Generative Action Description Prompts for Skeleton-based Action Recognition

Wangmeng Xiang[1,2] Chao Li[2] Yuxuan Zhou[2,3] Biao Wang[2] Lei Zhang[1*]
[1]The Hong Kong Polytechnic University [2]DAMO Academy, Alibaba Group [3]Mannheim University
{wangmeng.xwm,lllcho.lc,wb.wangbiao}@alibaba-inc.com, yuxuazho@mail.uni-mannheim.de
cslzhang@comp.polyu.edu.hk

Abstract

*Skeleton-based action recognition has recently received considerable attention. Current approaches to skeleton-based action recognition are typically formulated as one-hot classification tasks and do not fully exploit the semantic relations between actions. For example, "make victory sign" and "thumb up" are two actions of hand gestures, whose major difference lies in the movement of hands. This information is agnostic from the categorical one-hot encoding of action classes but could be unveiled from the action description. Therefore, utilizing action description in training could potentially benefit representation learning. In this work, we propose a **G**enerative **A**ction-description **P**rompts (GAP) approach for skeleton-based action recognition. More specifically, we employ a pre-trained large-scale language model as the knowledge engine to automatically generate text descriptions for body parts movements of actions, and propose a multi-modal training scheme by utilizing the text encoder to generate feature vectors for different body parts and supervise the skeleton encoder for action representation learning. Experiments show that our proposed GAP method achieves noticeable improvements over various baseline models without extra computation cost at inference. GAP achieves new state-of-the-arts on popular skeleton-based action recognition benchmarks, including NTU RGB+D, NTU RGB+D 120 and NW-UCLA. The source code is available at https://github.com/MartinXM/GAP.*

1. Introduction

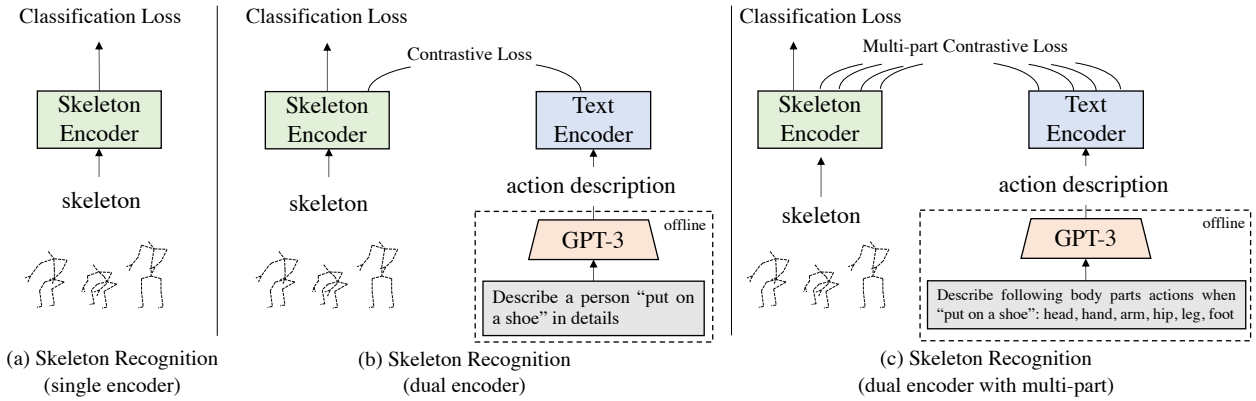
Action recognition has been an active research topic due to its wide range of applications in human-computer interaction, sports and health analysis, entertainment, *etc*. In recent years, with the emergence of depth sensors, such as Kinect [44] and RealSense [14], human body joints can be easily acquired. The action recognition approach utilizing body joints, *i.e*., the so-called skeleton-based action recognition, has drawn a lot of attentions due to its computation efficiency and robustness to lighting conditions, viewpoint variations and background noise.

Most of the previous methods in skeleton-based action recognition focus on modeling the relation of human joints, following a unimodal training scheme with a sequence of skeleton coordinates as inputs [41, 15, 27, 9, 28, 4, 25, 40, 30, 36, 35, 22]. Inspired by the recent success of multi-modal training with image and language [23, 1], we investigate an interesting question: whether action language description could unveil the action relations and benefit skeleton-based action recognition? Regrettably, due to the absence of a large-scale dataset consisting of skeleton-text pairs, constructing such a dataset would require significant time and financial resources. Consequently, the training scheme outlined in [23, 11, 39] cannot be directly applied to skeleton-based action recognition. As a result, the development of novel multi-modal training paradigms is necessary to address this issue.

We propose to leverage the generative category-level human action description in the form of language prompts. The language definition of an action contains rich prior knowledge. For example, different actions focus on the movement of different body parts: "make victory sign" and "thumb up" describe the gesture of hands; "arm circles" and "tennis bat swing" describe the movement of arms; "nod head" and "shake head" are the motions of head; "jump up" and "side kick" rely on movements of foot and leg. Some actions describe the interaction of multiple body parts, *e.g*., "put on a hat" and "put on a shoe" involve actions of hand and head, hand and foot, respectively. These prior knowledge about actions could provide fine-grained guidance for representation learning. In addition, to resolve the laborious work to collect human action prompts, we resort to pre-trained large language model (LLM), e.g. GPT-3 [1] for efficient automatic prompts generation.

In specific, we develop a new training paradigm, which employs generative action prompts for skeleton-based action recognition. We take advantages of the GPT-3 [1] as

*Corresponding author



(a) Skeleton Recognition (single encoder)

(b) Skeleton Recognition (dual encoder)

(c) Skeleton Recognition (dual encoder with multi-part)

Figure 1: Comparison of our proposed Generative Action-description Prompts (GAP) framework (dual encoder) with other skeleton recognition methods (single encoder). Besides classification loss, our proposed method contains additional contrastive loss. Notice that text encoder is only used at the training stage and GPT-3 is applied for offline action description generation. For every given action query, GPT-3 generates text description of actions with prompt templates, the action description is then employed for multi-modal training.

our knowledge engine to generate meaningful text descriptions for actions. With elaborately designed text prompts, detailed text descriptions for the whole action and each body part can be produced. In Figure 1, we compare our proposed frameworks (b) and (c) with traditional single encoder skeleton-based action recognition framework (a). In our framework, a multi-modal training scheme is developed, which contains a skeleton encoder and a text encoder. The skeleton encoder takes skeleton coordinates as inputs and generates both part feature vectors and global feature representations. The text encoder transforms global action description or body part descriptions into text features for the whole action or each body part. A multi-part contrastive loss (single contrastive loss for (b)) is used to align the text part features and skeleton part features, and the cross-entropy loss is applied on the global features.

Our contributions are summarized as follow:

- As far as we known, this is the first work to use generative prompts for skeleton-based action recognition, which applies a LLM as the knowledge engine and elaborately employs text prompts to generate detailed text descriptions of the whole action and body parts movements for different actions automatically.
- We propose a new multi-modal training paradigm that utilizes generative action prompts to guide skeleton-based action recognition, which enhances the representation by using knowledge about actions and human body parts. It could improve the model performance without bringing any computation cost at inference.
- With the proposed training paradigm, we achieve state-of-the-art performance on several popular skeleton-based action recognition benchmarks, including NTU RGB+D, NTU RGB+D 120 and NW-UCLA.

2. Related work

2.1. Skeleton-based Action Recognition

In recent years, various methods have been proposed for skeleton-based action recognition by designing efficient and effective model architecture. RNNs were applied to handle the sequence of human joints in [9, 28, 41]. HBRNN [9] employed an end-to-end hierarchical RNN to model long-term contextual information of temporal skeleton sequences. VA-LSTM [41] designed a view adaptive RNN, which enables the network to adapt to the most suitable observation viewpoints from end to end. Inspired by the success of CNN in image tasks, CNN-based methods [42, 37] have been utilized to model joints relations. A pure CNN architecture named Topology-aware CNN (TA-CNN) is proposed in [37]. As human joints can be naturally presented as graph nodes and joint connections can be described by adjacent matrix, GCN-based methods [38, 4, 25, 2, 30] have drawn a lot of attentions. For example, ST-GCN [38] applied spatial-temporal GCN to model human joints relations in both spatial and temporal dimension. CTR-GCN [2] proposed a channel-wise graph convolution for fine-grained relation modeling. Info-GCN [6] adopt an information bottleneck in GCN. With the recent popularity of vision transformer [8], transformer-based methods [22, 26, 35] have also been investigated for skeleton data. All the previous methods adopt a unimodal training scheme. As far as we known, our work is the first to apply a multi-modal training scheme for skeleton-based action recognition.

2.2. Human Part Prior

Human part prior for skeleton-based action recognition has been used by designing special model architectures in

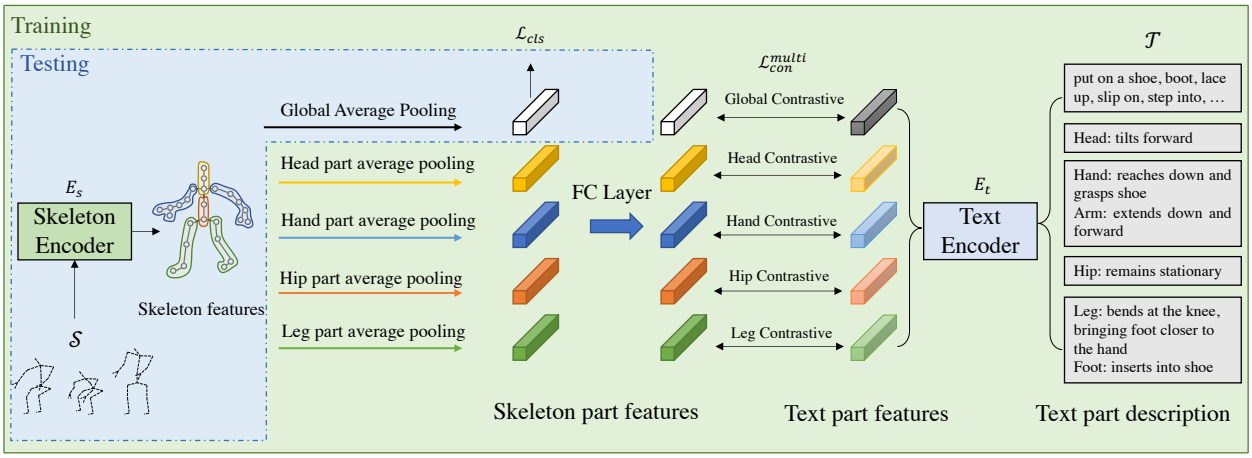


Figure 2: Overall framework of Generative Action-description Prompts (GAP) with multi-part contrastive loss. During training, the skeleton encoder is trained with both cross-entropy loss and multi-part contrastive loss. We use GPT-3 generated action description as input for text encoder to generate human part features. The part features are then aligned with skeleton encoder part features with multi-part contrastive loss. During testing, only global feature from skeleton encoder is used for classification, and text encoder is neglected.

previous works [32, 29, 35, 10]. PB-GCN [32] divided the skeleton graph into four subgraphs and learned a recognition model using a part-based graph convolutional network. PA-ResGCN [29] calculated attention weights for human body parts to improve the discriminative capability of the features. PL-GCN [10] proposed a part-level graph convolutional network to automatically learn the part partition strategy. IIP-transformer [35] applied transformer to learn inter-part and intra-part relations. Comparing to previous methods, we directly use part language description to guide representation learning during training with a multi-part contrastive loss. We do not design any complicated part modeling module and thus do not introduce extra computation cost at inference.

2.3. Multi-modal Representation Learning

Multimodal representation learning methods, such as CLIP [23] and ALIGN [11], have shown that vision-language co-training can learn powerful representation for downstream tasks such as zero-shot learning, image captioning, text-image retrieval, *etc*. UniCL [39] uses a unified contrastive learning method that regards image-label as image-text-label data to learn the generic visual-semantic space. However, these methods require a large-scale image-text paired dataset for training. ActionCLIP [34] follows the training scheme of CLIP for video action recognition. A pre-trained CLIP model is used and transformer layers are added for temporal modeling of video data. As for action description, label names are directly used as text prompts with prefix and suffix that do not contain much semantic meanings, *e.g.*,*"A video of [action name]"*, *"Human action of [action name]"*, *etc*. In contrast, we use a LLM (GPT-3), as knowledge engine to generate descriptions of human body movements in actions, which provide fine-grained guidance for representation learning. In addition, we employ multi-part contrastive loss on body parts to learn a fine-grained skeleton representation. Prompt Learning (PL) [46, 45, 12] approaches aim to tackle the challenges posed by zero-shot and few-shot learning by through the incorporation of learnable prompt vectors. While PL has demonstrated promising results, the interpretability of the learned prompt vectors remains a challenge. Recently, [20] applies LLM for generating descriptions for zero-shot image classification. STALE [21] applies parallel classification and localization/classification architecture for zero-shot action detection. MotionCLIP [31] is proposed to align action latent space with CLIP latent space for 3D human action generation. ActionGPT [13] uses LLM to generate detailed action description for action generation. Our research is conducted concurrently and independently. All these methods require a text encoder during inference, whereas our proposed framework only imposes overheads during the training phase, without adding any computational or memory costs during testing.

3. Methods

In this section, we present in detail the proposed **G**enerative **A**ction-description **P**rompts (GAP) framework. GAP aims to enhance skeleton representation learning with automatically generated action descriptions and it can be embedded into the existing backbone networks. Therefore, GAP can be coupled with various skeleton and language encoders. In the following sections, we first overview the GAP framework, then introduce the skeleton encoder, text

encoder and the main components of GAP in detail.

3.1. Generative Action Prompts Framework

The comprehensive framework of our GAP approach is presented in Figure 2. It is composed of a **skeleton encoder** E_s and a **text encoder** E_t, for generating skeleton features and text features, respectively. The training loss can be presented as:

$$\mathcal{L}_{total} = \mathcal{L}_{cls}(E_s(\mathcal{S})) + \lambda \mathcal{L}_{con}^{multi}(E_s(\mathcal{S}), E_t(\mathcal{T})), \quad (1)$$

where, $\mathcal{L}_{cls}$ is cross-entropy classification loss, $\mathcal{L}_{con}^{multi}$ is multi-part contrastive loss. Skeleton input $\mathcal{S} \in \mathbb{R}^{B\times 3\times N\times T}$, B is the batch size, 3 is the coordinate number, N and T are joint number and sequence length, respectively. λ is a learnable trade-off parameter. $\mathcal{T}$ is LLM generated text descriptions.

During training, the E_s is trained with cross-entropy loss and multi-part contrastive loss with part text descriptions as additional guidance. The global skeleton feature is generated by performing average pooling of all joint nodes and the part skeleton features are generated by aggregating the features of various groups of nodes using average pooling. The skeleton part features are mapped by fully connected layer (FC Layer) to keep the same feature dimension as text features. The text part descriptions are generated by LLM offline, and encoded by E_t during training for producing text part features. At the testing stage, we directly use global features of skeleton encoder for action probability prediction. Therefore, our GAP framework does not bring additional memory or computation cost at inference comparing to previous skeleton encoder only method.

3.2. Skeleton Encoder

Graph Convolution Network (GCN) is prevailing for skeleton action recognition due to its efficiency and strong performance. Therefore, we adopt GCN as the backbone network in our GAP framework. Our skeleton encoder consists of multiple GC-MTC blocks, while each block contains a graph convolution (GC) layer and a multiscale temporal convolution (MTC) module.

Graph Convolution. The human skeleton can be represented as a graph $G = \{V, \mathcal{E}\}$, where V is the set of human joints with $|V| = N$, and $\mathcal{E}$ is the set of edges. Denote by $\mathbf{H}^l \in \mathbb{R}^{N\times F}$ the features of human joints at layer l with feature dimension F. The graph convolution can be formulated as follows:

$$\mathbf{H}^{l+1} = \sigma(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^l\mathbf{W}^l), \quad (2)$$

where $\mathbf{D} \in \mathbb{R}^{N\times N}$ is the degree matrix, $\mathbf{A}$ is the adjacency matrix representing joints connections, $\mathbf{W}^l$ is the learnable parameter of the l-th layer and σ is the activation function.

Multiscale Temporal Modeling. To model the action at different temporal speed, we utilize the multiscale temporal convolution module in [19, 2] for temporal modeling. The module comprises four distinct branches, each of which incorporates a 1×1 convolution to decrease channel dimensionality. There are two temporal convolutions branches with varing dilations (1 and 2) and one MaxPool branch. The fourth branch only contains 1×1 convolution. The outputs of the four branches are concatenated to produce the final result.

Skeleton Classification. The skeleton-based action recognition methods map human skeleton data to one-hot encoding of action labels, which are trained with a cross-entropy loss:

$$\mathcal{L}_{cls} = -y \log p_\theta(x), \quad (3)$$

where y is the one-hot ground-truth action label, x is the global skeleton feature and $p_\theta(x)$ is the predicted probability distribution.

3.3. Text Encoder

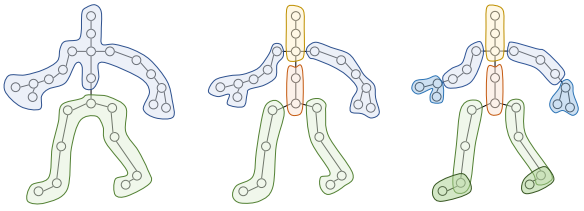
Considering the recent success of Transformer models in NLP, we employ a pre-trained transformer-based language model as our text encoder E_t, such as BERT [7] or CLIP-text-encoder [23]. The input is in the form of text and undergoes a standard tokenization process. Subsequently, the features are processed through a series of transformer blocks. The final output is a feature vector that represents the text description. For different human part, we use various part descriptions as text encoder's input.

3.4. Action Description Learning

Skeleton-language Contrastive Learning. Comparing to the one-hot label supervision for skeleton classification, skeleton-language contrastive learning employs the supervision from natural language. It has a dual-encoder design with a skeleton encoder E_s and a text encoder E_t, which encode skeleton data and action descriptions, respectively. The dual-encoders are jointly optimized by contrasting skeleton-text pairs in two directions within the batch:

$$\begin{aligned} p_i^{s2t}(\mathbf{s_i}) &= \frac{\exp(\text{sim}(\mathbf{s_i}, \mathbf{t_i})/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(\mathbf{s_i}, \mathbf{t_j})/\tau)}, \\ p_i^{t2s}(\mathbf{t_i}) &= \frac{\exp(\text{sim}(\mathbf{t_i}, \mathbf{s_i})/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(\mathbf{t_i}, \mathbf{s_j})/\tau)}, \end{aligned} \quad (4)$$

where $\mathbf{s}$, $\mathbf{t}$ are encoded features of skeleton and text, $\text{sim}(\mathbf{s}, \mathbf{t})$ is the cosine similarity, τ is the temperature parameter and B is the batch size. Unlike image-text pairs in CLIP, which are one-to-one mappings, in our setting, there could be more than one positive matching and actions of different categories forming negative pairs. Therefore, instead of using cross-entropy loss, we use KL divergence as



(a) Two partition (b) Four partition (b) Six partition

Figure 3: Different part partition strategies. (a) Two parts: upper and lower body. (b) Four parts: head, hand-arm, hip, leg-foot. (c) Six parts: head, arm, hand, hip, leg, foot.

Label name: Prefix: "put on a shoe", a video of action Prefix: "put on a shoe", this is an action Cloze: This is "put on a shoe", a video of action Suffix: Human action of "put on a shoe" Suffix: Playing a kind of action, "put on a shoe"…	**GPT-3 (Paragraph):** Put on a shoe: The man is putting on a shoe. He is bending down and putting his foot into the shoe. He is then tying the shoe. He is doing this quickly and efficiently.
HAKE: Put on a shoe: foot stand on, foot walk to, foot fall down, hand put on, foot tread on	**GPT-3 (Synonym):** Put on a shoe: boot, lace up, slip on, step into, strap on, tie, tuck in, zip up, don, fasten
Manual: Put on a shoe: hand reach for, hand put on, hip sit on, leg bend down, foot wear	**GPT-3 (Part description):** Put on a shoe: head tilts slightly forward; hand reaches down and grasps shoe; arm extends down and forward; hip remains stationary; leg bends at the knee, bringing foot closer to the hand; foot inserts into shoe.

Figure 4: Text description generated by different methods.

the skeleton-text contrastive loss:

$$\mathcal{L}_{con} = \frac{1}{2}\mathbf{E}_{\mathbf{s},\mathbf{t}\sim\mathcal{D}}[KL(p^{s2t}(\mathbf{s}), y^{s2t}) + KL(p^{t2s}(\mathbf{t}), y^{t2s})], \tag{5}$$

where $\mathcal{D}$ is the entire dataset, y^{s2t} and y^{t2s} are ground-truth similarity scores, which have a probability of 0 for negative pairs and a probability of 1 for positive.

Multi-part Contrastive Learning. Considering the prior of human body parts, skeleton can be divided into multiple groups. We illustrate this framework in Figure 1(c). We apply contrastive loss on different parts features as well as global feature, and propose a multi-part contrastive loss. The part feature could be obtained with part pooling, where joint features within the same group are aggregated to generate part representation. More specifically, we choose the features before the final classification layer for part feature pooling. In Figure 3, we show different part partition strategies. For two parts partition, the whole body is divided into upper and lower groups. For four parts partition, the body is divided into four groups: head, hand-arm, hip, leg-foot. For six parts partition, head, hand, arm, hip, leg, foot are grouped separately. The loss function of multi-part contrastive loss can be represented as follows:

$$\mathcal{L}_{con}^{multi} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{con}^{k}, \tag{6}$$

where K is the total part number.

3.5. Action Description Generation

The action description $\mathcal{T}$ for text encoder plays a vital role in GAP. Here, we explore several different description generation methods. Figure 4 illustrates the text descriptions of action "put on a shoe" by different methods.

Paragraph
Q: Describe a person "put on a shoe" in details.
A: The man is putting on a shoe. He is bending down and putting his foot into the shoe. He is then tying the shoe. He is doing this quickly and efficiently.
Synonym
Q: Suggest 10 synonyms for "put on a shoe"
A: boot, lace up, slip on, step into, strap on, tie, tuck in, zip up, don, fasten
Part Description
Q: Describing following body parts actions when "put on a shoe": head, hand, arm, hip, leg, foot.
A: head tilts slightly forward; hand reaches down and grasps shoe; arm extends down and forward; hip remains stationary; leg bends at the knee, bringing foot closer to the hand; foot inserts into shoe.

Figure 5: Text description generated from different prompt inputs by GPT-3.

Label Name. One straight-forward approach is to directly use the label name. Many methods [34] use this kind of text descriptions with prefix and suffix such as "Human action of [action]", "[action], a video of action", *etc*. Though these prompts could boost the performance for zero-shot and few-shot problems, in our case of supervised learning, this approach does not bring significant performance improvement (as shown in our ablation studies) since these prompts do not contain discriminative semantic information about actions.

HAKE Part State. The HAKE [17] dataset contains annotated part states of human-object interactions. For each sample, six body part movements (head, hand, arm, hip, leg, foot) are manually annotated, with 93 part states in total. In order to avoid laborious annotation for each sample, we apply an automatic pipeline which contains two steps: 1) generate text features for both label name and HAKE part states with a pre-trained transformer text encoder; 2) generate text description by finding the K nearest neighbors of action label name in HAKE part state feature space. Those HAKE part states that are closest to the action label name are selected for action description. We then use this generated part description for GAP.

Manual Description. We ask annotators to write down the description of body part movements following the temporal order of the action. The descriptions consist of the predefined atomic movements. The annotators are asked to focus on the most distinguished parts' motions.

Large-language Model. We use the large-scale language model (*e.g.*, GPT-3) to generate text descriptions. We design text prompts so that it can generate our desired action descriptions. Text descriptions are generated in three

ways. a) *paragraph*: a full paragraph that can describe the action in detail; b) *synonym*: we collect 10 synonyms of action labels; c) *part description*: we collect descriptions of different body parts for each action. The body partition strategies follow Figure 3 in previous section. We take "put on a shoe" as an example and present the prompts used for generating different descriptions in Figure 5.

4. Experiments

4.1. Datasets

NTU RGB+D [24] is a widely used dataset for skeleton-based human action recognition. It contains 56,880 skeletal action sequences. There are two benchmarks for evaluation, including Cross-Subject (X-Sub) and Cross-View (X-View) settings. For X-Sub, the training and test sets come from two disjoint sets, each having 20 subjects. For X-View, the training set contains 37,920 samples captured by camera views 2 and 3, and the test set includes 18,960 sequences captured by camera view 1.

NTU RGB+D 120 [18] is an extension of NTU RGB+D dataset with 57,367 additional skeleton sequences over 60 additional action classes. There are 120 action classes in total. Two benchmark evaluations were suggested by the authors, including Cross-Subject (X-Sub) and Cross-Setup (X-Setup) settings.

NW-UCLA [33] dataset is recorded by three Kinect V1 sensors from different viewpoints. The skeleton contains 20 joints and 19 bone connections. It includes 1,494 video sequences of 10 action categories.

4.2. Implementation Details

For NTU RGB+D and NTU RGB+D 120, each sample is resized to 64 frames, and we adopt the code of [43, 6] for data pre-processing. For NW-UCLA, we follow the data pre-processing procedures in [5, 2, 6]. We use CTR-GCN with single-scale temporal convolution for our ablation study, considering its good balance between performance and efficiency. For ablation study with ST-GCN backbone, please refer to supplementary material. When comparing with other methods, we adopt CTR-GCN with multiscale temporal convolution since it produces the best results. For text encoder, we use the pretrained text transformer model from CLIP or BERT and finetune its parameters during training. The temperature of contrastive loss is set to 0.1. As for the non-deterministic of action descriptions generated by GPT-3, we effectively employed generated results through sampling in the course of training. For example, in the context of our synonyms scenario, we generate numerous synonyms and select them randomly for use in training.

For NTU RGB+D and NTU RGB+D 120, we train the model for a total number of 110 epochs with batch size 200. We use a warm-up strategy for the first 5 epochs. The initial learning rate is set to 0.1 and reduced by a factor of 10 at 90 and 100 epochs, the weight decay is set to 5e-4 following the strategy in [6]. For NW-UCLA, the batchsize, epochs, learning rate,weight decay, reduced step, warm-up epochs are set to 64, 110, 0.2, 4e-4, [90,100], 5, respectively.

4.3. Ablation Study

In this section, we conduct experiments to evaluate the influences of different components. The experiments are conducted on NTU120 RGB+D with joint modality and X-Sub setting. For more ablation studies please refer to **supplementary materials**.

Partition Strategies. We test different body partition strategies for GAP and the results are shown in Table 1a. 'Global' represents using a global description of actions with a single contrastive loss, and it improves over the baseline by 0.6%. Using more parts and multi-part contrastive loss could steadily increase the performance, and it saturates at 85.4% when using 4 parts.

Influences of Text Prompt. The text prompt design has a large impact on the model performance. We show the influences of different text prompts in Table 1b. By directly using label name (with prefix or suffix) as the text prompt in GAP, the model only slightly outperforms (0.2%) the baseline model without text encoder, as this does not bring extra information for training. Utilizing a synonym list for label name or a global description paragraph could largely improve the performance (0.6%) over baseline, as it enriches the semantic meanings of each action class. Using part description prompts leads to strong performance with 0.8% improvement. The best performance is achieved by combining synonym of label name and body part description for prompts, resulting in 85.5% accuracy.

Influences of Text Encoder. In Table 1c, we show the influences of text encoders. We found that both XFMR (text encoder from CLIP [23]) and BERT all achieve good performance, indicating that skeleton encoder could benefit from text encoder with different pre-training sources (image-language or pure language). We use XFMR-32 as our default text encoder considering its good balance between efficiency and accuracy.

Effect of GAP on Different Skeleton Encoders. Our proposed GAP is decoupled from the network architecture and could be employed to improve different skeleton encoders. In Table 1d, we show experimental results of applying GAP to ST-GCN [38], CTR-baseline and CTR-GCN [2]. GAP brings consistent improvements (0.6-1.2%) without extra computation cost at inference, demonstrating the effectiveness and generalization ability of GAP.

Comparison of Description Methods. We compare several different methods of obtaining text prompts for text encoders in Table 1e, including: Manual description;

(a) Partition strategies

Partition Strategy	Acc(%)
None	84.6
Global	85.2
Upper, Lower	85.3
Head, Hand, Hip, Leg	**85.4**
Head, Hand, Arm, Hip, Leg, Foot	**85.4**

(b) Text prompt type

Prompt type	Acc(%)
None	84.6
Label name	84.8(↑ 0.2)
Synonym/Paragraph	85.2(↑ 0.6)
Body parts	85.4(↑ 0.8)
Synonym+Body parts	**85.5**(↑ 0.9)

(c) Text encoders

Text encoder	pretrain	Acc(%)
XFMR-32	img/text	**85.2**
XFMR-16	img/text	85.1
XFMR-14	img/text	**85.2**
BERT	text	**85.2**

(d) Effect of GAP on different skeleton encoders

Backbone	Acc(%) w./o. GAP	Acc(%) w. GAP
ST-GCN [38]	82.6	83.8(↑ 1.2)
CTR-baseline	83.7	84.6(↑ 0.9)
CTR-GCN (single scale)	84.6	**85.5**(↑ 0.9)
CTR-GCN (multi scale) [2]	84.9	**85.5**(↑ 0.6)

(e) Description methods

Methods	Acc(%)
Part CLS Baseline	84.2
Manual description	85.2
HAKE part state	85.3
GPT-3 generated	**85.5**

(f) Comparison with Prompt Learning

Methods	Prompt	TE	Acc
Baseline	Fixed	Tuned	84.8
PL[46]	Learned	Fixed	85.1
	Learned	Tuned	85.2
GAP	Generated	Tuned	**85.5**

Table 1: Ablation study of different components of GAP on NTU120, including partition strategy, text prompt type, text encoder, skeleton encoder, prompt methods. The Acc represents action recognition accuracy, and TE represents text encoder.

HAKE part state; Generating text prompts with GPT-3. For manual descriptions and HAKE results, we use them as global description for GAP. Among these methods, GPT-3 could provide very detailed description of human parts by using an elaborately designed text prompt, and the generated part text description achieves the best performance. We also implement a part pooling classification baseline for reference, which applies a classification head for every pooled part feature. This baseline does not work well as the part feature may not be sufficient to predict the action classes.

Comparing with prompt learning methods. In Table 1f, we compare GAP with PL methods that make prompts learnable parameters. PL outperforms baseline with both Text Encoder (TE)'s parameter fixed or tuned. GAP further outperforms PL by 0.3%, which indicates the effectiveness generated prompts and the multi-part paradigm.

Influences of λ Selection. To study the influences of trade-off parameter λ in Eq. 1, we search the value of λ in $\{1.0, 0.8, 0.5, 0.2\}$ with 5-fold cross-validation. The performance of models are 85.4%, 85.5%, 85.3% and 85.2%, respectively. We found that $\lambda = 0.8$ achieves the best performance; therefore, we utilize it as our default λ value and employ it for all the experiments on different benchmarks.

4.4. Comparison with State-of-the-arts

We compare our method with previous state-of-the-arts in Tables 2, 3 and 4. For fair comparison, we use the 4 ensembles strategy (Joint, Joint-Motion, Bone, Bone-Motion) as it is adopted by most of the previous methods. The results are means of 5 runs, the std is approximately 0.1. As shown in Table 2, on NW-UCLA, GAP outperforms CTR-GCN by 0.7%. It also outperforms the recent work Info-GCN [6] by 0.6%, which uses self-attention layer and information bottleneck. We argue that such improvement is significant considering that the model performance on this dataset is already very high. On NTU RGB+D, GAP outperforms CTR-GCN [2] by 0.5% on cross-subject and 0.2% on cross-view settings, and it outperforms Info-GCN by 0.2% and 0.1% on the two settings, respectively. On the largest dataset NTU RGB+D 120, as shown in Table 4, our method surpasses CTR-GCN by a large margin (1.0%) on cross-subject, and 0.5% on cross-set settings, respectively. Info-GCN also achieves strong performance on this dataset, while GAP still outperforms it by 0.5% and 0.4%, respectively. In summary, GAP consistently outperforms the SOTA on NW-UCLA, NTU RGB+D and NTU RGB+D 120 under different settings, validating its effectiveness and robustness.

Methods	Mode	NW-UCLA Top-1 (%)
Ensemble TS-LSTM [15]	2 ensemble	89.2
2S-AGC-LSTM [27]	2 ensemble	93.3
4S-Shift-GCN [5]	4 ensemble	94.6
DC-GCN+ADG [4]	4 ensemble	95.3
TA-CNN [37]	4 ensemble	96.1
CTR-GCN [2]	4 ensemble	96.5
Info-GCN [6]	4 ensemble	96.6
Ours	Joint/Joint-M	94.0/93.5
	Bone/Bone-M	95.3/91.2
	4 ensemble	**97.2**

Table 2: Action classification performance on the NW-UCLA dataset.

Methods	Mode	NTU-RGB+D X-Sub(%)	NTU-RGB+D X-View(%)
VA-LSTM [41]	2 ensemble	79.4	87.6
HCN [16]	2 ensemble	86.5	91.1
2S-AGCN [25]	2 ensemble	88.5	95.1
SGN [43]	2 ensemble	89.0	94.5
2S-AGC-LSTM [27]	2 ensemble	89.2	95.0
ST-TR (Plizzari et al. 2021)	4 ensemble	89.9	96.1
TA-CNN [37]	4 ensemble	90.4	94.8
4S-Shift-GCN [5]	4 ensemble	90.7	96.5
DC-GCN+ADG [4]	4 ensemble	90.8	96.6
PA-ResGCN-B19 [29]	4 ensemble	90.9	96.0
Dynamic GCN [40]	4 ensemble	91.5	96.0
MS-G3D [19]	2 ensemble	91.5	96.2
DSTA [26]	4 ensemble	91.5	96.4
MST-GCN [3]	4 ensemble	91.5	96.6
EfficientGCN-B4 [30]	4 ensemble	91.7	95.7
CTR-GCN [2]	4 ensemble	92.4	96.8
Info-GCN [6]	4 ensemble	92.7	96.9
Ours	Joint/Joint-M	90.2/88.0	95.6/93.7
	Bone/Bone-M	91.2/87.8	95.5/93.2
	4 ensemble	**92.9**	**97.0**

Table 3: Action classification performance on the NTU RGB+D dataset.

Methods	Mode	NTU-RGB+D 120 X-Sub(%)	NTU-RGB+D 120 X-Set(%)
SGN [43]	2 ensemble	79.2	81.5
ST-TR (Plizzari et al. 2021)	4 ensemble	82.7	84.7
2S-AGCN [25]	2 ensemble	82.9	84.9
TA-CNN [37]	4 ensemble	85.4	86.8
4S-Shift-GCN [5]	4 ensemble	85.9	87.6
DC-GCN+ADG [4]	4 ensemble	86.5	88.1
DSTA [26]	4 ensemble	86.6	89.0
MS-G3D [19]	2 ensemble	86.9	88.4
PA-ResGCN-B19 [29]	4 ensemble	87.3	88.3
Dynamic GCN [40]	4 ensemble	87.3	88.6
MST-GCN [3]	4 ensemble	87.5	88.8
EfficientGCN-B4 [30]	4 ensemble	88.3	89.1
CTR-GCN [2]	4 ensemble	88.9	90.6
Info-GCN [6]	4 ensemble	89.4	90.7
Ours	Joint/Joint-M	85.5/82.3	87.0/83.9
	Bone/Bone-M	87.5/82.4	88.7/84.4
	4 ensemble	**89.9**	**91.1**

Table 4: Action classification performance on the NTU RGB+D 120 dataset.

4.5. Discussions

To facilitate deeper discussions of the proposed GAP method, we utilized a model trained with joint modality on the NTU RGB+D 120 cross-subject mode dataset. In Figure 6, we present the action classes that exhibit over 4% absolute accuracy differences on NTU120 with and without GAP. A good case can be observed for actions such as "writing", "open a box," "eat meal", and "wield knife", which benefit significantly from GAP due to the language model generating detailed descriptions of body part movements for these actions. On the other hand, GAP performs poorly for action classes such as "cutting paper", "taking a selfie", "play magic cube", and "play with phone/tablet". Our analysis revealed that the primary distinguishing factor between these bad performing actions and good performing ones is that the former are object-related, making it challenging to recognize them using skeleton data. Additionally, the category bias present in the dataset may also contribute to the observed performance variations of our proposed method on object-related actions in NTU120 due to the presence of other action categories within the dataset. For instance, upon analyzing "cutting paper", we found that the primary distinguishing factor between it and "rubbing two hands" (which is also present in NTU120) is the presence of an object being held, such as paper and scissors. Conversely, although "opening a box" is also an object-related action, there are no other object-related similar actions within the NTU120 dataset, such as "unfold clothes". **For more discussions and visualization results, please refer to the supplementary materials.**

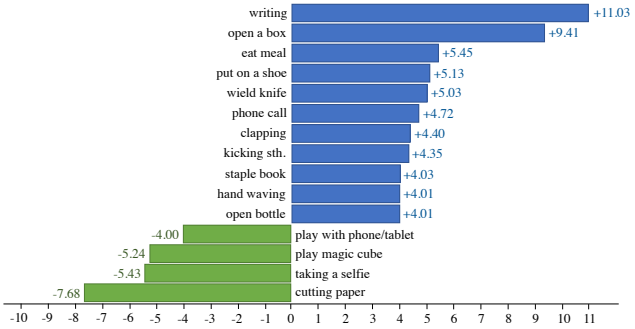


Figure 6: Action classes with accuracy differences higher than 4% between CTR-GCN and our method.

5. Conclusion

We developed a novel generative action-description prompts (GAP) framework for skeleton-based action recognition, which is the first work of its kind, as far as we known, to use action knowledge prior for skeleton action recognition. We employed large-scale language models as knowledge engine to automatically generate detailed descriptions of body parts without laborious manual annotation. GAP utilized knowledge prompting to guide skeleton encoder and enhance the learned representation with knowledge about relations of actions and human body parts. The extensive experiments demonstrated that GAP is a general framework and it can be coupled with various backbone networks to enhance representation learning. GAP achieved new state-of-the-arts on NTU RGB+D, NTU RGB+D 120 and NW-UCLA benchmarks.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. 1

[2] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368, 2021. 2, 4, 6, 7, 8

[3] Zhan Chen, Sicheng Li, Bing Yang, Qinghan Li, and Hong Liu. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1113–1122, 2021. 8

[4] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *European Conference on Computer Vision*, pages 536–553, 2020. 1, 2, 7, 8

[5] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020. 6, 7, 8

[6] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20186–20196, June 2022. 2, 6, 7, 8

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2018. 4

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021. 2

[9] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015. 1, 2

[10] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Part-level graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11045–11052, 2020. 3

[11] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *PMLR*, 2021. 1, 3

[12] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision (ECCV)*. Springer, 2022. 3

[13] Sai Shashank Kalakonda, Shubh Maheshwari, and Ravi Kiran Sarvadevabhatla. Action-gpt: Leveraging large-scale language models for improved and generalized action generation. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2023. 3

[14] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras, 2017. 1

[15] Inwoong Lee, Doyoung Kim, Seoungyoon Kang, and Sanghoon Lee. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1012–1020, 2017. 1, 7

[16] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 786–792. International Joint Conferences on Artificial Intelligence Organization, 7 2018. 8

[17] Yong-Lu Li, Xinpeng Liu, Xiaoqian Wu, Yizhuo Li, Zuoyu Qiu, Liang Xu, Yue Xu, Hao-Shu Fang, and Cewu Lu. Hake: A knowledge engine foundation for human activity understanding. *TPAMI*, 2023. 5

[18] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019. 6

[19] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152, 2020. 4, 8

[20] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *ICLR*, 2023. 3

[21] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Zero-shot temporal action detection via vision-language prompting. In *European Conference on Computer Vision (ECCV)*. Springer, 2022. 3

[22] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Spatial temporal transformer network for skeleton-based action recognition. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15,*

2021, Proceedings, Part III, pages 694–701. Springer, 2021. 1, 2

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 3, 4, 6

[24] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016. 6

[25] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12026–12035, 2019. 1, 2, 8

[26] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Decoupled spatial-temporal attention network for skeleton-based action recognition. *arXiv preprint arXiv:2007.03263*, 2020. 2, 8

[27] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019. 1, 7, 8

[28] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 1, 2

[29] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1625–1633, 2020. 3, 8

[30] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Constructing stronger and faster baselines for skeleton-based action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022. 1, 2, 8

[31] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision*, pages 358–374. Springer, 2022. 3

[32] Kalpit Thakkar and PJ Narayanan. Part-based graph convolutional network for action recognition. *arXiv preprint arXiv:1809.04983*, 2018. 3

[33] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2649–2656, 2014. 6

[34] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *CoRR*, abs/2109.08472, 2021. 3, 5

[35] Qingtian Wang, Jianlin Peng, Shuze Shi, Tingxi Liu, Jiabin He, and Renliang Weng. Iip-transformer: Intra-inter-part transformer for skeleton-based action recognition. *arXiv preprint arXiv:2110.13385*, 2021. 1, 2, 3

[36] Hailun Xia and Xinkai Gao. Multi-scale mixed dense graph convolution network for skeleton-based action recognition. *IEEE Access*, 9:36475–36484, 2021. 1

[37] Kailin Xu, Fanfan Ye, Qiaoyong Zhong, and Di Xie. Topology-aware convolutional neural network for efficient skeleton-based action recognition. *arXiv preprint arXiv:2112.04178*, 2021. 2, 7, 8

[38] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, pages 7444–7452, 2018. 2, 6, 7

[39] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space, 2022. 1, 3

[40] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 55–63, 2020. 1, 8

[41] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126, 2017. 1, 2, 8

[42] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1963–1978, 2019. 2

[43] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1112–1121, 2020. 6, 8

[44] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012. 1

[45] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[46] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 3, 7