentangled representation learning," CVPR, pp. 4710–4719, 2019. 2
- [24] Z. Zhang, L. Tran, F. Liu, and X. Liu, "On learning disentangled representations for gait recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 345–360, 2020.
- [25] S. Hou, C. Cao, X. Liu, and Y. Huang, "Gait lateral network: Learning discriminative and compact representations for gait recognition," *Euro*pean Conference on Computer Vision, pp. 382–398, 2020. 2
- [26] H. Wu, J. Tian, Y. Fu, B. Li, and X. Li, "Condition-aware comparison scheme for gait recognition," *IEEE Transactions on Image Processing*, 2020. 2
- [27] X. Huang, X. Wang, B. He, S. He, W. Liu, and B. Feng, "Star: Spatio-temporal augmented relation network for gait recognition," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 5, no. 1, pp. 115–125, 2022. 2
- [28] T. Wolf, M. Babaee, and G. Rigoll, "Multi-view gait recognition using 3d convolutional neural networks," *ICIP*, pp. 4165–4169, 2016. 2
- [29] B. Lin, S. Zhang, and F. Bao, "Gait recognition with multiple-temporal-scale 3d convolutional neural network," ACMMM, pp. 3054–3062, 2020.
- [30] M. S. Nixon, J. N. Carter, J. M. Nash, P. S. Huang, D. Cunado, and S. V. Stevenage, "Automatic gait recognition," 1999. 2
- [31] L. Wang, H. Ning, T. Tan, and W. Hu, "Fusion of static and dynamic body biometrics for gait recognition," *IEEE Transactions on circuits and* systems for video technology, vol. 14, no. 2, pp. 149–158, 2004. 2
- [32] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703, 2019. 2, 6
- [33] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 172–186, 2019. 2, 6
- [34] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *ICCV*, 2017. 2, 6
- [35] R. Liao, C. Cao, E. B. Garcia, S. Yu, and Y. Huang, "Pose-based temporal-spatial network (ptsn) for gait recognition with carrying and clothing variations," *Chinese conference on biometric recognition*, pp. 474–483, 2017. 2
- [36] S. Gao, J. Yun, Y. Zhao, and L. Liu, "Gait-d: Skeleton-based gait feature decomposition for gait recognition," *IET Computer Vision*, vol. 16, no. 2, pp. 111–125, 2022. 2, 3, 7
- [37] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang, "Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition," in *Proceedings of the 28th ACM International Conference* on Multimedia, 2020, pp. 1625–1633. 2
- [38] C. Zhang, X.-P. Chen, G.-Q. Han, and X.-J. Liu, "Spatial transformer network on skeleton-based gait recognition," arXiv preprint arXiv:2204.03873, 2022. 2, 3, 7
- [39] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7122–7131.
- [40] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141. 3
- [41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [42] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and* pattern recognition, 2018, pp. 7794–7803. 3
- [43] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020. 3
- [44] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," arXiv preprint arXiv:1904.04971, 2019. 3
- [45] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 030–11 039. 3

- [46] Y. Zhang, J. Zhang, Q. Wang, and Z. Zhong, "Dynet: Dynamic convolution for accelerating convolutional neural networks," arXiv preprint arXiv:2004.10694, 2020. 3
- [47] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16.* Springer, 2020, pp. 282–298.
 [48] D. Li, J. Hu, C. Wang, X. Li, Q. She, L. Zhu, T. Zhang, and
- [48] D. Li, J. Hu, C. Wang, X. Li, Q. She, L. Zhu, T. Zhang, and Q. Chen, "Involution: Inverting the inherence of convolution for visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer* Vision and Pattern Recognition, 2021, pp. 12321–12330. 3
- [49] Z. Liu, L. Wang, W. Wu, C. Qian, and T. Lu, "Tam: Temporal adaptive module for video recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13708–13718.
- [50] J. Zhou, V. Jampani, Z. Pi, Q. Liu, and M.-H. Yang, "Decoupled dynamic filter networks," in *Proceedings of the IEEE/CVF Conference* on Computer Vision and Pattern Recognition, 2021, pp. 6647–6656.
- [51] H. Dou, P. Zhang, W. Su, Y. Yu, and X. Li, "Metagait: Learning to learn an omni sample adaptive representation for gait recognition," in Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V. Springer, 2022, pp. 357–374. 3
- [52] Y. He, J. Zhang, H. Shan, and L. Wang, "Multi-task gans for view-specific feature learning in gait recognition," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 102–113, 2018.
- [53] S. Yu, H. Chen, E. B. Garcia Reyes, and N. Poh, "Gaitgan: Invariant gait feature extraction using generative adversarial networks," CVPR, pp. 30–37, 2017. 3
- [54] T. Chai, X. Mei, A. Li, and Y. Wang, "Silhouette-based view-embeddings for gait recognition under multiple views," in 2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021, pp. 2319–2323.
- [55] T. Chai, A. Li, S. Zhang, Z. Li, and Y. Wang, "Lagrange motion analysis and view embeddings for improved gait recognition," in *Proceedings of* the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 20249–20258, 3, 9
- [56] Y. Makihara, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi, "Gait recognition using a view transformation model in the frequency domain," in *European conference on computer vision*. Springer, 2006, pp. 151– 163. 3
- [57] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," arXiv preprint arXiv:1703.07737, 2017. 4
- [58] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6398–6407. 4
- [59] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and* pattern recognition, 2020, pp. 143–152. 4, 8
- [60] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, "A comprehensive study on cross-view gait based human identification with deep cnns," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 2, pp. 209–226, 2016. 6
- [61] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13359–13368.



Xinggang Wang (M'17) received the B.S. and Ph.D. degrees in Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2009 and 2014, respectively. He is currently an Associate Professor with the School of Electronic Information and Communications, HUST. His research interests include computer vision and machine learning. He services as associate editors for Pattern Recognition and Image and Vision Computing journals and an editorial board member of Electronics journal.



Zhidianqiu Jin received the M.E. degree in School of Electronic Information and Communications from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2021. His current research areas include computer vision and machine learning.



Bo Yang received the Master degree in School of mathematics and statistics form Wuhan University, Wuhan, China. He is currently the senior engineer of Wuhan FiberHome Digital Technology Co., Ltd. His research interests include computer vision and data mining.



Botao He received the Ph.D. degree in School of Optical and Electronic Information from Huazhong University of Science and Technology (HUST), Wuhan, China. He is currently the deputy general manager of Wuhan FiberHome Digital Technology Co., Ltd. His research interests include computer vision and data mining.



Xiaohu Huang received the B.E. and M.E. degree in School of Electronic Information and Communications from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2020 and 2023. His current research areas include computer vision and machine learning.



Bin Feng received the B.S. and Ph.D. degrees in School of Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2001 and 2006, respectively. He is currently an Associate Professor with the School of Electronic Information and Communications, HUST. His research interests include computer vision and intelligent video analysis.



Wenyu Liu (SM'15) received the B.S. degree in Computer Science from Tsinghua University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees, both in Electronics and Information Engineering, from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is now a professor and associate dean of the School of Electronic Information and Communications, HUST. His current research areas include computer vision, multimedia, and machine learning.