Zero-Shot Skeleton-Based Action Recognition With Prototype-Guided Feature Alignment

Kai Zhou, Shuhai Zhang, Zeng You, Jinwu Hu, *Graduate Student Member, IEEE*, Mingkui Tan, *Senior Member, IEEE*, and Fei Liu, *Member, IEEE*

***Abstract*—Zero-shot skeleton-based action recognition aims to classify unseen skeleton-based human actions without prior exposure to such categories during training. This task is extremely challenging due to the difficulty in generalizing from known to unknown actions. Previous studies typically use two-stage training: pre-training skeleton encoders on seen action categories using cross-entropy loss and then aligning pre-extracted skeleton and text features, enabling knowledge transfer to unseen classes through skeleton-text alignment and language models' generalization. However, their efficacy is hindered by 1) insufficient discrimination for skeleton features, as the fixed skeleton encoder fails to capture necessary alignment information for effective skeleton-text alignment; 2) the neglect of alignment bias between skeleton and unseen text features during testing. To this end, we propose a prototype-guided feature alignment paradigm for zero-shot skeleton-based action recognition, termed PGFA. Specifically, we develop an end-to-end cross-modal contrastive training framework to improve skeleton-text alignment, ensuring sufficient discrimination for skeleton features. Additionally, we introduce a prototype-guided text feature alignment strategy to mitigate the adverse impact of the distribution discrepancy during testing. We provide a theoretical analysis to support our prototype-guided text feature alignment strategy and empirically evaluate our overall PGFA on three well-known datasets. Compared with the top competitor SMIE method, our PGFA achieves absolute accuracy improvements of 22.96%, 12.53%, and 18.54% on the NTU-60, NTU-120, and PKU-MMD datasets, respectively.**

***Index Terms*—Zero-shot, skeleton-based action recognition, contrastive learning, distribution discrepancy, prototypical learning.**

Received 19 September 2024; revised 23 April 2025; accepted 23 June 2025. Date of publication 18 July 2025; date of current version 24 July 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62072190, Grant U24A20327, Grant U23B2013, and Grant 62276176; and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515010900. The associate editor coordinating the review of this article and approving it for publication was Dr. Zheng Wang. *(Kai Zhou and Shuhai Zhang contributed equally to this work.) (Corresponding authors: Mingkui Tan; Fei Liu.)*

Kai Zhou and Fei Liu are with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: kayjoe0723@gmail.com; feiliu@scut.edu.cn).

Shuhai Zhang and Jinwu Hu are with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Pazhou Laboratory, Guangzhou 510330, China (e-mail: shuhaizhangshz@gmail.com; fhujinwu@gmail.com).

Zeng You is with the School of Future Technology, South China University of Technology, Guangzhou 511442, China, and also with the Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: zengyou.yz@gmail.com).

Mingkui Tan is with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Key Laboratory of Big Data and Intelligent Robot, Ministry of Education, South China University of Technology, Guangzhou 510006, China (e-mail: mingkuitan@scut.edu.cn).

Data is available on-line at https://github.com/kaai520/PGFA

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2025.3586487, provided by the authors.

Digital Object Identifier 10.1109/TIP.2025.3586487

I. Introduction

SKELETON-BASED action recognition aims to classify human actions based on the movements of skeletal joints. It offers notable advantages in robustness to appearance variations and privacy preservation compared to RGB-based methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], making it applicable in various domains, including human-robot interaction [12], [13], healthcare and rehabilitation (*e.g.*, fall detection) [14], [15], and sports analysis [16].

Despite the significant progress achieved by existing fully supervised deep learning methods on this task [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], handling numerous action classes in real-world scenarios is uneconomical due to the high cost of annotating and labeling 3D video skeleton action data. Each annotated action clip requires costly spatiotemporal segmentation, posing practical limitations on scalability [29]. This necessitates effective zero-shot skeleton-based action recognition methods [30], [31], [32], [33] to overcome these challenges, enabling accurate action recognition without the extensive need for labeled data of new actions.

Zero-shot skeleton-based action recognition methods recognize and classify unseen human actions using skeletal data without prior exposure to such categories during training. This approach leverages knowledge transfer from previously seen actions to recognize new, unseen actions, thereby mitigating the need to collect or annotate new actions. However, this task is challenging due to the following primary difficulties. 1) The subtle differences between similar actions require highly discriminative models to distinguish them accurately [33]. 2) The substantial variance in human actions makes it difficult to generalize from known to unknown actions [32].

Existing zero-shot skeleton-based action recognition methods [30], [31], [32], [33] typically use a pre-trained text encoder to extract text features of action classes, which are then matched with skeleton features to predict the action class of the input skeleton sequence. This approach is adopted because text descriptions encapsulate rich semantic information that aids in generalizing to unseen action categories [32]. Previous methods generally follow a two-stage training framework, as shown in Fig. 1(a). Initially, they train a skeleton encoder using cross-entropy loss on data from seen action categories. Afterward, the skeleton encoder is fixed and used to extract skeleton features. Concurrently, they use a pre-trained text encoder to extract text features from class

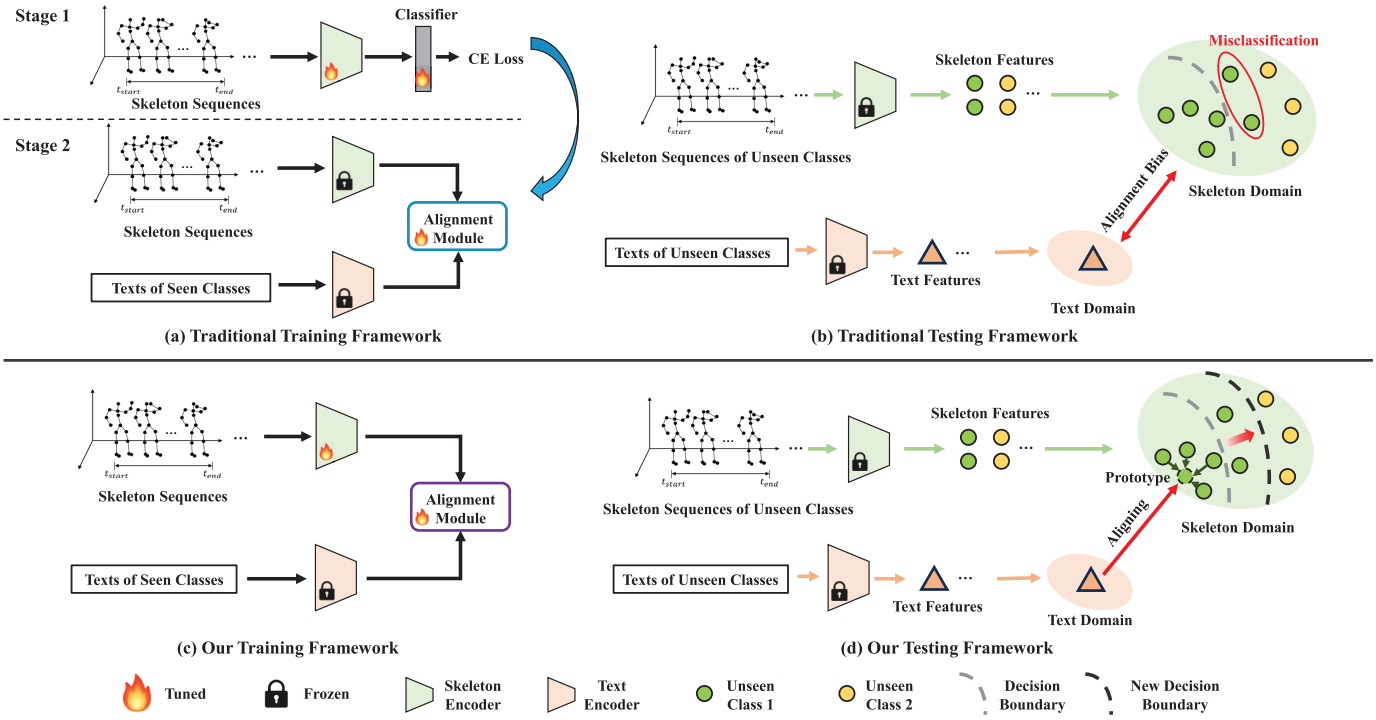


Fig. 1. Illustration of traditional framework and our framework for zero-shot skeleton-based action recognition. **(a)** Traditional training framework. "CE Loss" denotes the cross-entropy loss. **(b)** Traditional testing framework. Unseen skeleton samples are classified by calculating similarity with text features of unseen classes (i.e., prediction based on the highest similarity score). However, due to the distributional shifts from seen to unseen classes, alignment bias between skeleton features and unseen text features is inevitable. **(c)** Our training framework. Unlike the traditional two-stage training framework, our end-to-end training framework significantly enhances training efficiency and avoids potential inconsistencies arising from independently trained components. **(d)** Our testing framework. Compared to the traditional testing framework, our testing framework leverages prototype features derived from the unseen skeleton feature distribution to replace unseen text features. It mitigates the negative impact of alignment bias between skeleton features and unseen text features.

name [30], [32], descriptions [32], [33], or part-of-speech tagged words [31]. These text and skeleton features are used to train an alignment module to map the text and skeleton features into the same semantic feature space for seen action categories. During testing, they compute similarities between the skeleton features of test samples and the text features of unseen classes and then obtain their predictions based on the highest similarity score (see Fig. 1(b)).

Despite advances in these methods, several limitations persist: 1) *Insufficient discrimination in training:* The skeleton feature extractor, trained separately using the "pretrained&fixed" two-stage framework, fails to capture the intrinsic correlation between text and skeleton features effectively. Even with the subsequent alignment module, it struggles to produce discriminative skeleton features that align well with the textual space due to potential inconsistencies between the training modules in the two-stage process. Specifically, the skeleton encoder extracts features from unseen categories with insufficient discriminative power, resulting in a small margin between different categories (see a visualization of t-SNE in Fig. 7), making it difficult to match the text representations correctly. 2) *Alignment bias in testing:* During testing, substantial distributional differences between seen and unseen categories inevitably cause misalignment between skeleton features and the corresponding unseen text features, which we call alignment bias. Ignoring these distributional differences during testing will severely hamper the model's generalization. Addressing this alignment bias to improve zero-shot prediction capability remains an important yet unresolved challenge.

To address the above issues, we investigate how to improve the alignment of the distributions of skeleton and text features from both *training* and *testing* perspectives. In the *training* phase, we seek to employ an end-to-end contrastive learning approach to improve cross-modal alignment, ensuring sufficient discrimination for skeleton features. The fundamental idea behind contrastive learning is to ensure intra-class compactness by pulling together similar instances while pushing apart dissimilar ones [34], showing great potential in cross-modal alignment tasks [35], [36], [37], [38]. Moreover, the end-to-end manner reduces the potential inconsistencies from independently trained components and enables efficient optimization within a single unified framework. Benefiting from these merits, we propose **an end-to-end cross-modal contrastive training framework**. This framework avoids the limitations associated with pre-training skeleton encoders using cross-entropy loss by employing contrastive training with high-quality pre-extracted text features in an end-to-end manner. Additionally, we explore four action description generation methods to create diverse text inputs using pre-trained large language models. This approach aims to acquire higher-quality text features, thereby improving skeleton-text alignment and enhancing the zero-shot recognition performance.

In the *testing* phase, we seek an effective prototypical learning approach to alleviate the adverse impact of the distri-

bution discrepancy between unseen and seen action categories. Prototypical learning has proven to be a highly effective strategy for few-shot learning [39], [40] and domain adaptation [41], [42], [43]. When skeleton features of unseen classes exhibit modest intra-class compactness, selecting and creating prototype features from these skeleton features can provide better alignment with unseen skeleton features than using the original unseen text features. Since prototype features are derived from the distribution of unseen skeleton features, they better represent the central tendency of the unseen classes, leading to more consistent and robust alignment compared to isolated unseen text features. To achieve this, we propose a **prototype-guided text feature alignment strategy**, adjusting the unseen text features to better align with the skeleton features. Initially, we obtain the pseudo-label of each skeleton sequence based on its similarity to the original unseen text features. To reduce the impact of mislabeling and noise, we introduce a filtering mechanism that selects skeleton features with high confidence, thereby improving the accuracy of subsequent prototype creation. We then create prototypes for each unseen class by calculating the centroid of the high-confidence features. Finally, we reclassify each test sample based on its similarity to the prototype feature of each unseen class. By using dynamically derived prototypes instead of static unseen text features, we address the alignment bias issue and enhance the model's generalization.

We integrate the aforementioned training and testing frameworks into a paradigm, termed **prototype-guided feature alignment paradigm** for zero-shot skeleton-based action recognition (**PGFA**). We qualitatively and quantitatively evaluate our PGFA on NTU-60, NTU-120, and PKU-MMD datasets. Experimental results demonstrate that our PGFA achieves absolute accuracy improvements of 22.96%, 12.53%, and 18.54% on the NTU-60, NTU-120, and PKU-MMD datasets, respectively, compared with the top competitor SMIE [32]. Our contributions are summarized as follows:

- We propose a prototype-guided feature alignment paradigm for zero-shot skeleton-based action recognition, termed PGFA, to mitigate issues of insufficient discrimination and alignment bias during training and testing. Extensive experiments demonstrate the superior performance of PGFA in zero-shot action recognition.
- We introduce an **end-to-end** cross-modal contrastive training framework to ensure sufficient discrimination for skeleton features. Unlike traditional two-stage frameworks, our end-to-end approach enhances training efficiency and ensures intra-class compactness by aligning skeleton and text features for the same category while promoting dissimilarity for different categories. Additionally, we explore four description generation methods to improve skeleton-text alignment. Qualitative and quantitative results validate the effectiveness of our framework in enhancing skeleton feature discrimination and skeleton-text alignment.
- We propose a prototype-guided text feature alignment strategy to alleviate alignment bias between skeleton and unseen text features during testing. To the best of our knowledge, we are the first to improve skeleton-text alignment during testing for zero-shot skeleton-based action recognition. Theoretical and empirical results demonstrate the effectiveness of this strategy in mitigating the negative impact of alignment bias.

II. Related Work

A. Skeleton-Based Action Recognition

With the rise of accurate depth sensors like Kinect cameras and pose estimation algorithms, skeleton-based action recognition is gaining attention.

In earlier times, Recurrent Neural Networks (RNNs) are applied to handle skeleton sequences [44], [45], [46]. Drawing inspiration from the achievements of Convolutional Neural Networks (CNNs) in image-related tasks, 2D-CNN-based methods [20], [47], [48], [49] initially represent the skeleton sequence as a pseudo image and employ CNNs to model relationships between skeletal joints. PoseConv3D [26] is the first to apply 3D-CNNs to skeleton-based action recognition. It utilizes a 3D heatmap volume to represent human skeletons, offering a novel perspective in capturing spatial-temporal dynamics of human actions. Graph Convolutional Network (GCN)-based methods [17], [18], [19], [21], [22], [23], [24], [25], [27], [28], [50] have gained significant attention for representing human joints as graph nodes and their connections through adjacency matrices. ST-GCN [17] introduces a spatial-temporal graph convolutional approach to capture human joint relationships across both spatial and temporal dimensions, using predefined spatial graphs reflecting the body's natural joint connections and temporal edges for consecutive frames. Shift-GCN [50] advances skeleton-based action recognition by leveraging shift graph convolutional networks, enabling efficient spatial and temporal joint relationship modeling without the computational burden of traditional convolutions. Recent interest in transformers [51], [52] lead to exploring transformer-based methods [53], [54], [55] for skeleton data. STST [55] introduces a Spatial-Temporal Specialized Transformer, effectively modeling skeleton sequences by employing distinct joint organization strategies for spatial and temporal dimensions, capturing skeletal movements efficiently. In this paper, we utilize ST-GCN [17] and Shift-GCN [50] as skeleton encoders to extract features following SMIE [32]. Our method enables the replacement of the skeleton encoder with various network architectures for training.

B. Zero-Shot Skeleton-Based Action Recognition

Despite the advancements made in skeleton-based action recognition, the existing methods [30], [31], [32], [33] explored in zero-shot skeleton-based action recognition remain limited. Jasani et al. [30] are the first to explore zero-shot skeleton-based action recognition. They extend DeViSE [56] and RelationNet [57] to skeleton-based action recognition by extracting skeleton features with ST-GCN pre-trained on seen classes and text embeddings with Word2Vec [58] or Sentence-Bert [59]. Gupta et al. [31] introduce the SynSE method, which incorporates a generative multi-modal alignment module to align skeleton features with parts of speech-tagged words. They use additional PoS syntactic information to

classify labels into verbs and nouns, finding that separately extracted text embeddings enhance generalization. Their proposed SynSE outperforms their reimplemented baseline methods, including ReViSE [60], JPoSE [61], and CADA-VAE [62], in zero-shot prediction capabilities. The above methods map the skeleton or text features to fixed points in the embedding space, which does not consider the global distribution of the semantic features. To address this issue, Zhou et al. [32] propose the SMIE method, which incorporates a global alignment module to estimate the mutual information between skeleton and text features, along with a temporal constraint module to capture the inherent temporal information of actions. Recently, Zhu et al. [33] propose the PURLS framework, which introduces a sophisticated text prompting module and a novel skeleton partitioning module to generate aligned textual and skeleton representations across different levels. Our method differs from previous approaches [30], [31], [32], [33] in two key aspects: **(1) Training:** We use an end-to-end training framework that aligns skeleton and text features via direct cross-modal contrastive learning, unlike the commonly used two-stage training frameworks, such as SMIE [32], which pre-trains with cross-entropy loss and fixes the skeleton encoder. **(2) Testing:** During testing, our prototype-guided text feature alignment strategy adapts the unseen text features to better align with the skeleton features, which is a significant improvement over prior methods that rely on static unseen text features.

C. Multi-Modal Representation Learning

Recently, multi-modal representation learning methods such as CLIP [35] and ALIGN [63] have gained significant attention for demonstrating that vision-language co-training can develop robust representations for downstream tasks, including zero-shot image classification and text-image retrieval. Subsequently, multi-modal representation learning has excelled in other fields, including video understanding [36], [64], 3D point cloud understanding [37], [38], [65], and 3D human action generation [66]. ActionCLIP [36] adapts CLIP's training scheme for video action recognition by incorporating additional transformer layers into a pre-trained CLIP model for temporal modeling of video data. ULIP [37] and CG3D [38] enhance 3D point cloud understanding by learning a unified representation across images, texts, and point clouds. MotionCLIP [66] aligns human action latent space with CLIP latent space for 3D human action generation. In the field of skeleton-based action recognition, the fully supervised GAP [67] method first utilizes generative prompts and a multi-modal training paradigm to guide action recognition. GAP employs the alignment of skeleton and text modalities as an auxiliary task to assist supervised learning, primarily utilizing multi-part contrastive learning based on parts descriptions of the human body in motion. In contrast to GAP, the work most closely related to ours, our method mainly differs in two key aspects. First, the alignment of skeleton and text features in our method is not an auxiliary task; without the support of supervised learning, the parts descriptions used by GAP may not be suitable in zero-shot scenarios (cf. Section IV-E). Second, beyond aligning skeleton and text features of seen action classes, we introduce a prototype-guided text feature alignment strategy that adjusts unseen text features for better alignment with skeleton features.

Algorithm 1 Training Scheme for PGFA

Input: The training dataset $\mathcal{D}_s = \{(\mathbf{x}_i^s, t_i^s)\}_{i=1}^{N_s}$, the pre-trained text encoder E_t, the batch size B, the number of training epochs T.
Output: The skeleton encoder E_x and the projection layer ψ.

```
for j = 1, ..., T do
    for each batch {(x_i^s, t_i^s)}_{i=1}^{B} ⊂ D_s do
        Calculating the training loss L_KL via Eq. (5).
        Updating the skeleton encoder E_x and the projection
        layer ψ by minimizing L_KL.
    end for
end for
```

D. Prototypical Learning

Prototypical learning involves learning class prototypes to represent class features for classification tasks. It has been widely applied in few-shot learning [39], [40], semi-supervised learning [68], and domain adaptation [41], [42], [43], where it effectively handles limited labeled data and evolving learning tasks by leveraging prototype-based classification. Snell et al. [39] propose Prototypical Networks for few-shot classification, utilizing prototype representations of each class in a learned metric space to achieve excellent results. Xu et al. [68] propose a novel approach for semi-supervised semantic segmentation that addresses intra-class variation by regularizing within-class feature distribution, leveraging consistency between linear and prototype-based predictors to encourage proximity to within-class prototypes while maintaining separation from between-class prototypes. Iwasawa et al. [43] propose T3A, which improves model robustness to distribution shifts by adjusting classifiers during test time using pseudo-prototype representations. Lin et al. [42] proposed ProCA, which effectively aligns domains and preserves prior knowledge through label prototype identification and prototype-based alignment and replay strategies. However, while prototypical learning in most existing methods typically focuses on known classes, our approach concerns prototypical learning for unknown classes.

III. Proposed Methods

A. Problem Formulation and Method Overview

We aim to address zero-shot skeleton-based action recognition, where the model is trained on seen classes and tested on disjoint unseen classes. In this context, we define the training set as $\mathcal{D}_s = \{(\mathbf{x}_i^s, t_i^s)\}_{i=1}^{N_s}$, where N_s denotes the number of training samples, $\mathbf{x}_i^s$ and t_i^s denote a skeleton sequence and the corresponding action description from *seen* classes, respectively. Similarly, we define the testing set as $\mathcal{D}_u = \{(\mathbf{x}_i^u, t_i^u)\}_{i=1}^{N_u}$, which includes N_u testing samples, with $\mathbf{x}_i^u$ and t_i^u representing a skeleton sequence and the corresponding action description from *unseen* classes, respectively. In the zero-shot setting [32], we have each $t_i^s \in T^s$, $t_i^u \in T^u$, and $T^s \cap T^u = \varnothing$, where T^s and T^u denote action descriptions set of *seen* and *unseen* classes,

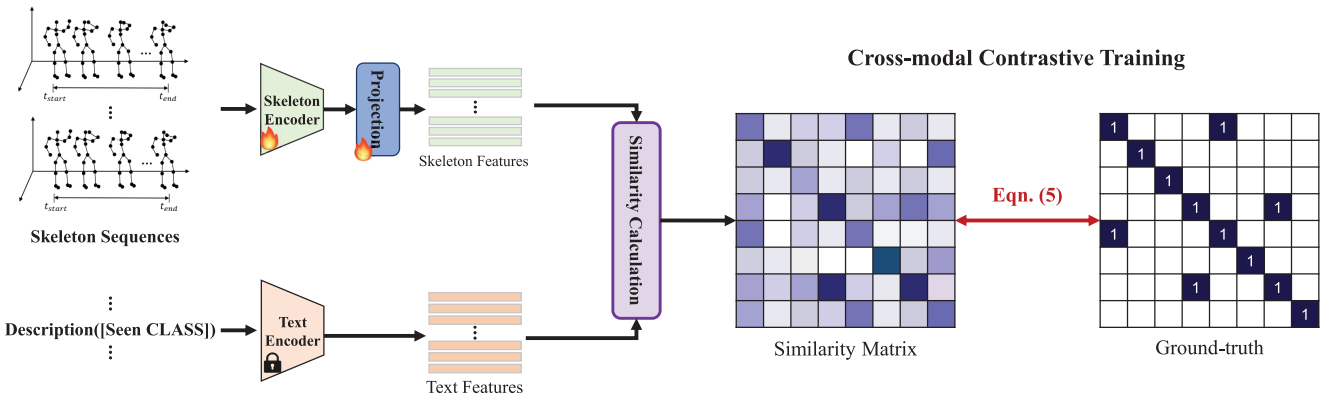


Fig. 2. Training framework of proposed PGFA. During *training*, we employ contrastive cross-modal training in an end-to-end manner to simultaneously train a skeleton encoder and a projection layer, aiming to learn a unified representation space for skeleton sequences and text.

Class Name:
Drink water

Parts Description:
Drink water: head tilts back slightly; hand grasps cup; arm lifts cup to mouth; hip remains stationary; leg remains stationary; foot remains stationary.

Complete Description:
"Drink water" refers to the act of taking in water by mouth to hydrate or quench thirst as a human action.

Skeleton-focused Description:
In the "Drink water" action sequence, the skeletal motion begins with the upper limb, particularly the shoulder and elbow joints of one arm, elevating and flexing to bring the hand upward towards the head level. The forearm and wrist joints articulate to position the hand directly in front of the head. As the elbow maintains flexion, there is slight stabilization noticeable in the shoulder girdle. Concurrently, minimal adjustments in the spine and neck may align the head slightly forward. Following this, the arm's motion reverses as it lowers back to the starting position by extending at the elbow and lowering at the shoulder.

Fig. 3. Different descriptions of action "drink water".

Algorithm 2 Testing Scheme for PGFA

Input: The testing dataset $\mathcal{D}_u$, the text encoder E_t, the skeleton encoder E_x and the projection layer ψ.
Output: The final prediction set $\{\tilde{y}_i\}_{i=1}^{N_u}$.

```
for each x_i^u ∈ D_u do
    Generating a pseudo label ŷ_i of x_i^u via Eq. (6).
end for
for each unseen class k = 1, ..., K do
    Calculating a prototype feature c^{u,k} via Eq. (10).
end for
for each x_i^u ∈ D_u do
    Generating a final predicted label ỹ_i of x_i^u via Eq. (11).
end for
```

respectively. This task is challenging due to the potential substantial variance between seen and unseen human actions, but the subtle differences between similar actions.

To address this, we aim to learn a mapping between textual and visual feature spaces to recognize new action categories and propose a prototype-guided feature alignment paradigm for zero-shot skeleton-based action recognition, called PGFA. PGFA consists of 1) **an end-to-end cross-modal contrastive training framework**, which learns a unified representation space of skeleton sequence and text, as illustrated in Fig. 2 and Alg. 1, and 2) **a prototype-guided text feature alignment strategy** for testing, which adjusts the unseen text features to align skeleton features better, as shown in Fig. 4 and Alg. 2.

B. End-to-End Cross-Modal Contrastive Training

To construct a mapping between textual and skeletal feature spaces, existing methods [30], [31], [32], [33] typically involve pre-training skeleton encoders using cross-entropy loss on seen classes, subsequently fixing them to extract features from seen skeletons, which are then matched with text features derived from a pre-trained text encoder. However, this two-stage training often fails to generate discriminative skeleton features that align effectively with the textual space, thereby compromising generalization performance. Additionally, numerous studies have criticized cross-entropy loss for its lack of intra-class compactness and inadequate margins [69], [70], [71], [72], which hinder subsequent modality alignment. To address this, we introduce **an end-to-end cross-modal contrastive training framework** for zero-shot skeleton-based action recognition.

Formally, given an input skeleton sequence $\mathbf{x}_i^s$ and a text description of corresponding action t_i^s, we first employ a skeleton encoder E_x and a text encoder E_t to extract the skeleton feature $\mathbf{v}_i^s$ and text feature $\mathbf{w}_i^s$ by:

$$\mathbf{v}_i^s = \psi(E_x(\mathbf{x}_i^s)), \tag{1}$$

$$\mathbf{w}_i^s = E_t(t_i^s), \tag{2}$$

where ψ is a projection layer, ensuring the output dimension of the skeleton feature and text feature are consistent. Inspired by [35] and [36], we calculate the bidirectional softmax-normalized similarity scores represented by:

$$p_b^{x2t}(\mathbf{v}_i^s) = \frac{\exp(\text{sim}(\mathbf{v}_i^s, \mathbf{w}_b^s)/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(\mathbf{v}_i^s, \mathbf{w}_j^s)/\tau)}, \tag{3}$$

$$p_b^{t2x}(\mathbf{w}_i^s) = \frac{\exp(\text{sim}(\mathbf{v}_b^s, \mathbf{w}_i^s)/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(\mathbf{v}_j^s, \mathbf{w}_i^s)/\tau)}, \tag{4}$$

where $\text{sim}(\cdot,\cdot)$ is the cosine similarity, τ is a learnable temperature parameter and B is batch size, b ranges from 1 to B. Unlike the one-to-one image-text pairs in CLIP [35], our

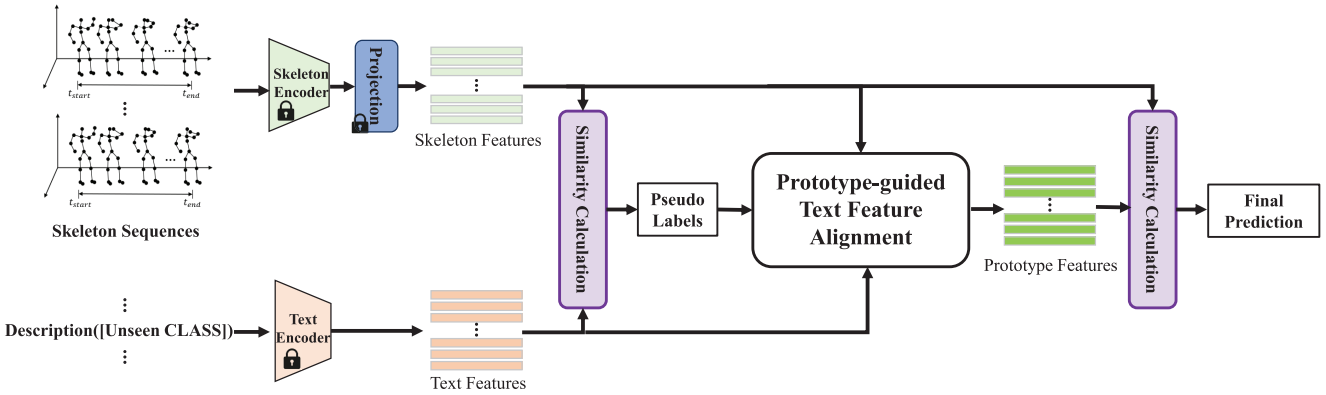


Fig. 4. Testing framework of proposed PGFA. During *testing*, we first generate pseudo-labels for each skeleton sequence based on its similarity to the original unseen text features. Using these pseudo-labels, we create a prototype feature for each unseen class. Finally, we reclassify each sample based on its similarity to the prototype feature rather than the text feature of each unseen class. For a detailed illustration of the prototype-guided text feature alignment, please refer to Fig. 5.

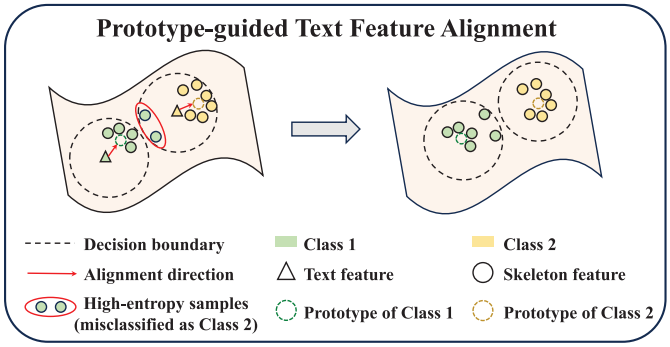


Fig. 5. Illustration of prototype-guided text feature alignment strategy. Class 1 and Class 2 represent two unseen classes. We illustrate the alignment directions for text features within the feature space before alignment. After alignment, we use the prototype features to replace the original text features in constructing the decision boundaries.

task involves multiple positive matches within a batch, as several samples may belong to the same action category. It is not proper to regard this similarity learning as a 1-in-N classification problem with InfoNCE loss [73]. Instead, we utilize Kullback–Leibler (KL) divergence as the contrastive loss for skeleton-text alignment learning:

$$\mathcal{L}_{KL}=\frac{1}{2}\sum_{i=1}^{B}\left[\mathrm{KL}\left(\mathbf{p}^{x2t}(\mathbf{v}_i^s),\mathbf{m}_i^{x2t}\right)+\mathrm{KL}\left(\mathbf{p}^{t2x}(\mathbf{w}_i^s),\mathbf{m}_i^{t2x}\right)\right], \quad (5)$$

where $\mathbf{p}^{x2t}(\mathbf{v}_i^s)=[p_1^{x2t}(\mathbf{v}_i^s),\ldots,p_B^{x2t}(\mathbf{v}_i^s)]$ is the i-th row vector of the whole similarity scores matrix. Similarly, $\mathbf{p}^{t2x}(\mathbf{w}_i^s)$ also denotes the i-th column vector of its similarity scores matrix. $\mathbf{m}_i^{x2t}$ and $\mathbf{m}_i^{t2x}$ represent the corresponding ground-truth similarity scores vector, which assigns a probability of 1 for positive pairs and a probability of 0 for negative pairs.

Note that $\mathcal{L}_{KL}$ is exclusively employed for training the skeleton encoder E_x and the projection layer ψ. This is necessitated by the limited number of skeleton-text pairs available for training, which is insufficient to support the training of the text encoder, unlike the ample training data used for language or vision-language pre-training [35], [59]. Therefore, we adopt a pre-trained large-scale language model (such as Sentence-BERT [59]) as our text encoder E_t, and we keep the parameters of the text encoder fixed during training. To save computational resources, we pre-extract text features to eliminate the need for text encoder inference during training.

Action Description Generation. The quality of action description is crucial for skeleton-text alignment learning. We explore four action description generation methods in our zero-shot settings. In Fig. 3, we present various generation methods for generating the action "drink water". The details of the four methods are as follows.

(a) Class Name: One simple and quick solution is to directly use the class name of the action as input for the text encoder. Using this generation method within our training framework serves as our baseline model. However, these action names contain only a few words and cannot fully and accurately describe the corresponding action semantics. Thus, this method leads to limited zero-shot prediction capability (as shown in our ablation studies). **(b) Parts Description:** The fully-supervised model GAP [67] first uses generative prompts for skeleton-based action recognition. The primary action description in GAP is based on the parts description. GAP leverages GPT-3 [74] to generate action descriptions from different perspectives of body parts, including the head, hand, arm, hip, leg, and foot. Although this generation method demonstrates superior performance in supervised scenarios, its effectiveness in zero-shot scenarios is uncertain. **(c) Complete Description:** We follow the approach of SIME [32], expanding each class name of action into a complete action description using ChatGPT. This generation method can offer comprehensive descriptions of action semantics, which is crucial in skeleton-text alignment learning. Finally, it is worth emphasizing that action description generation is not the primary focus of our paper. We leave ample room for further improvements, such as exploring more complex prompt engineering or prompt tuning techniques. **(d) Skeleton-focused Description:** We use ChatGPT with a carefully designed prompt to ensure the output aligns closely with observable skeletal motion patterns. The prompt is as follows: "Describe the skeletal motion pattern of a human action sequence labeled [CLASS] in a video in a single paragraph of fewer than 100 words. Focus only on joint movements, avoiding References to objects, facial expressions, or any unobservable body parts (e.g., mouth, hair, and toes). Emphasize key joint displacements, relative

limb movements, and the temporal progression of the action based purely on skeletal data." This prompt guides the LLM to generate concise, motion-focused descriptions that exclude irrelevant details and emphasize joint dynamics.

C. Prototype-Guided Text Feature Alignment

In this section, we first review the standard zero-shot testing procedure and then introduce our prototype-guided text feature alignment strategy.

1) Standard Zero-Shot Testing: For each test skeleton sequence $\mathbf{x}_i^u$, the predicted action class $\hat{y}_i$ is determined by taking the argmax over the similarity scores with the unseen semantic features:

$$\hat{y}_i = \arg\max_k P(\hat{y}_i = k), \tag{6}$$

$$P(\hat{y}_i = k) = \frac{\exp(\text{sim}(\mathbf{v}_i^u, \mathbf{w}^{u,k}))}{\sum_{j=1}^{K} \exp(\text{sim}(\mathbf{v}_i^u, \mathbf{w}^{u,j}))}, \tag{7}$$

where $\mathbf{v}_i^u = \psi(E_x(\mathbf{x}_i^u))$ denotes the skeleton feature of $\mathbf{x}_i^u$, and $\mathbf{w}^{u,k} = E_t(t^{u,k})$ denotes the text features of $t^{u,k}$, with $t^{u,k} \in T^u$ representing the action descriptions of the unseen class k. Here, $T^u = \{t^{u,1}, \ldots, t^{u,K}\}$ is a set of action descriptions for unseen classes, consisting of K action descriptions. In the zero-shot setting, $\mathbf{w}^{u,k}$ is not involved in training, so there is no guarantee that it will align well with unseen skeleton features.

2) Prototype-Guided Text Feature Alignment Strategy: To address the above issue, benefitting from the effectiveness of prototypical learning [41], [42], [43] in transfer learning scenarios, we propose **a prototype-guided text feature alignment strategy** to better align with unseen skeleton features in a zero-shot setting. Specifically, we first acquire the pseudo label $\hat{y}_i$ of each skeleton sequence $\mathbf{x}_i^u$ based on Eq. (6). Then we create a prototype feature for each unseen class to replace the corresponding unseen text feature, using skeleton features whose pseudo-labels belong to that unseen class. To this end, we create a support set $\mathbb{S}^k$ for each unseen class k:

$$\mathbb{S}^k = \left\{ \frac{\mathbf{v}_i^u}{\|\mathbf{v}_i^u\|} \mid \hat{y}_i = k, i \le N_u, i \in \mathbb{N}^+ \right\}, \tag{8}$$

where $\|\cdot\|$ denotes the L2 norm. N_u denotes the number of testing samples in the testing set. Intuitively, taking the centroid of the support set $\mathbb{S}^k$ as a prototype feature to replace the unseen text feature $\mathbf{w}^{u,k}$ can align well with the skeleton features belonging to the unseen class k. However, inevitably, some pseudo-labels are misassigned to incorrect classes, making their utilization undesirable due to the introduction of noise into the prototype feature. To tackle this issue, we use the prediction entropy $H(\mathbf{z}) = -\sum_k P(\hat{y}_i = k \mid \mathbf{z}) \log P(\hat{y}_i = k \mid \mathbf{z})$ to filter the element $\mathbf{z} = \frac{\mathbf{v}_i^u}{\|\mathbf{v}_i^u\|}$ in $\mathbb{S}^k$. To be specific, we create a final support set $\mathbb{Z}^k$ for each seen class k:

$$\mathbb{Z}^k = \left\{ \mathbf{z} \mid \mathbf{z} \in \mathbb{S}^k, H(\mathbf{z}) \le h^k \right\}, \tag{9}$$

where h^k is the ρ^k-th smallest entropy of the support set $\mathbb{S}^k$, with $\rho^k = \lfloor \alpha \cdot |\mathbb{S}^k| \rfloor$ denoting the size of $\mathbb{Z}^k$ and $\alpha \in [0, 1]$ being a tolerance margin. We can modify the size of $\mathbb{Z}^k$ by the hyperparameter α. The impact of α is illustrated in Fig. 6. We define a prototype feature to replace $\mathbf{w}^{u,k}$ as follows:

$$\mathbf{c}^{u,k} = \begin{cases} \dfrac{1}{|\mathbb{Z}^k|} \displaystyle\sum_{\mathbf{z} \in \mathbb{Z}^k} \mathbf{z}, & if\ \mathbb{Z}^k \neq \varnothing, \\ \mathbf{w}^{u,k}, & otherwise, \end{cases} \tag{10}$$

where $\mathbf{c}^{u,k}$ is the prototype feature of unseen class k. Note that even if there exist no features belonging to $\mathbb{Z}^k$, *i.e.*, $\mathbb{Z}^k = \varnothing$, the centroid of $\mathbb{Z}^k$ can be reduced to the original text feature $\mathbf{w}^{u,k}$. Built upon the prototype features, We reclassify each test skeleton sequence $\mathbf{x}_i^u$ by:

$$\tilde{y}_i = \arg\max_k \frac{\exp(\text{sim}(\mathbf{v}_i^u, \mathbf{c}^{u,k}))}{\sum_{j=1}^{K} \exp(\text{sim}(\mathbf{v}_i^u, \mathbf{c}^{u,j}))}, \tag{11}$$

where $\tilde{y}_i$ denotes the final predicted action class of $\mathbf{x}_i^u$.

3) Theoretical Analysis for Prototype-Guided Text Feature Alignment Strategy: To elucidate the mechanism of our proposed alignment strategy, we would like to analyze it from a statistical view, making it easier to understand the core insights of this method. To achieve this, we provide the following theorem to give an intuitive explanation, where we do not consider the filter operation on the support set $\mathbb{Z}^k$ and $\mathbb{Z}^k \neq \varnothing$. For simplicity, we denote the normalized skeleton feature of the k-th class as $\mathbf{v}^{(k)}$, where $\|\mathbf{v}^{(k)}\| = 1$. The normalized features are constrained to lie on the unit hypersphere. Therefore, we assume that the features follow a von Mises–Fisher distribution [75], [76], which is considered a generalization of the normal distribution to the unit hypersphere. This distribution's advantageous property in representing cosine similarity by dot-products simplifies theoretical analysis.

Theorem 1: Assuming that the distributions of the K classes of normalized skeleton feature $\mathbf{v}^{(k)}$ are from K von Mises–Fisher distributions [75] with the same concentration parameter κ but different mean directions $\boldsymbol{\mu}_k$, i.e., $f_{p_k}(\mathbf{v}; \boldsymbol{\mu}_k, \kappa) = C_d(\kappa) \exp(\kappa \boldsymbol{\mu}_k^T \mathbf{v})$, $k = 1, \ldots, K$, where $C_d(\kappa)$ is a normalization constant related to the concentration parameter κ and feature dimension d, and the k-th class of prototype feature is $\hat{\boldsymbol{\mu}}_k = \frac{\frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i^{(k)}}{\|\frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i^{(k)}\|}$, which is a normalized $\mathbf{c}^{u,k}$ in Eq. (10). Then, as $n \to \infty$, for $\forall\ \mathbf{v} \sim p_m$, $m = 1, \ldots K$, if $k = \arg\max_i \frac{\exp(\text{sim}(\mathbf{v}, \hat{\boldsymbol{\mu}}_i))}{\sum_{j=1}^{K} \exp(\text{sim}(\mathbf{v}, \hat{\boldsymbol{\mu}}_j))}$, we have

$$P(\mathbf{v}|\text{class } k) > P(\mathbf{v}|\text{class } j), \forall\ j \neq k. \tag{12}$$

For the proof of Theorem 1, please refer to the supplementary. Theorem 1 shows that when the number of samples n in $\mathbb{Z}$ is large, if a sample has the highest similarity to the class center vector of a particular class, then the probability that it belongs to this class is higher than that of it belonging to other classes. Therefore, reclassification according to the maximum similarity criterion in Eq. (11) can improve action classification accuracy from a statistical view. Note that a large n results in a more accurate class center, whereas a small n makes the class center more sensitive to noise. Our entropy-based filtering mechanism mitigates this issue, as demonstrated by the experiments in Fig. 6.

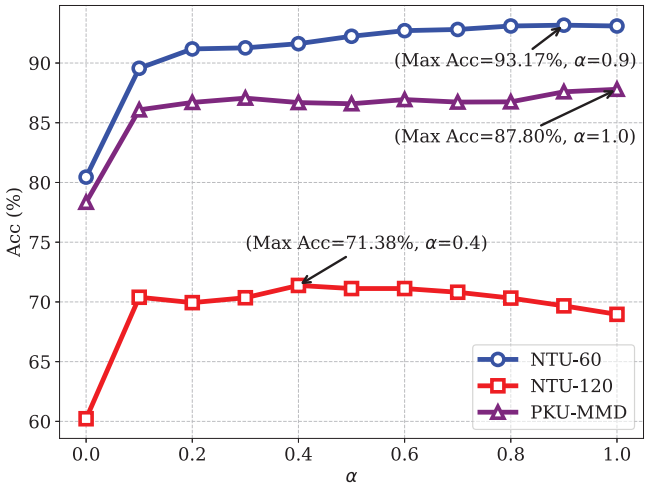


Fig. 6. Effect of tolerance margin α. The x-axis represents the value of α, while the y-axis represents the average accuracy for the three class splits under Setting I.

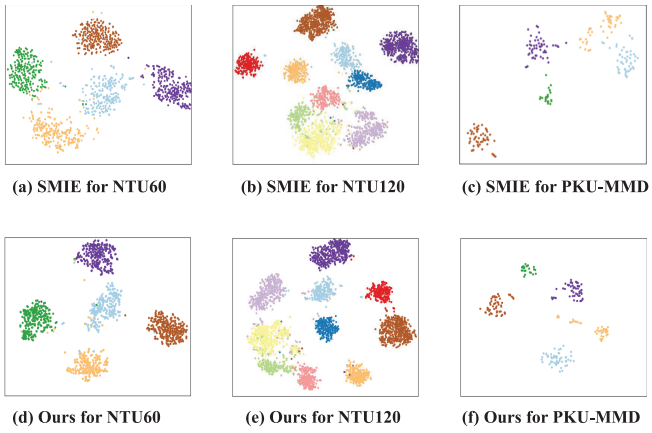


Fig. 7. Visualization of skeleton features using t-SNE [88] from the skeleton encoder pre-trained with cross-entropy loss in SMIE [32] and the skeleton encoder of our method. These encoders are trained on data from seen classes in NTU-60, NTU-120, and PKU-MMD under Setting I. We visualize the skeleton features from unseen classes.

IV. Experiments

A. Datasets

NTU-RGB+D 60 [77] comprises 56,578 skeleton sequences belonging to 60 action categories, which were performed by 40 subjects. These skeleton sequences were captured using Microsoft Kinect sensors, with each subject being represented by 25 joints. In supervised settings, the dataset is usually split into training and testing sets based on subjects or views. However, in zero-shot settings, this dataset is divided into training and test sets based on action categories.

NTU-RGB+D 120 [78] is an extended dataset of NTU-60 (short for NTU-RGB+D 60), encompassing data from 106 subjects and comprising 113,945 skeleton sequences across 120 action categories. In our zero-shot settings, this dataset is also divided into training and test sets based on action categories.

PKU-MMD [79] dataset consists of action samples across 51 categories, collected from 66 subjects. The data is captured using Kinect v2 sensors from multiple viewpoints. This dataset is divided into two parts: Part I includes 21,539 samples, and Part II includes 6,904 samples. We utilize Part I to split the training and test sets according to action categories in this setting, following the SMIE method [32].

B. Evaluation Settings

Setting I. Due to the substantial influence of various class splits on zero-shot results, SMIE [32] suggests a setting on 3 datasets for further verifying the stability of different methods. To achieve this, a three-fold test is applied to each dataset to reduce variance. Each fold comprises distinct groups of seen and unseen classes, and the **average accuracy** of the predicted unseen classes is reported. For NTU-60, they offer three groups of 55/5 class splits, with each group comprising 55 seen and 5 unseen classes. For NTU-120, they provide three groups of 110/10 class splits, with each group comprising 110 seen and 10 unseen classes. For PKU-MMD, they offer three groups of 46/5 class splits, with each group comprising 46 seen and 5 unseen classes. For a fair comparison, we employ ST-GCN [17] as our skeleton encoder and Sentence-BERT [59] as our text encoder in this setting, following the SMIE method.

Setting II. Additionally, we also offer a more extensive comparison with another evaluation setting proposed by SynSE [31]. This setting maintains two fixed class partitions on NTU-60 and NTU-120 datasets. Specifically, for the NTU-60 dataset, SynSE presents 55/5 and 48/12 splits, encompassing 5 and 12 unseen classes, respectively. For the NTU-120 dataset, SynSE offers 110/10 and 96/24 splits. In contrast to Setting I, a one-fold test is required for each class split, rather than a three-fold test. The **accuracy** is reported. We employ Shift-GCN [50] as our skeleton encoder and Sentence-BERT [59] as our text encoder, following the approach used in SynSE.

C. Implementation Details

In Setting I, we adopt the identical data processing procedure as Cross-CLR [80], following SMIE. This procedure entails eliminating invalid frames and adjusting the skeleton sequences to a length of 50 frames through linear interpolation. The skeleton feature extractor employed is ST-GCN with 16 hidden channels, resulting in an extracted feature dimension of 256. We utilize Sentence-BERT to acquire the 768-dimensional text feature following SMIE guidelines. Hence, the projection layer ψ is implemented as a linear layer, projecting from 256 to 768 dimensions. For all experimental runs, we employ the SGD optimizer with 50 epochs. The learning rate is set to 5×10^{-2} for the NTU-60 and PKU-MMD datasets, whereas for the NTU-120 dataset with a larger data size, the learning rate is adjusted to 5×10^{-3} to ensure smoother training. We opt for the complete description of action classes as the default action description due to its superior performance. The tolerance margin α, which controls the size of final support sets, is set to 0.9, 0.4, and 1.0 for the NTU-60, NTU-120, and PKU-MMD datasets, respectively. The effect of α will be analyzed in ablation studies.

TABLE I

COMPARISON WITH PREVIOUS METHODS ON NTU-60, NTU-120, AND PKU-MMD DATASETS UNDER SETTING I. SMIE† INDICATES THAT IT UTILIZES THE COMPLETE DESCRIPTION OF ACTION CLASSES AS TEXT INPUT. PGFA* REFERS TO OUR METHOD UTILIZING THE CLASS NAME AS TEXT INPUT BUT NOT EMPLOYING OUR PROTOTYPE-GUIDED TEXT FEATURE ALIGNMENT STRATEGY. PGFA† REFERS TO OUR METHOD UTILIZING THE COMPLETE DESCRIPTION BUT NOT EMPLOYING OUR PROTOTYPE-GUIDED TEXT FEATURE ALIGNMENT STRATEGY. PGFA DENOTES OUR FULL METHOD

Method Split: (Seen/Unseen)	NTU-60(%) (55/5)	NTU-120(%) (110/10)	PKU-MMD(%) (46/5)
DeViSE [56]	49.80	44.59	47.94
RelationNet [57]	48.16	40.55	51.97
ReViSE [60]	56.97	49.32	65.65
SMIE [32]	63.57	56.37	67.15
SMIE† [32]	70.21	58.85	69.26
PGFA*(Ours)	65.17	57.56	74.76
PGFA†(Ours)	80.45	60.22	78.34
PGFA (Ours)	**93.17**	**71.38**	**87.80**

In Setting II, apart from utilizing Shift-GCN as the skeleton encoder, all other implementation details remain consistent with Setting I. Notably, SynSE employs 4s-Shift-GCN pre-trained on seen classes to extract skeleton features. The 4s-Shift-GCN means utilizing the joint stream, bone stream, joint-motion stream, and bone-motion stream of skeleton sequences to train 4 Shift-GCN models with cross-entropy loss, and finally averages the outputs of these 4 Shift-GCN models. However, our method does not require a pre-trained skeleton encoder with cross-entropy loss, while employing four encoders in our end-to-end training paradigm would be overly complex. Therefore, we only use the joint stream with one Shift-GCN model to train our method, which may lead to relatively weaker skeleton feature extraction but improve training efficiency.

D. Main Results

Under Setting I, we compare our PGFA with previous methods, following the comparison methods outlined in SMIE [32]. Previous methods commonly utilize the class name of action classes as input to a text encoder. SMIE additionally provides experimental results using the complete description of actions as input. For fair comparisons, we provide PGFA* and PGFA†, which use class names and complete descriptions as input, respectively, but do not incorporate our prototype-guided text feature alignment strategy. Additionally, we offer a version incorporating our prototype-guided text feature alignment strategy as our full method to validate its performance. All the datasets get 3 groups of class splits provided by SMIE and the average accuracies are reported on Tab. I. As shown in Tab. I, both PGFA* and PGFA† demonstrate enhancements compared to the top competitor SMIE, especially with PGFA† notably surpassing SMIE† by a large margin. This indicates that higher-quality action descriptions better unlock the potential of our end-to-end training framework for achieving improved skeleton-text alignment compared to the previous two-stage training framework. By integrating our prototype-guided text feature alignment strategy, our full method PGFA achieves average accuracies surpassing SMIE† on the NTU-60, NTU-120, and PKU-MMD datasets by 22.96%, 12.53%, and 18.54%, respectively. The larger performance gains by our full method demonstrate the effectiveness of our prototype-guided text feature alignment strategy.

TABLE II

COMPARSION WITH PREVIOUS METHODS ON NTU-60 AND NTU-120 DATASETS UNDER SETTING II

Method Split: (Seen/Unseen)	NTU-60(%) (55/5)	 (48/12)	NTU-120(%) (110/10)	 (96/24)
DeViSE [56]	60.72	24.51	47.49	25.74
RelationNet [57]	40.12	30.06	52.59	29.06
ReViSE [60]	53.91	17.49	55.04	32.38
JPoSE [61]	64.82	28.75	51.93	32.44
CADA-VAE [62]	76.84	28.96	59.53	35.77
SynSE [31]	75.81	33.30	62.69	38.70
SMIE [32]	77.98	40.18	65.74	45.30
PURLS [33]	79.23	40.99	71.95	52.01
PGFA (Ours)	**80.26**	**55.99**	**79.99**	**59.42**

TABLE III

EFFECT OF ACTION DESCRIPTION GENERATION. WE EXPLORE THE EFFECT OF ACTION DESCRIPTIONS ON SKELETON-TEXT ALIGNMENT TRAINING, WITHOUT USING PROTOTYPE-GUIDED TEXT FEATURE ALIGNMENT OF OUR METHOD

Action Description	NTU-60(%)	NTU-120(%)
Class Name	65.17	57.56
Parts Description	75.65	52.73
Complete Description	**80.45**	60.22
Skeleton-focused Description	77.25	**62.22**

Under Setting II, we compare our PGFA with previous methods, following the comparison methods outlined in SynSE [31], SMIE [32], and PURLS [33]. Apart from using a single Shift-GCN in our end-to-end training framework instead of 4s-Shift-GCN, all other experimental settings are identical to those used in SynSE. Despite the potential for relatively weaker skeleton feature extraction in our PGFA, PGFA still achieves state-of-the-art performance. In the 55/5 and 48/12 splits of the NTU-60 dataset, our PGFA surpasses the top competitor PURLS [33] by 1.03% and 15.00% (relatively 1.30% and 36.59%), respectively. Compared to our method, PURLS shows similar performance in the 55/5 split, which can be attributed to using various complex description prompts and a more powerful text encoder of CLIP [35] instead of Sentence-BERT [59]. In the 110/10 and 96/24 splits of the NTU-120 dataset, our PGFA surpasses PURLS by 8.04% and 7.41% (relatively 11.17% and 14.25%), respectively. As the number of unseen classes increases, aligning skeleton and text modalities becomes increasingly challenging. Other methods performed the worst in the 48/12 split of the NTU-60 dataset. In contrast, our method achieved the largest relative improvement in this split (e.g., relatively 36.59% compared with PURLS), demonstrating promising potential in skeleton-text alignment.

E. Ablation Studies

We conducted experiments to evaluate the effect of the key components and hyper-parameters of our method. To ensure

TABLE IV
EFFECT OF ACTION DESCRIPTIONS GENERATED BY DIFFERENT LLMS ON ZERO-SHOT SKELETON-BASED ACTION RECOGNITION. ALL MODELS DO NOT USE THE PROTOTYPE-GUIDED TEXT FEATURE ALIGNMENT STRATEGY

Action Description	LLM	NTU-60(%)	NTU-120(%)
Complete Description	MiniCPM3	71.60	52.29
	DeepSeek-V3	78.75	59.08
	ChatGPT	**80.45**	**60.22**
Skeleton-focused Description	MiniCPM3	76.59	52.42
	DeepSeek-V3	75.41	57.26
	ChatGPT	**77.25**	**62.22**

the stability of key components and hyper-parameters, we conduct ablation studies under Setting I suggested by SMIE.

1) Effect of Action Description Generation: In skeleton-text alignment, without pre-training skeleton encoders with cross-entropy, the quality of pre-extracted text features becomes crucial for enhancing zero-shot capability. We explore the effect of four action description generation methods and validate these effects on the NTU-60 and NTU-120 datasets. Since GAP [67] does not provide parts descriptions for PKU-MMD, we are concerned that our self-generated parts descriptions may differ in style. Therefore, we do not perform validation on the PKU-MMD dataset. As shown in Tab. III, utilizing the class name of the action as input indeed results in limited performance, primarily because it fails to describe the corresponding action semantics accurately. While parts description performs well in supervised scenarios [67], it is not as effective as the complete description in zero-shot scenarios. The possible reason is that parts descriptions for some actions are too similar. For instance, actions like "put on a shoe" and "take off a shoe" differ only in the description of the foot part, such as "foot inserts into shoe" and "foot grasps the shoe and pulls it off", while descriptions of other body parts remain largely the same. In supervised scenarios, parts description serves as auxiliary supervision. However, in zero-shot scenarios, the text features generated by actions like "put on a shoe" and "take off a shoe" exhibit high similarity, which may introduce interference in primary skeleton-text alignment. In contrast, using complete descriptions directly is more effective than parts descriptions in such cases. The results in Tab. III show that both Complete Description and Skeleton-focused Description outperform the other two methods. On the NTU-60 dataset, using complete descriptions performs better, while on the NTU-120 dataset, using skeleton-focused descriptions yields superior results. This may be due to the larger size and more classes in NTU-120, which demand better alignment with the skeleton data, needing Skeleton-focused Description to more accurately reflect key joint movements for the more complex actions. Considering the fairness of comparison with other zero-shot methods, we primarily use Class Name and Complete Description for comparison. Additionally, employing more complex prompt engineering or prompt tuning techniques might further improve results. However, this aspect is not the primary focus of our paper and will be left for future exploration.

2) Effect of Action Descriptions Generated by Different LLMs: We explore the impact of action descriptions generated by different LLMs in our zero-shot settings. We select the open-source models MiniCPM3 [81] and DeepSeek-V3 [82], as well as the closed-source model ChatGPT, to generate descriptions. As shown in Tab. IV, descriptions generated by ChatGPT consistently achieve the best performance on both NTU-60 and NTU-120 datasets, for both complete and skeleton-focused descriptions. In contrast, MiniCPM3, with its lightweight 4B parameters, demonstrates a performance gap relative to the other two models. This emphasizes the crucial role of LLM quality in improving zero-shot recognition accuracy.

3) Effect of Our End-to-End Training Framework: To further investigate the effects of different training frameworks, we assess the zero-shot prediction capability of three frameworks on our method. **1) Pretraining&Fixed.** This represents the training framework of previous methods, which involves initially pre-training a skeleton encoder with a cross-entropy loss using data from seen categories. Subsequently, the skeleton encoder is fixed, and alignment with the text modality is performed. **2) Pretraining&Finetuning.** This framework involves initially pre-training a skeleton encoder with a cross-entropy loss but finetuning the skeleton encoder for subsequent skeleton-text alignment. **3) End-to-End.** This represents our end-to-end training framework, which involves directly training the skeleton encoder for skeleton-text alignment. To ensure fairness and leverage the superiority of complete descriptions, all models use complete descriptions as text input. As shown in Tab. V, the Pretraining&Fixed framework performs the worst for our method, indicating that the fixed skeleton feature distribution from pretraining the skeleton encoder with cross-entropy loss is not conducive to subsequent skeleton-text alignment. By allowing finetuning of the skeleton encoder during subsequent skeleton-text alignment, the Pretraining&Finetuning framework naturally performs better than the Pretraining&Fixed framework. For our method, the End-to-End framework proves to be the most effective across all three datasets, highlighting its superiority. Pretraining the skeleton encoder is unnecessary, as it yields suboptimal performance and incurs additional pretraining costs.

4) Effect of Our Prototype-Guided Text Feature Alignment: To further evaluate the effectiveness and stability of our prototype-guided text feature alignment strategy, we assess its performance across three different training frameworks of our method on three datasets. As shown in Tab. V, employing our prototype-guided text feature alignment strategy in the testing phase yields remarkable improvements across all training frameworks and datasets, achieving an absolute accuracy improvement of approximately 10%. This demonstrates the strategy's effectiveness and stability.

5) Effect of the Tolerance Margin α: In our prototype-guided text feature alignment strategy, the tolerance margin α controls the size of the final support set $\mathbb{Z}^k$. All models use complete descriptions as text input for training. Fig. 6 shows the performance for different values of α. When $\alpha = 0$, the prototype-guided text feature alignment strategy is not utilized, resulting in significantly lower baseline performance.

TABLE V

EFFECT OF OUR END-TO-END TRAINING FRAMEWORK AND PROTOTYPE-GUIDED TEXT FEATURE ALIGNMENT. NOTE THAT "END-TO-END" IS OUR DEFAULT TRAINING FRAMEWORK. "FEATURE ALIGNMENT" DENOTES OUR PROTOTYPE-GUIDED TEXT FEATURE ALIGNMENT STRATEGY

Training Framework			Testing Strategy	NTU-60(%)	NTU-120(%)	PKU-MMD(%)
Pretraining&Fixed	Pretraining&Finetuning	End-to-End	Feature Alignment	(55/5)	(110/10)	(46/5)
✓				73.65	56.12	71.85
	✓			76.99	57.55	74.63
		✓		80.45	60.22	78.34
✓			✓	87.23	68.66	80.39
	✓		✓	88.55	68.74	82.48
		✓	✓	**93.17**	**71.38**	**87.80**

TABLE VI

EFFECT OF DIFFERENT PSEUDO-LABELING STRATEGIES ON PROTOTYPE GENERATION

Pseudo-labeling Strategy	NTU-60(%)	NTU-120(%)	PKU-MMD(%)
weighted	82.90	57.75	81.51
argmax (default)	**93.17**	**71.38**	**87.80**

TABLE VII

COMPUTATIONAL COST COMPARISON BETWEEN OUR METHOD AND BASELINE SMIE. PGFA† REFERS TO OUR METHOD WITHOUT THE PROTOTYPE-GUIDED TEXT FEATURE ALIGNMENT STRATEGY

Method	Param	FLOPs (G)
SMIE [32]	2.38M	0.58
PGFA†(Ours)	1.00M	0.57

TABLE VIII

EVALUATION ON ONE-SHOT SKELETON-BASED ACTION RECOGNITION. THE TABLESHOWS ACCURACY (%) ON UNSEEN CLASSES

Method	NTU-60(%)	NTU-120(%)	PKU-MMD(%)
Split: (Seen/Unseen)	(50/10)	(100/20)	(41/10)
ProtoNet [39]	74.8	60.4	78.1
FEAT [83]	74.3	61.5	75.9
Subspace [84]	75.6	60.9	75.6
Dynamic Filter [85]	75.9	60.6	78.8
M&C-scale [86]	82.7	68.7	**86.9**
PGFA (Ours)	**83.8**	**69.8**	**86.9**

When $\alpha = 1$, it means that we use all elements of support set $\mathbb{S}^k$ to construct $\mathbb{Z}^k$. Critically, for $\alpha \in [0.1, 1.0]$, our method achieves stable performance (>95% of peak accuracy) across all datasets while consistently surpassing the $\alpha = 0$ baseline by large margins (e.g., NTU-120 attains 71.38% at $\alpha = 0.4$ vs. 60.22% baseline). Intuitively, setting α too low may restrict the prototype calculation to few low-entropy elements, introducing bias, while overly high α risks including misclassified samples. Certainly, the optimal α may vary across different datasets. For NTU-120, the best performance is achieved with $\alpha = 0.4$. Perhaps due to the high accuracy achieved by our method on the NTU-60 and PKU-MMD datasets, it performs optimally when $\alpha = 0.9$ and $\alpha = 1.0$, requiring only a small portion of elements with high prediction entropy to be filtered out. However, empirical results reveal robustness: deviations from optimals (e.g., NTU-120's $\alpha = 0.4$ vs. $\alpha = 0.9$ yielding 98% of peak accuracy) maintain near-optimal performance, highlighting adaptability to data distributions. Optimal α varies slightly (e.g., 0.9 for NTU-60 vs. 0.4 for NTU-120), yet $\alpha \geq 0.1$ ensures significant and stable improvements without requiring fine-tuning, underscoring the strategy's reliability for zero-shot generalization.

6) Effect of Different Pseudo-Labeling Strategies on Prototype Generation: In our prototype-guided text feature alignment strategy, the pseudo-labels used for prototype generation are typically derived through argmax classification. We explored an alternative approach where the pseudo-labels are weighted by their probability. Specifically, we propose to modify the calculation of the prototype feature of class k as follows: $\mathbf{c}^{u,k} = \frac{\sum_{i=1}^{|\mathcal{D}_u|} P_i^k \cdot (\mathbf{v}_i^u / \|\mathbf{v}_i^u\|)}{\sum_{i=1}^{|\mathcal{D}_u|} P_i^k}$, where $|\mathcal{D}_u|$ is the total number of samples in the test set, P_i^k represents the probability of the pseudo-label for sample i belonging to class k, and $\mathbf{v}_i^u$ is the corresponding skeleton feature. As shown in Tab. VI, the performance of the weighted pseudo-labeling strategy is generally lower than the default argmax-based approach. This could be due to the inclusion of features from many samples that do not belong to the correct class, introducing noise into the prototype calculation. Specifically, when probability weighting is used, low-probability samples still contribute to the prototype, which is especially problematic when the model's classification performance is weak. For example, the NTU-120 dataset shows a significant performance drop with the weighted strategy (57.75%) compared to the argmax approach (71.38%).

7) Computational Complexity Analysis: We compare the computational cost of our method and the baseline SMIE in terms of parameters and floating point operations per second (FLOPs). As shown in Tab. VII, our method significantly reduces the number of parameters to 1.00M, compared to SMIE's 2.38M. This is because, with the same ST-GCN backbone, our method adds only one extra linear layer, while SMIE adds three fully connected layers. Without the prototype-guided text feature alignment strategy, our method's FLOPs are also slightly lower than SMIE's. With the feature alignment strategy, FLOPs double as each sample is processed twice during inference.

F. Extension Experiments

1) Evaluation on One-Shot Skeleton-Based Action Recognition: To further evaluate the generalizability of our method, we extend it to the one-shot skeleton-based action recognition setting, where each unseen class is associated with only one labeled skeleton sample during testing.

TABLE IX

EXPLORING SELF-SUPERVISED MODELS IN ZERO-SHOT SCENARIOS. THE EVALUATION IS BASED ON OUR SETTING I. PGFA† REFERS TO OUR METHOD UTILIZING THE COMPLETE DESCRIPTION BUT NOT EMPLOYING OUR PROTOTYPE-GUIDED TEXT FEATURE ALIGNMENT STRATEGY

Method Split: (Seen/Unseen)	Training Framework	NTU-60(%) (55/5)	NTU-120(%) (110/10)	PKU-MMD(%) (46/5)
Supervised [17]	Pretraining&Fixed	73.65	56.12	71.85
Supervised [17]	Pretraining&Finetuning	76.99	57.55	74.63
Skeleton-logoCLR [87]	Pretraining&Fixed	75.01	55.59	73.27
Skeleton-logoCLR [87]	Pretraining&Finetuning	79.30	56.57	77.33
PGFA†(Ours)	End-to-End	**80.45**	**60.22**	**78.34**

TABLE X

CROSS-DATASET EVALUATION. "NTU-60→PKU-MMD" REPRESENTS TRAINING ON NTU-60 AND TESTING ON PKU-MMD, WHILE "PKU-MMD→NTU-60" REPRESENTS TRAINING ON PKU-MMD AND TESTING ON NTU-60

Method	NTU-60→PKU-MMD(%)	PKU-MMD→NTU-60(%)
SMIE [32]	75.07	55.92
PGFA (Ours)	**94.73**	**61.02**

In our framework, adapting to the one-shot scenario is straightforward. Specifically, during testing, we use the available one-shot skeleton samples to generate class prototypes using our skeleton encoder and feature alignment strategy, instead of relying solely on text descriptions. Following [86], we evaluate on the NTU-60, NTU-120, and PKU-MMD datasets, with seen/unseen class splits of 50/10, 100/20, and 41/10, respectively. *For detailed dataset splits, please refer to the supplementary.* As shown in Tab. VIII, our PGFA demonstrates competitive or superior performance compared to existing one-shot approaches [39], [83], [84], [85], [86], highlighting its flexibility and effectiveness in one-shot scenarios.

2) Exploring Self-Supervised Models in Zero-Shot Scenarios: We explore the potential of self-supervised models for zero-shot skeleton-based action recognition. To this end, we re-implement the state-of-the-art self-supervised model, Skeleton-logoCLR [87], replacing the supervised pre-trained skeleton encoder used in the "Pretraining&Fixed" and "Pretraining&Finetuning" training frameworks. The key distinction between these frameworks is whether the pre-trained skeleton encoder is fixed during the subsequent skeleton-text alignment. As shown in Tab. IX, under the "Pretraining&Fixed" framework, using Skeleton-logoCLR for pretraining outperforms using cross-entropy loss for supervised pertaining on average across three datasets, suggesting that self-supervised training may produce more discriminative features. In the "Pretraining&Finetuning" framework, using Skeleton-logoCLR also naturally yields better performance on average. However, our method, which uses the End-to-End framework, achieves the best performance across all datasets. This suggests that supervised and self-supervised pretraining may not be necessary in zero-shot settings. Our End-to-End training framework enables effective skeleton-text alignment while avoiding the additional cost of pretraining.

3) Cross-Dataset Evaluation of Our Method: We further evaluate the generalization of our method through cross-dataset experiments. Since NTU-120 is an extension of NTU-60, using them together for cross-dataset evaluation is less meaningful. Instead, we perform evaluations between NTU-60 and PKU-MMD. Specifically, we train on one dataset and test directly on the other without fine-tuning: (1) training on NTU-60 and testing on PKU-MMD, and (2) training on PKU-MMD and testing on NTU-60. To further validate the stability of our method under cross-dataset settings, we follow a protocol similar to Setting I. We train on three groups of 55 seen classes from NTU-60 (Class Split 1/2/3) and test on three groups of 5 unseen classes from PKU-MMD (Class Split 1/2/3), and vice versa—training on 46 seen classes from PKU-MMD and testing on 5 unseen classes from NTU-60. We report the average accuracy on the predicted unseen classes across all splits. As shown in Tab. X, our method outperforms the baseline SMIE [32] by a significant margin. Specifically, our method achieves an impressive 94.73% average accuracy (NTU-60 →PKU-MMD), even surpassing performance from training on seen classes in PKU-MMD. This could be due to the larger training set in NTU-60, which enables the model to learn more discriminative features. Conversely, the smaller training set in PKU-MMD leads to slightly lower performance when transferring to NTU-60, though our method still outperforms the SMIE baseline. These results strongly demonstrate the transferability of our method.

G. Qualitative Analysis

1) Visualization of Skeleton Feature Space: Existing methods always pre-train skeleton encoders using cross-entropy loss before modality alignment. However, this approach does not ensure the intra-class compactness [72] of skeleton features, which poses challenges for subsequent skeleton-text alignment. To verify this, we compared the t-SNE [88] visualizations of skeleton features between SMIE and our end-to-end training framework. As illustrated in Fig. 7, the intra-class compactness of skeleton features from the existing training framework is inadequate. The use of cross-entropy loss for pre-training, which leads to a lack of intra-class compactness and the possibility of poor margins, has been analyzed in existing works [69], [70], [71], [72]. The fixed distribution of such skeleton features presents challenges in skeleton-text alignment. In contrast, our training framework improves intra-class compactness through cross-modal contrastive training. Additionally, our method simplifies the entire training process without the need to pre-train the skeleton encoder.

2) Fisher Discrimination Ratio of Skeleton Features: We use the Fisher Discrimination Ratio (FDR) [89] to further assess class separability. FDR measures how well-separated

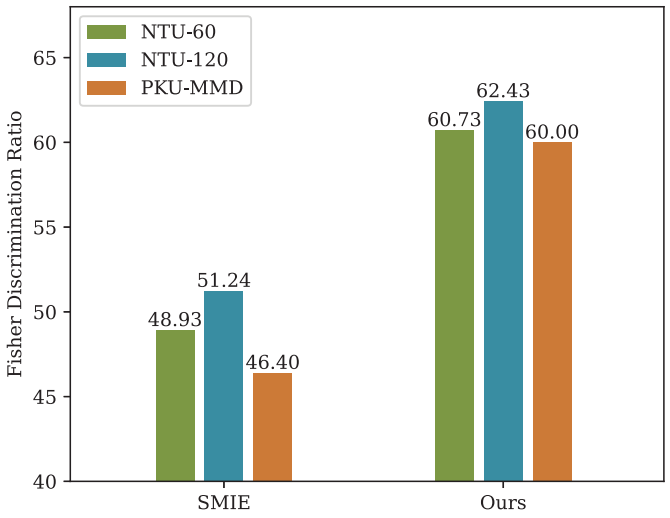


Fig. 8. The average Fisher Discrimination Ratio (FDR) of skeleton features from unseen classes across 3 class splits under Setting I. A higher FDR indicates better class separability.

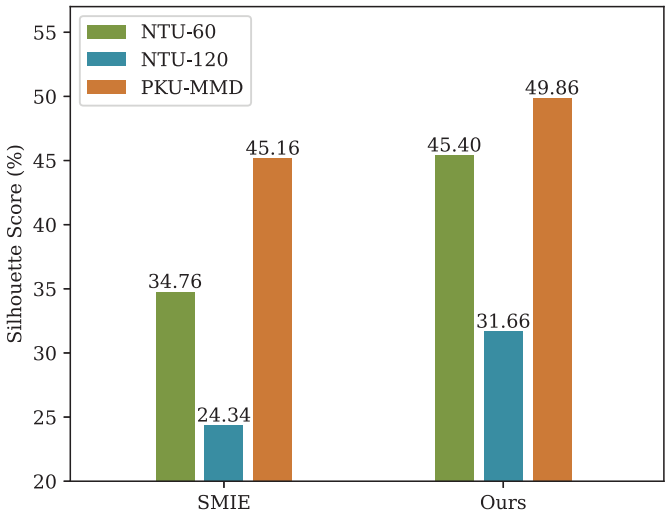
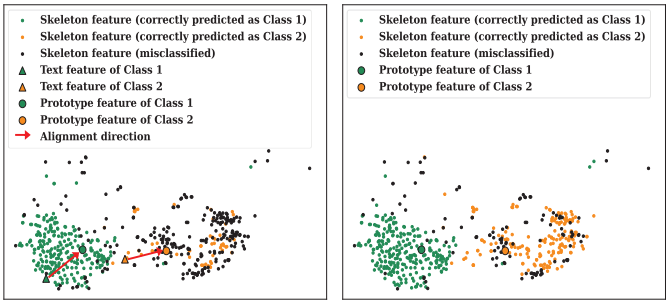


Fig. 9. The average silhouette scores [90] of unseen skeleton features across 3 class splits under Setting I. The silhouette score closer to 1 (e.g.i.e., 100%) indicates well-clustered data, reflecting denser clusters with minimal within-cluster distance.

classes are based on their feature distributions. For multi-class classification, FDR is calculated as the ratio of the between-class scatter to the within-class scatter: FDR = $\text{tr}(S_w^{-1}S_b)$, where S_w is the within-class scatter matrix, S_b is the between-class scatter matrix, and $\text{tr}(\cdot)$ is the trace operator. The S_w is computed as the sum of the covariance matrices for each class, while S_b captures the variance between the class means and the overall mean. A higher FDR indicates better class separability. For each dataset (NTU-60, NTU-120, and PKU-MMD), we calculate the average FDR of skeleton features from unseen classes across three class splits under Setting I. As shown in Fig. 8, our method consistently demonstrates higher FDRs than SMIE across all three datasets, indicating better class separability and discrimination in our approach.

3) Silhouette Score of Skeleton Features: To better demonstrate the clustering differences of unseen skeleton features between SMIE (all previous methods using the same pre-trained skeleton encoder) and our method, we utilize the silhouette score [90] to measure the intrinsic structure of the created clusters. The silhouette score is defined as $S = \frac{1}{N}\sum_i^N \frac{b_i - a_i}{\max(a_i, b_i)}$, where N is the number of samples, a_i is the average distance between sample i and all other samples within the same cluster, and b_i is the average distance from i to all samples in the nearest neighboring cluster. The silhouette score S ranges from -1 to 1. The S closer to 1 indicates well-clustered data, reflecting denser clusters with minimal within-cluster distance. In this case, we use cosine distance for computation. For each dataset (NTU-60, NTU-120, and PKU-MMD), we calculate the average silhouette score for skeleton features from unseen classes across three class splits under Setting I. The cluster labels correspond to the true unseen class labels. As shown in Fig. 9, our method consistently demonstrates higher silhouette scores than SMIE across all datasets. It demonstrates the effectiveness of our training framework in clustering unseen skeleton features, indicating that our method better captures the inherent structure and similarities within the skeleton data.



(a) Before Alignment (b) After Alignment

Fig. 10. Visualization of prototype-guided text feature alignment in feature space. Class 1 and Class 2 represent the unseen action categories "Typing Keyboard" and "Check Time", respectively. We visualize the skeleton features of these two unseen classes and their corresponding text features using t-SNE [88]. Through our prototype-guided text feature alignment strategy, many previously misclassified samples are correctly predicted as Class 2.

4) Visualization of Prototype-Guided Text Feature Alignment in Feature Space: To further demonstrate the effectiveness of our prototype-guided text feature alignment strategy, we visualize this strategy in the feature space. To enhance the clarity of the visualization, we select the unseen classes "Typing Keyboard" and "Check Time" from Class Split 2 of NTU-60 under Setting I *(for the full confusion matrices, providing a more intuitive view of each class accuracy, please refer to the supplementary)*. The skeleton features of these two unseen classes and their corresponding text features are projected into a 2D space using t-SNE [88], as shown in Fig. 10(a). Through our prototype-guided text feature alignment strategy, we obtain the prototype features for the corresponding unseen classes. These prototype features are used in place of text features for similarity calculations with skeleton features to produce the final predictions. As shown in Fig. 10(b), many misclassified samples are corrected. This demonstrates that our prototype-guided text feature alignment strategy can effectively utilize the distribution of unseen skeleton features to derive prototype features and rectify classification errors.

V. Discussion and Conclusion

In this paper, we propose a prototype-guided feature alignment (PGFA) paradigm to address issues of insufficient discrimination and alignment bias during training and testing in previous zero-shot skeleton-based action recognition methods. Our PGFA paradigm comprises: 1) an end-to-end contrastive training framework to ensure sufficient discrimination for skeleton features, and 2) a prototype-guided text feature alignment strategy to alleviate alignment bias between skeleton and unseen text features during testing. Experimental results on NTU-60, NTU-120, and PKU-MMD datasets demonstrate the superior performance of PGFA in zero-shot action recognition.

Future Work: Although our prototype-guided text feature alignment strategy is highly effective, it requires classifying all test samples twice, making it unsuitable for real-time online scenarios where models cannot know all test samples in advance. Modifications may be necessary for such scenarios. For instance, creating a continuously updated "prototype bank" could allow for real-time updates of prototype features by calculating the predicted label of each single or batch sample during testing. We believe this is a promising and feasible direction and plan to explore it in future work.

References

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[2] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[3] L. Wang et al., "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[4] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6299–6308.

[5] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Mali, Oct. 2019, pp. 6202–6211.

[6] T. Zhan, Y. Song, J. Wang, and L. Wang, "VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 10078–10093.

[7] C. Li et al., "Deep manifold structure transfer for action recognition," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4646–4658, Sep. 2019.

[8] Y. Liu, K. Wang, G. Li, and L. Lin, "Semantics-aware adaptive knowledge distillation for sensor-to-vision action recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 5573–5588, 2021.

[9] J. Qin, L. Liu, Z. Zhang, Y. Wang, and L. Shao, "Compressive sequential learning for action similarity labeling," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 756–769, Feb. 2016.

[10] Y. Tang, J. Lu, Z. Wang, M. Yang, and J. Zhou, "Learning semantics-preserving attention and contextual interaction for group activity recognition," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4997–5012, Oct. 2019.

[11] W. Liu, X. Zhong, Z. Zhou, K. Jiang, Z. Wang, and C.-W. Lin, "Dual-recommendation disentanglement network for view fuzz in action recognition," *IEEE Trans. Image Process.*, vol. 32, pp. 2719–2733, 2023.

[12] B. Reily, F. Han, L. E. Parker, and H. Zhang, "Skeleton-based bio-inspired human activity prediction for real-time human–robot interaction," *Auto. Robots*, vol. 42, no. 6, pp. 1281–1298, Aug. 2018.

[13] C. Bandi and U. Thomas, "Skeleton-based action recognition for human–robot interaction using self-attention mechanism," in *Proc. 16th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Dec. 2021, pp. 1–8.

[14] J. Yin, J. Han, C. Wang, B. Zhang, and X. Zeng, "A skeleton-based action recognition system for medical condition detection," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2019, pp. 1–4.

[15] N. Noor and I. K. Park, "A lightweight skeleton-based 3D-CNN for real-time fall detection and action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 2171–2180.

[16] A. Elaoud, W. Barhoumi, E. Zagrouba, and B. Agrebi, "Skeleton-based comparison of throwing motion for handball players," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 1, pp. 419–431, Jan. 2020.

[17] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 7444–7452.

[18] H. Wang and L. Wang, "Beyond joints: Learning representations from primitive geometries for skeleton-based action recognition and detection," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4382–4394, Sep. 2018.

[19] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12026–12035.

[20] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1963–1978, Aug. 2019.

[21] 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 *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 55–63.

[22] X. Hao, J. Li, Y. Guo, T. Jiang, and M. Yu, "Hypergraph neural network for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 2263–2275, 2021.

[23] K. Cheng, Y. Zhang, X. He, J. Cheng, and H. Lu, "Extremely lightweight skeleton-based action recognition with ShiftGCN++," *IEEE Trans. Image Process.*, vol. 30, pp. 7333–7348, 2021.

[24] 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 13359–13368.

[25] H.-G. Chi, M. Ha, S. Chi, S. Lee, Q. Huang, and K. Ramani, "InfoGCN: Representation learning for human skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 20154–20164.

[26] H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 2969–2978.

[27] Y. Zhu, H. Shuai, G. Liu, and Q. Liu, "Multilevel spatial–temporal excited graph network for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 32, pp. 496–508, 2023.

[28] W. Myung, N. Su, J.-H. Xue, and G. Wang, "DeGCN: Deformable graph convolutional networks for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 33, pp. 2477–2490, 2024.

[29] X. Xu, T. Hospedales, and S. Gong, "Transductive zero-shot action recognition by word-vector embedding," *Int. J. Comput. Vis.*, vol. 123, no. 3, pp. 309–333, Jul. 2017.

[30] B. Jasani and A. Mazagonwalla, "Skeleton based zero shot action recognition in joint pose-language semantic space," 2019, *arXiv:1911.11344*.

[31] P. Gupta, D. Sharma, and R. K. Sarvadevabhatla, "Syntactically guided generative embeddings for zero-shot skeleton action recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 439–443.

[32] Y. Zhou, W. Qiang, A. Rao, N. Lin, B. Su, and J. Wang, "Zero-shot skeleton-based action recognition via mutual information estimation and maximization," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 5302–5310.

[33] A. Zhu, Q. Ke, M. Gong, and J. Bailey, "Part-aware unified representation of language and skeleton for zero-shot action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 18761–18770.

[34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[35] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[36] M. Wang, J. Xing, J. Mei, Y. Liu, and Y. Jiang, "ActionCLIP: Adapting language-image pretrained models for video action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 625–637, Jan. 2025.

[37] L. Xue et al., "ULIP: Learning a unified representation of language, images, and point clouds for 3D understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 1179–1189.

[38] D. Hegde, J. M. J. Valanarasu, and V. M. Patel, "CLIP goes 3D: Leveraging prompt tuning for language grounded 3D recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 2020–2030.

[39] J. Wang and Y. Zhai, "Prototypical Siamese networks for few-shot learning," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 178–181.

[40] W. Huang, B. Xiao, J. Hu, and X. Bi, "Location-aware transformer network for few-shot medical image segmentation," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2023, pp. 1150–1157.

[41] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2001–2010.

[42] H. Lin et al., "Prototype-guided continual adaptation for class-incremental unsupervised domain adaptation," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 351–368.

[43] Y. Iwasawa and Y. Matsuo, "Test-time classifier adjustment module for model-agnostic domain generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 2427–2440.

[44] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1110–1118.

[45] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 4263–4270.

[46] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive recurrent neural networks for high performance human action recognition from skeleton data," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2117–2126.

[47] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "PoTion: Pose MoTion representation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7024–7033.

[48] C. Caetano, J. Sena, F. Brémond, J. A. Dos Santos, and W. R. Schwartz, "SkeleMotion: A new representation of skeleton joint sequences based on motion information for 3D action recognition," in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2019, pp. 1–8.

[49] K. Xu, F. Ye, Q. Zhong, and D. Xie, "Topology-aware convolutional neural network for efficient skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 3, pp. 2866–2874.

[50] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 183–192.

[51] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[52] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[53] C. Plizzari, M. Cannici, and M. Matteucci, "Spatial temporal transformer network for skeleton-based action recognition," in *Proc. ICPR Int. Workshops Challenges*, 2021, pp. 694–701.

[54] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Decoupled spatial–temporal attention network for skeleton-based action-gesture recognition," in *Proc. Asian Conf. Comput. Vis.*, 2021, pp. 38–53.

[55] Y. Zhang, B. Wu, W. Li, L. Duan, and C. Gan, "STST: Spatial–temporal specialized transformer for skeleton-based action recognition," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3229–3237.

[56] A. Frome et al., "DeViSE: A deep visual-semantic embedding model," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 2121–2129.

[57] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[58] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, Dec. 2013, pp. 3111–3119.

[59] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Language Process. 9th Int. Joint Conf. Natural Language Process. (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[60] Y.-H.-H. Tsai, L.-K. Huang, and R. Salakhutdinov, "Learning robust visual-semantic embeddings," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3591–3600.

[61] M. Wray, G. Csurka, D. Larlus, and D. Damen, "Fine-grained action retrieval through multiple parts-of-speech embeddings," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 450–459.

[62] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero-and few-shot learning via aligned variational autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 8247–8255.

[63] C. Jia et al., "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, pp. 4904–4916.

[64] X. Wang, Y. Yan, H.-M. Hu, B. Li, and H. Wang, "Cross-modal contrastive learning network for few-shot action recognition," *IEEE Trans. Image Process.*, vol. 33, pp. 1257–1271, 2024.

[65] R. Zhang et al., "PointCLIP: Point cloud understanding by CLIP," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8552–8562.

[66] G. Tevet, B. Gordon, A. Hertz, A. H. Bermano, and D. Cohen-Or, "MotionCLIP: Exposing human motion generation to CLIP space," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 358–374.

[67] W. Xiang, C. Li, Y. Zhou, B. Wang, and L. Zhang, "Generative action description prompts for skeleton-based action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 10276–10285.

[68] H.-M. Xu, L. Liu, Q. Bian, and Z. Yang, "Semi-supervised semantic segmentation with prototype-based consistency regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 26007–26020.

[69] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 507–516.

[70] G. F. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio, "Large margin deep networks for classification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 842–852.

[71] P. Khosla et al., "Supervised contrastive learning," in *Proc. NIPS*, 2020, pp. 18661–18673.

[72] Z. Shi, H. Wang, and C.-S. Leung, "Constrained center loss for convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 1080–1088, Feb. 2023.

[73] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[74] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NIPS*, 2020, pp. 1877–1901.

[75] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von Mises-Fisher distributions," *J. Mach. Learn. Res.*, vol. 6, no. 46, pp. 1345–1382, 2005.

[76] A. T. A. Wood, "Simulation of the von Mises Fisher distribution," *Commun. Statist. - Simul. Comput.*, vol. 23, no. 1, pp. 157–164, Jan. 1994.

[77] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1010–1019.

[78] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "NTU RGB+D 120: A large-scale benchmark for 3D human activity understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2684–2701, Oct. 2020.

[79] J. Liu, S. Song, C. Liu, Y. Li, and Y. Hu, "A benchmark dataset and comparison study for multi-modal human action analytics," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 2, pp. 1–24, May 2020.

[80] L. Li, M. Wang, B. Ni, H. Wang, J. Yang, and W. Zhang, "3D human action representation learning via cross-view consistency pursuit," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4741–4750.

[81] S. Hu et al., "MiniCPM: Unveiling the potential of small language models with scalable training strategies," 2024, *arXiv:2404.06395*.

[82] DeepSeek-AI et al., "DeepSeek-V3 technical report," 2024, *arXiv:2412.19437*.

[83] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8808–8817.

[84] C. Simon, P. Koniusz, R. Nock, and M. Harandi, "Adaptive subspaces for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4136–4145.

[85] C. Xu et al., "Learning dynamic alignment via meta-filter for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5182–5191.

[86] S. Yang, J. Liu, S. Lu, E. M. Hwa, and A. C. Kot, "One-shot action recognition via multi-scale spatial–temporal skeleton matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 7, pp. 5149–5156, Jul. 2024.
[87] J. Hu, Y. Hou, Z. Guo, and J. Gao, "Global and local contrastive learning for self-supervised skeleton-based action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 11, pp. 10578–10589, Nov. 2024.
[88] L. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
[89] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Hoboken, NJ, USA: Wiley, 2005.
[90] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

Kai Zhou received the B.E. degree in software engineering from the School of Software Engineering, South China University of Technology, China, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include deep learning and computer vision.

Shuhai Zhang is currently pursuing the Ph.D. degree with the South China University of Technology, China. He has published papers in *Neural Networks*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, ICCV, ICML, ICLR, and CVPR. His research interests are broadly in machine learning and mainly focus on large language model, model compression, and adversarial robust.

Zeng You received the B.E. degree in software engineering from the School of Software Engineering, South China University of Technology, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Future Technology. His research interests include deep learning, video understanding, and computer vision.

Jinwu Hu (Graduate Student Member, IEEE) received the B.E. degree from Foshan University, Foshan, China, in 2020, and the M.S. degree from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2023. He is currently pursuing the Ph.D. degree with the South China University of Technology, Guangzhou, China. He has published several journal/conference papers, including IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON MEDICAL IMAGING, IEEE TRANSACTIONS ON BIG DATA, ICML, IJCAI, CVPR, and ACM MM. His research interests include computer vision, machine learning, large language models, and reinforcement learning. He has served as a reviewer for many academic journals, including IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, NN, and PR.

Mingkui Tan (Senior Member, IEEE) received the bachelor's degree in environmental science and engineering and the master's degree in control science and engineering from Hunan University, Changsha, China, in 2006 and 2009, respectively, and the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2014. He is currently a Professor with the School of Software Engineering, South China University of Technology. From 2014 to 2016, he was a Senior Research Associate in computer vision with the School of Computer Science, University of Adelaide, Australia. His research interests include machine learning, sparse analysis, deep learning, and large-scale optimization.

Fei Liu (Member, IEEE) received the B.E. degree in automatic testing and control and the M.E. and Ph.D. degrees in control science and engineering from Harbin Institute of Technology. He was with the Department of Computer Science, Brandenburg University of Technology, from 2009 to 2012, and the Department of Computer, Yamaguchi University, from 2015 to 2016, as a JSPS Scholar. He is currently a Professor with the School of Software Engineering, South China University of Technology. His research interests include modeling and simulation, artificial intelligence, and systems biology.