LLMs are Good Action Recognizers

Haoxuan Qu[1] Yujun Cai[2] Jun Liu[1†]
[1]Singapore University of Technology and Design [2]Nanyang Technological University
haoxuan_qu@mymail.sutd.edu.sg, yujun001@e.ntu.edu.sg, jun_liu@sutd.edu.sg

Abstract

*Skeleton-based action recognition has attracted lots of research attention. Recently, to build an accurate skeleton-based action recognizer, a variety of works have been proposed. Among them, some works use large model architectures as backbones of their recognizers to boost the skeleton data representation capability, while some other works pre-train their recognizers on external data to enrich the knowledge. In this work, we observe that large language models which have been extensively used in various natural language processing tasks generally hold both large model architectures and rich implicit knowledge. Motivated by this, we propose a novel **LLM-AR** framework, in which we investigate treating the **L**arge **L**anguage **M**odel as an **A**ction **R**ecognizer. In our framework, we propose a linguistic projection process to project each input action signal (i.e., each skeleton sequence) into its "sentence format" (i.e., an "action sentence"). Moreover, we also incorporate our framework with several designs to further facilitate this linguistic projection process. Extensive experiments demonstrate the efficacy of our proposed framework.*

1. Introduction

Human action recognition aims to categorize the actions performed by humans into a pre-defined list of classes. It is relevant to a variety of applications, such as human-computer interaction [37], intelligent surveillance [36], and virtual reality [13]. In the past few years, skeleton-based action recognition [3, 5, 6, 12, 53, 54, 62, 70] has received a lot of research attention with the notice that skeleton is a succinct yet informative representation of human behaviors. Yet, despite the considerable progress, skeleton-based action recognition still remains a challenging task [78], and to build a more accurate skeleton-based action recognizer, various recent works have been proposed from different perspectives. Among them, some recent works [15, 59] proposed to utilize *large model architectures* (such as transformers) as the backbone of their action recognizers to achieve stronger representation capability and capture more subtle differences among different actions. On the other hand, some other works [72, 80] proposed to pre-train their action recognizers on external data in order for their action recognizers to handle this task with richer knowledge.

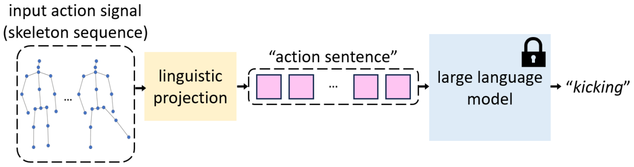

Figure 1. Overview of our proposed LLM-AR framework. In our framework, given an input action signal, we first perform a linguistic projection process to acquire the corresponding "action sentence". We then perform action recognition via the large language model with its pre-trained weights untouched to keep its pre-learned rich knowledge.

Recently, large language models such as GPT [2] and LLaMA [56] have become quite popular and have been extensively applied in handling various human languages. For example, having been pre-trained over various human languages and learned common language-related characteristics, large language models can be very effectively and efficiently adapted to even new human languages unseen during pre-training [49]. Moreover, people have also treated large language models as interpreters in code interpretation [14] and generators in essay generation [22], and found that large language models can handle these tasks effectively. Motivated by this, in this work we are wondering, *if we can also treat the large language model as an action recognizer in skeleton-based human action recognition?* In general, large language models hold some characteristics that are useful for action recognition. Specifically, pre-trained over a tremendously large corpus [2, 56] that generally contains very rich human-centric behavior descriptions, large language models that hold *large model architectures* could naturally contain rich implicit knowledge [46, 50] of human behaviors, and hold strong capability in handling different inputs. Thus, it can be very promising if we treat the large

† Corresponding author

language model as a human action recognizer.

However, despite the success of large language models in handling various human languages, it can be challenging to treat them as action recognizers. This is because, while large language models typically take sentences in human languages as input instructions, the input signal of the action recognizer (i.e., the skeleton sequence) is not in a "sentence format" that is "friendly" to large language models. To handle this issue, given a large language model initialized with its pre-trained weights, one potential way is to adjust the weights of the full model to fit the non-linguistic input action signals. Recently, several works [25, 38] have been proposed to fine-tune large language models and adjust their pre-trained weights to handle non-language tasks. However, as shown in the previous study [64], the adjustment of the large language model's pre-trained weights can hurt its generalizability and lead to the loss of its pre-learned rich knowledge. This is clearly undesirable here, as we hope the large language model to be knowledgeable in order for it to be an accurate action recognizer.

Taking this into account, in this work, we aim to harness the large language model as an action recognizer, and at the same time keep its pre-trained weights untouched to preserve its pre-learned rich knowledge. Specifically, we propose a novel action recognition framework named **L**arge **L**anguage **M**odel as an **A**ction **R**ecognizer (**LLM-AR**). As shown in Fig. 1, to perform action recognition, our LLM-AR framework first involves a linguistic projection process to project the input action signal into its "sentence format", that can be "friendly" and compatible to the large language model pre-trained over human sentences. Then the large language model takes in this "sentence format" of the action signal together with a human sentence as the instruction to predict the name of the action. In the rest of this work, for simplicity, we call the action signal in its "sentence format" an "action sentence".

In our framework, to project the action signals into "action sentences" (i.e., performing linguistic projection), we incorporate several designs. Basically, to perform such a process, we observe that, every sentence in a language generally can be regarded as a sequence consisting of discrete word tokens. Inspired by both this observation and previous works [75, 77], in LLM-AR, we first learn an action-based vector quantized variational autoencoder (VQ-VAE) model to project each action signal into a sequence of discrete tokens.

However, while we can basically represent each "action sentence" by a sequence of discrete tokens, it is not necessary that any discrete token sequence can represent an "action sentence" well. In fact, to enable a large language model to become an accurate action recognizer based on the "action sentences", besides being discrete sequences of tokens, ideally, the "action sentences" also need to fulfill some requirements as discussed below. Firstly, since large language models are generally pre-trained over corpus consisting of sentences in human languages, to facilitate large language models in taking "action sentences" as instructions together with sentences in human languages, these "action sentences" should be "like" sentences in human languages so as to be more friendly to the large language model. On the other hand, "action sentences" should still maintain good representations of their original action signals in order for the large language model to accurately perform action recognition based on them.

To better meet the above requirements in the linguistic projection process, we further incorporate our framework with two designs below. Firstly, to make the "action sentences" more "like" sentences in human languages, we get inspiration from previous linguistic and natural language processing works [27, 47, 48, 55, 82] which show that, languages as the communication tools of human beings often contain human inductive biases. Thus, we propose to incorporate our framework with a learning strategy to regularize the projected "action sentences" to follow human inductive biases as well. Secondly, to make the projected "action sentence" a good representation of its original action signal, motivated by the human skeleton's tree-like nature [63] and hyperbolic space's superior ability in representing tree structures [18], and inspired by previous works [33, 51], we further incorporate our action-based VQ-VAE model with a hyperbolic codebook.

Once we finish the learning of the linguistic projection process, we perform low-rank adaptation (LoRA) [23] on the large language model to let the model understand the projected "action sentences". Note that since the pre-trained weights of the large language model are untouched throughout the LoRA process, we can preserve the pre-learned rich knowledge in the large language model and thus utilize the large language model conveniently in our framework.

The contributions of our work are as follows. 1) We propose a novel action recognition framework named LLM-AR. To the best of our knowledge, this work is the first exploration on treating a large language model with its pre-trained weights untouched as an action recognizer. 2) We introduce several designs in our framework to facilitate the linguistic projection process and produce the "action sentences". 3) LLM-AR achieves state-of-the-art performance on the evaluated benchmarks.

2. Related Work

Human Action Recognition. For tackling human action recognition, most of the existing methods can be roughly categorized into two groups: RGB-based methods [58, 60] and skeleton-based methods [5, 6, 53, 54, 70].

As skeleton sequences can represent human behaviors in a succinct yet informative manner, skeleton-based action recognition has received lots of research attention [3–7, 12, 15, 17, 21, 28, 32, 39, 40, 53, 54, 59, 62, 65–68, 70, 73, 76, 80]. In the early days, different works have been proposed to use different network architectures to handle skeleton-based action recognition. Liu et al. [39] proposed to perform skeleton-based action recognition through a spatial-temporal LSTM architecture. Ke et al. [28] proposed to transform skeleton sequences into grey images and process these images through a CNN network. As time passed, GCN tends to be a popular network architecture [5, 7, 8, 34, 54, 65, 67, 70]. Yan et al. [70] proposed ST-GCN that made the first attempt in performing skeleton-based action recognition using a GCN architecture. After that, various different GCN-based methods have been further proposed, such as AS-GCN [34], 2s-AGCN [54], Shift-GCN [7], and CTR-GCN [5]. More recently, a number of works [15, 59, 76, 80] further proposed to perform skeleton-based action recognition via training a model with a large transformer architecture in an end-to-end manner. Some of such methods include 3Mformer [59] and UPS [15].

Different from these approaches, in this work, from a novel perspective, we investigate how to instruct the large language model to perform skeleton-based action recognition while keeping its pre-trained weights untouched to preserve its pre-learned rich knowledge. To achieve this, we involve our framework with a novel linguistic projection process to project each input action signal into an "action sentence". Specifically, we incorporate several designs in the linguistic projection process to make the projected "action sentences" more "like" sentences of human languages so as to make them more friendly to large language models, and meanwhile keep the "action sentence" a good representation of the input action signal.

Large Language Models. Recently, a variety of large language models have been proposed, such as GPT [2] and LLaMA [56]. Pre-trained over a tremendous number of word tokens, these large language models with large model architectures have been shown to contain very rich knowledge [46, 50]. Consequently, large language models have been extensively explored in various tasks [14, 16, 19, 22, 26, 77, 79], such as language translation [26], essay generation [22], and code interpretation [14]. In this work, we design a novel framework treating the large language model as a human action recognizer leveraging our designed linguistic projection process.

3. Proposed Method

Given a skeleton sequence as the input action signal, the goal of action recognition is to predict its corresponding action class. Recently, holding large model architectures and containing very rich knowledge, pre-trained large language models have been shown to be powerful in handling sentences of human languages, and thus have become useful tools in many natural language processing tasks. Inspired by this, in this work, we aim to *leverage the large language model as an effective action recognizer.* To achieve this, we propose a novel framework **LLM-AR**. Specifically, LLM-AR first performs a linguistic projection process to project the input action signal (i.e., the skeleton sequence) into an "action sentence". After that, LLM-AR passes the "action sentence" into the large language model to derive the corresponding human action. Below, we first describe how LLM-AR performs the linguistic projection process, and then introduce the overall training and testing scheme of LLM-AR.

3.1. Linguistic Projection

In our framework, to enable the large language model to perform action recognition, we first perform a linguistic projection process to project each input action signal into an "action sentence". To learn to perform such a projection, with the observation that each sentence in human languages is essentially a sequence of discrete word tokens, motivate by previous works [75, 77], we first learn an action-based VQ-VAE model to project each input action signal into a sequence of discrete tokens. We then further (1) involve the above learning process with a human-inductive-biases-guided learning strategy to make the projected "action sentences" more "like" sentences in human languages, and (2) incorporate the action-based VQ-VAE model with a hyperbolic codebook to keep "action sentences" good representations of the action signals.

Action-based VQ-VAE Model. VQ-VAE models [51, 57, 74] have been popularly used in converting an image into a discrete token sequence. Inspired by this, in our framework, to convert the input action signal into a sequence of discrete tokens, we first learn an action-based VQ-VAE model.

Specifically, similar to the architecture of previous VQ-VAE models [57, 75, 77], our action-based VQ-VAE model involves an encoder E, a decoder D, and a codebook C consisting of U learnable tokens (i.e., $C = \{c_u\}_{u=1}^{U}$ where $c_u \in \mathbb{R}^{d_u}$). We here keep d_u to be the same as the dimension of the word token in the used large language model, and keep U an even number. Among the aforementioned three components of the model, given an input action signal $s_{1:V}$, where V represents the action length, the encoder E first encodes the action signal through 1D convolutions over the time dimension into a sequence of latent features $f_{1:W}$, where $f_w \in \mathbb{R}^{d_u}$ and W represents the length of the latent feature sequence. After this, to discretize the latent features $f_{1:W}$, a quantization operation is performed to replace each feature f_w with its nearest token in the codebook as:

$$f_w^d = \underset{c_u \in C}{\operatorname{argmin}} \big(dist(f_w, c_u) \big) \tag{1}$$

where f_w^d is the discrete version of f_w, and $dist(\cdot, \cdot)$ represents a distance function. Finally, the decoder aims to recover the original action signal $s_{1:V}$ from the sequence of discrete latent features (tokens) $f^d_{1:W}$. Through the above process of encoding, quantization, and decoding, we can project an action signal into a sequence of informative discrete tokens $f^d_{1:W}$. The more detailed architecture of this action-based VQ-VAE model is provided in supplementary.

Human-inductive-biases-guided Learning Strategy. Above we learn to project each input action signal into a sequence of discrete tokens. To enable the learned token sequences to be more friendly to the large language model, here we aim to make these sequences more "like" sentences in human languages. To achieve this, we get inspiration from massive existing studies [27, 47, 48, 55, 82] which show that, human languages as tools created for human communication naturally contain human inductive biases. Besides, it is also further shown by the previous study [47] that, inputs following human inductive biases are more friendly to large language models. Taking these into consideration, we aim to optimize the set of learned token sequences to also follow human inductive biases, so that these sequences can be more "like" human sentences and thus become more friendly to be used by large language models.

Below, we first introduce the human inductive biases that are generally recognized as being present in human languages. Such biases include: (a) a human language naturally follows the Zipf's law [47, 48, 82], and (b) a human language is generally context-sensitive [27, 47, 55]. After introducing these human inductive biases, we then describe our proposed human-inductive-biases-guided learning strategy.

As for the Zipf's law in (a), during daily communications of human beings, there naturally exist some words that are used more commonly and some words more rarely. Zipf's law intuitively represents this imbalanced usage frequency of word tokens in human languages. Specifically, Zipf's law states that, in a human language, the i-th most commonly used word has its usage frequency roughly proportional to:

$$\frac{1}{(i+\beta)^{\alpha}} \tag{2}$$

where $\alpha \approx 1$ and $\beta \approx 2.7$ [82]. The reason why Zipf's law consistently appears across different human languages is also theoretically analyzed by various works [9, 43] from different perspectives, such as from the evolution of human communications.

With respect to the context-sensitivity of human languages in (b), intuitively, the context-sensitivity refers to the inductive bias that when people formulate their sentences, they often do not use each word token independently. Instead, their usage of different tokens in formulating a sentence is often correlated. Note that, while human languages are generally believed by linguists to be context-sensitive, how to explicitly represent context-sensitivity remains a difficult problem. Here, inspired by [47] in its way of representing context-sensitivity, we represent this human inductive bias as follows. Specifically, given the codebook C consisting of U tokens where U is an even number, during initializing C, we first randomly split the codebook into two halves (i.e., $\{1, ..., \frac{U}{2}\}$ and $\{1+\frac{U}{2}, ..., U\}$). Next, we regard each token c_u in the first half (i.e., $u \in \{1, ..., \frac{U}{2}\}$) and the token $c_{u+\frac{U}{2}}$ in the second half to be a pair of correlated tokens. As shown in [47], such a pairing mechanism is a simple yet very effective way of representing the context-sensitivity bias in human languages.

To optimize the set of learned token sequences (i.e., "action sentences") $\{f^d_{1:W}\}$ to also follow the above biases like human languages, we then aim to (a) regularize the set of "action sentences" to follow Zipf's law, as well as (b) regularize each "action sentence" to be formulated using more correlated word tokens so that the formulated "action sentences" can better follow the context-sensitivity bias. To achieve this, we design a human-inductive-biases-guided learning strategy that consists of the following steps. (1) Given a batch of action signals $\{s^b_{1:V}\}^B_{b=1}$ with batch size B, we first encode these signals through the encoder E of the action-based VQ-VAE model to get their corresponding latent features $\{f^b_{1:W}\}^B_{b=1}$. (2) After that, during discretizing the latent features using tokens $\{c_u\}^U_{u=1}$ in the codebook C, we measure the token usage of each sequence of latent features $f^b_{1:W}$ in a differentiable way via the Gumbel Softmax trick [24] as:

$$t^b = \sum_{w=1}^{W} d^b_w, \textbf{where } d^b_w = \mathrm{Gumble\_Softmax}(-dist(f^b_w, c_u)) \tag{3}$$

where $\mathrm{Gumble\_Softmax}(\cdot)$ is the Gumble Softmax trick, and d^b_w is the one-hot vector with length U as the output of the Gumble Softmax trick. Besides, t^b is a vector with length U, and the value of the u-th element of t^b represents the number of times token c_u is used in discretizing $f^b_{1:W}$ and formulating its corresponding "action sentence". (3) Next, denoting D_{Zipf} the Zipf distribution with $\alpha = 1$ and $\beta = 2.7$, we regularize the set of "action sentences" to follow Zipf's law through L_{Zipf} as:

$$L_{Zipf} = JS(D_{freq} \| D_{Zipf}), \textbf{where} D_{freq} = \frac{sort(\sum_{b=1}^{B} t^b)}{B \times W} \tag{4}$$

where $sort(\cdot)$ is the sorting operation used to order tokens based on their usage frequency, D_{freq} is the distribution representing the token usage frequency, and $JS(\cdot\|\cdot)$ represents the JS divergence between two distributions. (4) At the same time, we encourage the "action sentences" to follow the context-sensitivity bias and use more correlated tokens via $L_{context}$ as:

$$L_{context} = 1 - \frac{\sum_{b=1}^{B} Corr(t^b)}{B \times W} \tag{5}$$

where $Corr(t^b)$ leverages min pooling over every pair of

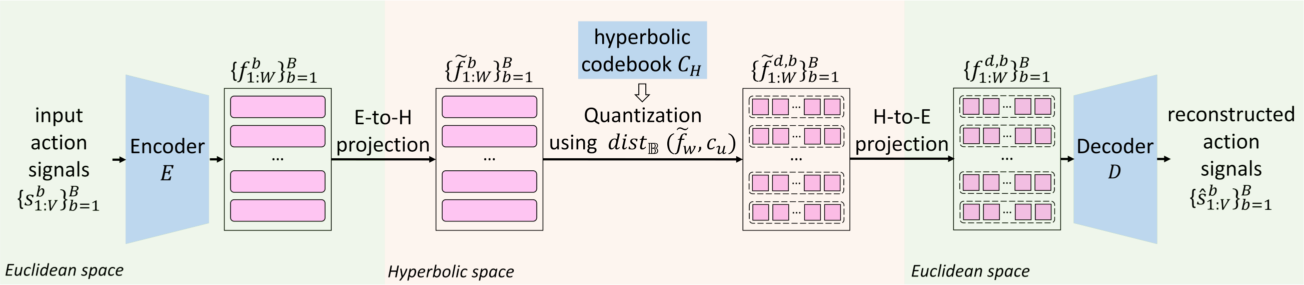
input action signals $\{s^b_{1:V}\}_{b=1}^B$
Encoder E
$\{f^b_{1:W}\}_{b=1}^B$
E-to-H projection
$\{\widetilde{f}^b_{1:W}\}_{b=1}^B$
hyperbolic codebook C_H
Quantization using $dist_{\mathbb{B}}\ (\widetilde{f}_w, c_u)$
$\{\widetilde{f}^{d,b}_{1:W}\}_{b=1}^B$
H-to-E projection
$\{f^{d,b}_{1:W}\}_{b=1}^B$
Decoder D
reconstructed action signals $\{\hat{s}^b_{1:V}\}_{b=1}^B$
Euclidean space
Hyperbolic space
Euclidean space

Figure 2. Overview of the action-based VQ-VAE model with the hyperbolic codebook C_H incorporated. Given a batch of input action signals $\{s^b_{1:V}\}_{b=1}^B$, to optimize the action-based VQ-VAE model, $\{s^b_{1:V}\}_{b=1}^B$ are first fed to the encoder E to get the corresponding latent features $\{f^b_{1:W}\}_{b=1}^B$. Next, to leverage the hyperbolic codebook C_H that can serve as a good representation of the tree-like human skeletons to perform quantization, $\{f^b_{1:W}\}_{b=1}^B$ are projected into the hyperbolic space via the process of E-to-H projection. After that, the quantization is performed in the hyperbolic space using $dist_{\mathbb{B}}(\widetilde{f}_w, c_u)$ defined in Eq. 9 as the distance function. Finally, after quantization, the discrete version of the latent features are passed back into the Euclidean space via the process of H-to-E projection to reconstruct the input action signals through the decoder D.

elements in t^b (e.g., the pair of u-th element and $(u+\frac{U}{2})$-th element) to calculate the number of correlated tokens used in discretizing $f^b_{1:W}$ (more details about this method to measure $Corr(t^b)$ are provided in supplementary). Note that via minimizing $L_{context}$, we encourage more correlated tokens to be used in formulating "action sentences". (5) Finally, we formulate the loss function L_{human} that we use to perform our human-inductive-biases-guided strategy as:

$$L_{human} = L_{Zipf} + L_{context} \tag{6}$$

Via incorporating the above learning strategy into the learning process of the action-based VQ-VAE model, we can regularize the "action sentences" to better follow the human inductive biases that are present in human languages and thus make them more "like" sentences in human languages.

Above we inject human inductive biases into "action sentences". Here, we notice that, besides inductive biases that can be explicitly described, human languages can also present other implicit characteristics. Thus, to enable "action sentences" to be more friendly to large language models which are generally pre-trained over various human languages, we aim to further incorporate implicit characteristics into "action sentences" as well. Specifically, since large language models such as LLaMA typically use word tokens $\{c_{LLM}\}$ stored in its first layer to formulate sentences in human languages, we here further align (1) the tokens $\{c_u\}_{u=1}^U$ in codebook C that are used to formulate the "action sentences" with (2) the word tokens $\{c_{LLM}\}$ used in the large language model. To achieve such an alignment, we leverage Maximum Mean Discrepancy (MMD) [20] as an effective feature alignment technique to measure the discrepancy between $\{c_u\}_{u=1}^U$ and $\{c_{LLM}\}$. Specifically, the less $\text{MMD}(\{c_u\}_{u=1}^U, \{c_{LLM}\})$ is, $\{c_u\}_{u=1}^U$ and $\{c_{LLM}\}$ can be regarded as more aligned. We then incorporate this alignment procedure into our proposed strategy via rewriting L_{human} in Eq. 6 as:

$$L_{human} = L_{Zipf} + L_{context} + \text{MMD}(\{c_u\}_{u=1}^U, \{c_{LLM}\}) \tag{7}$$

By incorporating this alignment procedure into the previously mentioned learning strategy, we can then lead the formulated "action sentences" $\{f^d_{1:W}\}$ to be more friendly to be used by large language models.

Hyperbolic Codebook. Besides making the "action sentences" more friendly to the large language model, we also aim to keep them good representations of the original input action signals (i.e., the skeleton sequences). Motivated by the tree-like nature of human skeletons [63], we aim to make the word tokens in the "action sentences" to well represent such a structure. To achieve this, inspired by the superior capability of the hyperbolic space in embedding tree structures [18] and motivated by previous VQ-VAE works [33, 51], we here further incorporate our action-based VQ-VAE model with a hyperbolic codebook utilizing the Poincaré ball model [18, 45]. The Poincaré ball model is an isometric model that can represent the hyperbolic space. Formally, the n-dimensional Poincaré ball model is defined as $(\mathbb{B}^n_c, g^c_{\mathbf{x}})$, where $\mathbb{B}^n_c = \{\mathbf{x} \in \mathbb{R}^n : c\|\mathbf{x}\| < 1\}$, $g^c_{\mathbf{x}} = (\gamma^c_{\mathbf{x}})^2 I_n$ is the Riemannian metric tensor, $\gamma^c_{\mathbf{x}} = \frac{2}{1-c\|\mathbf{x}\|^2}$ is the conformal factor, I_n is the Euclidean metric tensor, and c is the curvature.

Denoting the hyperbolic codebook $C_H = \{c_u\}_{u=1}^U$, where $c_u \in \mathbb{B}^{d_u}_c$, we make the following three changes to incorporate C_H into our action-based VQ-VAE model. We also illustrate where these changes take place in our action-based VQ-VAE model in Fig. 2. **(1) Euclidean-to-Hyperbolic (E-to-H) Projection.** Given a latent feature f_w which is originally in the Euclidean space, to convert it into a discrete token using a hyperbolic codebook, we first need to project f_w into the hyperbolic space. Specifically, following [18], we perform such a projection through the exponential map $\exp_{\mathbf{0}}^{c}(\cdot)$ as:

$$\widetilde{f}_w = \exp_{\mathbf{0}}^{c}(f_w) = tanh(\sqrt{c}\|f_w\|)\frac{f_w}{\sqrt{c}\|f_w\|} \tag{8}$$

where $\widetilde{f}_w$ is the projected feature of f_w in the hyperbolic space. **(2) Hyperbolic Distance Calculation.** Given the projected feature $\widetilde{f}_w$, to perform quantization of $\widetilde{f}_w$ based on the tokens in C_H, we need a distance function defined in the hyperbolic space. Here, for simplicity, we use the popularly used *geodesic/induced distance* [18, 29]. Denote this distance as $dist_{\mathbb{B}}(\cdot,\cdot)$. $dist_{\mathbb{B}}(\widetilde{f}_w, c_u)$ between $\widetilde{f}_w$ and c_u can be defined as:

$$dist_{\mathbb{B}}(\widetilde{f}_w, c_u) = arccosh\big(1 + 2\frac{\|\widetilde{f}_w - c_u\|^2}{(1-\|\widetilde{f}_w\|^2)(1-\|c_u\|^2)}\big) \tag{9}$$

(3) Hyperbolic-to-Euclidean (H-to-E) Projection. After quantization, we need to project $\widetilde{f}_w^d$ (the discrete version of $\widetilde{f}_w$) back into the Euclidean space in order for it to be passed to the decoder D. To achieve this, we follow [18] to leverage the logarithmic map $\log_{\mathbf{0}}^{c}(\cdot)$ as:

$$f_w^d = \log_{\mathbf{0}}^{c}(\widetilde{f}_w^d) = arctanh(\sqrt{c}\|\widetilde{f}_w^d\|)\frac{\widetilde{f}_w^d}{\sqrt{c}\|\widetilde{f}_w^d\|} \tag{10}$$

where f_w^d represents the discrete version of the feature projected back in the Euclidean space. With the above changes made, we can seamlessly involve the hyperbolic codebook C_H into our action-based VQ-VAE model, and make the projected "action sentence" a better representation of the input action signal.

3.2. Overall Training and Testing

Above we describe how we perform the linguistic projection process in our framework to project each input action signal into an "action sentence". Below, we introduce the overall training and testing scheme of our framework.

Training. The training process of our framework consists of the following two stages: (1) optimizing the action-based VQ-VAE model incorporated with the hyperbolic codebook C_H to acquire the ability to project each action signal $s_{1:V}$ into its corresponding "action sentence" $\widetilde{f}_{1:W}^d$; and (2) perform low-rank adaptation (LoRA) [23] on the large language model for it to better understand the projected "action sentences".

At the first stage, to optimize the action-based VQ-VAE model, given a batch of action signals, we first follow previous VQ-VAE works [57, 75, 77] to use the following loss functions, including the reconstruction loss, the embedding loss, and the commitment loss. Among them, the reconstruction loss L_{re} is designed to regularize the difference between the input action signal and the action signal reconstructed from the decoder D, the embedding loss L_{embed} is designed to learn the tokens in the codebook, and the commitment loss L_{commit} is designed to encourage each of the encoded latent features to stay close to its discrete version. Formally speaking, denoting $\{s_{1:V}^b\}_{b=1}^B$ a batch of input action signals, $\{\hat{s}_{1:V}^b\}_{b=1}^B$ the corresponding batch of reconstructed action signals, $\{f_{1:W}^b\}_{b=1}^B$ the latent features encoded from $\{s_{1:V}^b\}_{b=1}^B$, and $\{f_{1:W}^{d,b}\}_{b=1}^B$ the discrete versions of these latent features, the above three loss functions can be written as:

$$\begin{aligned} L_{re} &= L_1^{smooth}(\{s_{1:V}^b\}_{b=1}^B, \{\hat{s}_{1:V}^b\}_{b=1}^B) \\ L_{embed} &= \|sg(\{f_{1:W}^b\}_{b=1}^B) - \{f_{1:W}^{d,b}\}_{b=1}^B\|^2 \\ L_{commit} &= \|\{f_{1:W}^b\}_{b=1}^B - sg(\{f_{1:W}^{d,b}\}_{b=1}^B)\|^2 \end{aligned} \tag{11}$$

where $L_1^{smooth}(\cdot)$ represents the L1 smooth loss and $sg(\cdot)$ is the stop-gradient operation that is used to prevent the gradient from flowing through its operand. In addition, to make the learned "action sentences" more like sentences in human languages, we also incorporate the above three loss functions with L_{human} defined in Eq. 7. Overall, we can write the total loss L_{total} for the first training stage as:

$$L_{total} = L_{re} + L_{embed} + \omega_1 L_{commit} + \omega_2 L_{human} \tag{12}$$

where ω_1 and ω_2 are weighting hyperparameters.

We then perform LoRA [23] on the large language model in the second stage in order for it to better understand the "action sentences" while keeping the pre-trained weights of the model untouched. Specifically, for each training sample consisting of an input action signal and its corresponding ground-truth action, we perform the following steps. (1) We first project the action signal into an "action sentence" of discrete tokens through the action-based VQ-VAE model learned in the first stage. (2) We then instruct the large language model to act as an action recognizer via a simple instruction as: "`Given a sequence of action tokens [tokens], please predict the corresponding action.`", where [tokens] represent the word tokens of the "action sentence" derived in step (1). (3) Finally, during LoRA process, we pass the above instruction into the large language model and encourage the similarity between the tokens t_p predicted by the large language model and the tokens t_g representing the ground-truth action as:

$$L_{LoRA} = L_{ce}(t_p, t_g) \tag{13}$$

where $L_{ce}(\cdot,\cdot)$ represents the cross-entropy loss.

Testing. During testing, for each testing action signal, we first use the learned action-based VQ-VAE model to derive its corresponding "action sentence". After that, we use the same instruction as in step (2) above to instruct the large language model to predict the corresponding action.

4. Experiments

To evaluate the efficacy of our framework, we conduct experiments on 4 datasets including NTU RGB+D, NTU RGB+D 120, Toyota Smarthome, and UAV-Human.

4.1. Datasets

NTU RGB+D [52] is a large-scale dataset popularly used in human action recognition. It contains around 56k skeleton sequences from 60 activity classes. On this dataset, following [52], we evaluate our method under the Cross-Subject (X-Sub) and Cross-View (X-View) evaluation protocols.

NTU RGB+D 120 [41] is an extension of the NTU RGB+D dataset. It consists of more than 114k skeleton sequences across 120 activity classes. Following [41], we evaluate our method on this dataset under the Cross-Subject (X-Sub) and Cross-Setup (X-Set) evaluation protocols.

Toyota Smarthome [10] contains 16,115 video samples over 31 activity classes. On this dataset, we use the skeleton sequences pre-processed by [71] and we follow it to evaluate our method under the Cross-Subject (X-Sub) and two Cross-View (X-View1 & X-View2) evaluation protocols.

UAV-Human [35] is a dataset that is captured by unmanned aerial vehicles (UAV). It contains more than 20k skeleton sequences over 155 activity classes, and it is collected from 119 distinct subjects. Following [35], we use 89 subjects for training and 30 subjects for testing.

4.2. Implementation Details

We conduct our main experiments on Nvidia V100 GPU and we use LLaMA-13B [56] as the large language model. During the training process of the action-based VQ-VAE model, we optimize the model for 300,000 iterations using the AdamW [42] optimizer with an initial learning rate of 2e-4. We set the batch size B to 256, the number of tokens U to 512, and the curvature c of the hyperbolic codebook to 1. Additionally, we set the dimension of each word token d_u to the same size (5120) as the word token in LLaMA-13B. Moreover, we set w_1 to 0.02 following previous VQ-VAE works [44, 75] and set w_2 to 0.2. Besides, we set the length (W) of each sequence of latent features to be a quarter of the length (V) of its corresponding input action signal. During the LoRA process, we develop our code based on the Github repo [1], set the number of iterations to 75,000, and use the AdamW optimizer with an initial learning rate of 3e-3. Moreover, we set the batch size to 256, partitioned into micro-batches of 4. Besides, we set the two hyperparameters of the LoRA process, i.e., r_{LoRA} and α_{LoRA}, to 64 and 16 respectively.

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

On NTU RGB+D and NTU RGB+D 120 datasets, following the experimental setting of recent works [12, 80], we only use the joint modality of human skeletons during our experiments. We report the results in Tab. 1. As shown, compared to existing skeleton-based action recognition methods, our method consistently achieves the best performance across all the evaluation protocols. This demonstrates the effectiveness of our method. Besides, we also report results on the Toyota Smarthome dataset in Tab. 2, and on the UAV-Human dataset in Tab. 3. As shown, our method consistently achieves the best performance on these datasets. This further shows the efficacy of our method.

Table 1. Performance comparison on the NTU RGB+D and NTU RGB+D 120 datasets.

Method	NTU RGB+D		NTU RGB+D 120	
	X-Sub	X-View	X-Sub	X-Set
ST-GCN [70]	85.7	92.4	82.1	84.5
Shift-GCN [7]	87.8	95.1	80.9	83.2
InfoGCN [8]	89.8	95.2	85.1	86.3
PoseC3D [11]	93.7	96.5	85.9	89.7
FR-Head [78]	90.3	95.3	85.5	87.3
Koopman [62]	90.2	95.2	85.7	87.4
GAP [65]	90.2	95.6	85.5	87.0
HD-GCN [31]	90.6	95.7	85.7	87.3
STC-Net [30]	91.0	96.2	86.2	88.0
DSTformer [80]	93.0	97.2	-	-
SkeleTR [12]	94.8	97.7	87.8	88.3
Ours	**95.0**	**98.4**	**88.7**	**91.5**

Table 2. Performance comparison on the Toyota Smarthome dataset.

Method	X-Sub	X-View1	X-View2
2S-AGCN [54]	58.8	32.2	57.9
SSTA-PRS [71]	62.1	22.8	54.0
UNIK [72]	62.1	33.4	63.6
ML-STGNet [81]	64.6	29.9	63.5
Ours	**67.0**	**36.1**	**66.6**

Table 3. Performance comparison on the UAV-Human dataset.

Method	X-Sub
ST-GCN [69]	30.3
2S-AGCN [54]	34.8
Shift-GCN [7]	38.0
ACFL [61]	44.2
Ours	**46.3**

4.4. Ablation Studies

We conduct extensive ablation experiments on the X-Set protocol of the NTU RGB+D 120 dataset. **More ablation studies such as experiments w.r.t. hyperparameters are in supplementary.**

Table 4. Evaluation on the human-inductive-biases-guided learning strategy.

Method	Accuracy
w/o biases	87.6
w/o L_{Zipf}	89.9
w/o $L_{context}$	89.8
w/o MMD alignment	90.3
LLM-AR	91.5

Impact of the human-inductive-biases-guided learning strategy. In our framework, to lead the formulated "action sentences" to be more friendly to large language models, we design a human-inductive-biases-guided learning strategy consisting of three components (as shown in Eq. 7). To evaluate the efficacy of this strategy, we test four variants. In the first variant (**w/o biases**), we remove the whole strategy (i.e., all its three components L_{Zipf}, $L_{context}$, and MMD alignment) from the learning process. In the second variant (**w/o L_{Zipf}**), we still utilize the strategy but remove its L_{Zipf} component. Moreover, in the third variant (**w/o $L_{context}$**), we remove the $L_{context}$ component from the strategy, whereas in the fourth variant (**w/o MMD alignment**), we remove the MMD alignment component. As shown in Tab. 4, compared to our framework, the performance of the first variant drops significantly. This shows the importance of formulating "action sentences" like sentences in human languages. Moreover, our framework also outperforms all the other three variants. This further shows the effectiveness of all the three components of our proposed learning strategy.

Impact of discretizing latent features into "action sentences". In our framework, we discretize the encoded latent features to formulate "action sentences" consisting of discrete word tokens (**with discretization**). To valid this design, we test a variant. In this variant (**w/o discretization**), instead of discretizing the encoded latent features into "action sentences", we directly pass these continuous features to the intermediate layers of the large language model. As shown in Tab. 5, our framework with discretization outperforms this variant. This shows the advantage of discretizing latent features into "action sentences", which are more "like" human sentences consisting of discrete word tokens, and thus are more friendly to the large language model pre-trained over human sentences.

Table 5. Evaluation on discretizing latent features into "action sentences".

Method	Accuracy
w/o discretization	83.4
with discretization	91.5

Impact of the hyperbolic codebook C_H. In our framework, we incorporate our action-based VQ-VAE model with a hyperbolic codebook C_H (**with** C_H). To validate the efficacy of C_H, we test a variant (**w/o** C_H) in which the codebook is set up in the Euclidean instead of hyperbolic space. As shown in Tab. 6, our framework involving C_H performs better than this variant. This shows the efficacy of C_H in the hyperbolic space, which can facilitate the "action sentences" in representing the tree-like-structured input action signals better.

Table 6. Evaluation on the hyperbolic codebook C_H.

Method	Accuracy
w/o C_H	89.7
with C_H	91.5

Impact of the LoRA process. In our framework, to make the large language model understand the "action sentences" while keeping its pre-trained weights untouched to preserve its rich pre-learned knowledge, we tune the large language model through a LoRA process. To validate this scheme, here we also test a variant (**all tuning**) on A100 GPU. In this variant, during tuning, after initializing the large language model with its pre-trained weights, all the parameters of the model will undergo gradient updates. As shown in Tab. 7, our framework achieves much better performance than this variant. This shows the superiority of our framework in choosing to perform tuning via LoRA, which enables the large language model's pre-trained weights to be untouched and maintains its pre-learned rich knowledge.

Table 7. Evaluation on the LoRA process.

Method	Accuracy
All tuning	79.6
LLM-AR	91.5

Evaluation on unseen activity classes. In the main experiments, following [31, 52, 70], we evaluate our framework on activity classes that have been seen during training. Here, inspired by that large language models naturally could contain rich knowledge beyond the training activity classes used in our experiments, we are curious, *assuming we have a list of testing activity classes unseen during training, can we also use our framework to perform action recognition on these classes?* To answer this question, we first build a new evaluation protocol for unseen activity classes based on the NTU RGB+D 120 dataset. Under this new protocol, during each time of evaluation, we randomly select 3 classes to form the unseen class list (i.e., the list of testing classes), and use the remaining classes as the classes seen during training (i.e., the training classes). Besides, we instruct the large language model as: "Given a sequence of action tokens [tokens], please predict the corresponding action from [list].", where [tokens] represent the word tokens of the "action sentence" and [list] represents the unseen class list. We then perform the above evaluation for five times and report the average performance. As shown in Tab. 8, even testing on activity classes unseen during training, our framework can still achieve a relatively good performance, while the afore-defined **all tuning** variant performs much worse. This can be analyzed as, large language models could contain rich pre-learned knowledge w.r.t. the list of unseen classes. Thus, our framework that maintains such rich knowledge can still perform promising recognition on these unseen classes, while the **all tuning** variant that can lose amount of pre-learned knowledge of the large language model would yield a much worse performance. This further shows the advantage of our framework in maintaining the pre-learned rich knowledge of the large language model.

Table 8. Evaluation on unseen activity classes.

Method	Accuracy
All tuning	37.7
LLM-AR	62.4

5. Conclusion

In this paper, we have proposed a novel action recognition framework LLM-AR. In LLM-AR, we treat the large language model as an action recognizer, and instruct the large language model to perform action recognition using its contained rich knowledge. Specifically, to lead the input action signals (i.e., the skeleton sequences) to be more friendly to the large language model, we first propose a linguistic projection process to project each action signal into an "action sentence". Moreover, we also introduce several designs to further facilitate this process. Our framework consistently achieves SOTA performance across different benchmarks.

Acknowledgement. This project is supported by the Ministry of Education, Singapore, under the AcRF Tier 2 Projects (MOE-T2EP20222-0009 and MOE-T2EP20123-0014).

References

[1] Lit-llama. https://github.com/Lightning-AI/lit-llama. 7

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 3

[3] Dongqi Cai, Yangyuxuan Kang, Anbang Yao, and Yurong Chen. Ske2grid: Skeleton-to-grid representation learning for action recognition. 2023. 1, 3

[4] Jinghong Chen, Chong Zhao, Qicong Wang, and Hongying Meng. Hmanet: Hyperbolic manifold aware network for skeleton-based action recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 2022.

[5] 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. 1, 2, 3

[6] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 536–553. Springer, 2020. 1, 2

[7] 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. 3, 7

[8] 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*, pages 20186–20196, 2022. 3, 7

[9] Bernat Corominas-Murtra, Jordi Fortuny, and Ricard V Solé. Emergence of zipf's law in the evolution of communication. *Physical Review E*, 83(3):036115, 2011. 4

[10] Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 833–842, 2019. 7

[11] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978, 2022. 7

[12] Haodong Duan, Mingze Xu, Bing Shuai, Davide Modolo, Zhuowen Tu, Joseph Tighe, and Alessandro Bergamo. Skeletr: Towards skeleton-based action recognition in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13634–13644, 2023. 1, 3, 7

[13] Abassin Sourou Fangbemi, Bin Liu, Neng Hai Yu, and Yanxiang Zhang. Efficient human action recognition interface for augmented and virtual reality applications based on binary descriptor. In *Augmented Reality, Virtual Reality, and Computer Graphics: 5th International Conference, AVR 2018, Otranto, Italy, June 24–27, 2018, Proceedings, Part I 5*, pages 252–260. Springer, 2018. 1

[14] Faisal Firdous, Saimul Bashir, Syed Zoofa Rufai, and Sanjeev Kumar. Openai chatgpt as a logical interpreter of code. In *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, pages 1192–1197. IEEE, 2023. 1, 3

[15] Lin Geng Foo, Tianjiao Li, Hossein Rahmani, Qiuhong Ke, and Jun Liu. Unified pose sequence modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13019–13030, 2023. 1, 3

[16] Lin Geng Foo, Hossein Rahmani, and Jun Liu. Ai-generated content (aigc) for various data modalities: A survey, 2023. 3

[17] Luca Franco, Paolo Mandica, Bharti Munjal, and Fabio Galasso. Hyperbolic self-paced learning for self-supervised skeleton-based action representations. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[18] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018. 2, 5, 6

[19] Jia Gong, Lin Geng Foo, Yixuan He, Hossein Rahmani, and Jun Liu. Llms are good sign language translators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[20] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006. 5

[21] Ryo Hachiuma, Fumiaki Sato, and Taiki Sekii. Unified keypoint-based action recognition framework via structured keypoint pooling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22962–22971, 2023. 3

[22] Steffen Herbold, Annette Hautli-Janisz, Ute Heuer, Zlata Kikteva, and Alexander Trautsch. Ai, write an essay for me: A large-scale comparison of human-written versus chatgpt-generated essays. *arXiv preprint arXiv:2304.14276*, 2023. 1, 3

[23] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 6

[24] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[25] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *arXiv preprint arXiv:2306.14795*, 2023. 2

[26] Wenxiang Jiao, Wenxuan Wang, JT Huang, Xing Wang, and ZP Tu. Is chatgpt a good translator? yes with gpt-4 as the engine. *arXiv preprint arXiv:2301.08745*, 2023. 3

[27] Aravind K Joshi, K Vijay Shanker, and David Weir. The convergence of mildly context-sensitive grammar formalisms. *Technical Reports (CIS)*, page 539, 1990. 2, 4

[28] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3288–3297, 2017. 3

[29] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 2020. 6

[30] Jungho Lee, Minhyeok Lee, Suhwan Cho, Sungmin Woo, Sungjun Jang, and Sangyoun Lee. Leveraging spatio-temporal dependency for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10255–10264, 2023. 7

[31] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoun Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10444–10453, 2023. 7, 8

[32] Zikang Leng, Hyeokhyen Kwon, and Thomas Plötz. On the benefit of generative foundation models for human activity recognition. *arXiv preprint arXiv:2310.12085*, 2023. 3

[33] Lei Li, Tingting Liu, Chengyu Wang, Minghui Qiu, Cen Chen, Ming Gao, and Aoying Zhou. Resizing codebook of vector quantization without retraining. *Multimedia Systems*, pages 1–14, 2023. 2, 5

[34] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3595–3603, 2019. 3

[35] Tianjiao Li, Jun Liu, Wei Zhang, Yun Ni, Wenqian Wang, and Zhiheng Li. Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16266–16275, 2021. 7

[36] Weiyao Lin, Ming-Ting Sun, Radha Poovendran, and Zhengyou Zhang. Activity recognition using a combination of category components and local models for video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1128–1139, 2008. 1

[37] Hong Liu, Qinqin He, and Mengyuan Liu. Human action recognition using adaptive hierarchical depth motion maps and gabor filter. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1432–1436. IEEE, 2017. 1

[38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 2

[39] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 816–833. Springer, 2016. 3

[40] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1647–1656, 2017. 3

[41] 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. 7

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[43] Benoit Mandelbrot. Structure formelle des textes et communication: Deux études par. *Word*, 10(1):1–27, 1954. 4

[44] Evonne Ng, Sanjay Subramanian, Dan Klein, Angjoo Kanazawa, Trevor Darrell, and Shiry Ginosar. Can language models learn to listen? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10083–10093, 2023. 7

[45] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017. 5

[46] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022. 1, 3

[47] Isabel Papadimitriou and Dan Jurafsky. Pretrain on just structure: Understanding linguistic inductive biases using transfer learning. *arXiv preprint arXiv:2304.13060*, 2023. 2, 4

[48] Steven T Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21:1112–1130, 2014. 2, 4

[49] Yiwei Qin, Graham Neubig, and Pengfei Liu. Searching for effective multilingual fine-tuning methods: A case study in summarization, 2022. 1

[50] Haoxuan Qu, Xiaofei Hui, Yujun Cai, and Jun Liu. Lmc: Large model collaboration with cross-assessment for training-free open-set object recognition. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[51] Acquaviva Sam. Hyperbolic vq-vaes. 2, 3, 5

[52] 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. 7, 8

[53] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7912–7921, 2019. 1, 2, 3

[54] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019. 1, 2, 3, 7

[55] Stuart M Shieber. Evidence against the context-freeness of natural language. In *The Formal complexity of natural language*, pages 320–334. Springer, 1985. 2, 4

[56] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 3, 7

[57] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 3, 6

[58] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 4041–4049, 2015. 2

[59] Lei Wang and Piotr Koniusz. 3mformer: Multi-order multi-mode transformer for skeletal action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5620–5631, 2023. 1, 3

[60] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 2

[61] Xuanhan Wang, Yan Dai, Lianli Gao, and Jingkuan Song. Skeleton-based action recognition via adaptive cross-form learning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1670–1678, 2022. 7

[62] Xinghan Wang, Xin Xu, and Yadong Mu. Neural koopman pooling: Control-inspired temporal dynamics encoding for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10597–10607, 2023. 1, 3, 7

[63] Shenghua Wei, Yonghong Song, and Yuanlin Zhang. Human skeleton tree recurrent neural network with joint relative motion feature for skeleton based action recognition. In *2017 IEEE international conference on image processing (ICIP)*, pages 91–95. IEEE, 2017. 2, 5

[64] Rongxiang Weng, Wen Sen Cheng, and Min Zhang. G-tuning: Improving generalization of pre-trained language models with generative adversarial network. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4747–4755, 2023. 2

[65] Wangmeng Xiang, Chao Li, Yuxuan Zhou, Biao Wang, and Lei Zhang. Generative action description prompts for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10276–10285, 2023. 3, 7

[66] Wentian Xin, Qiguang Miao, Yi Liu, Ruyi Liu, Chi-Man Pun, and Cheng Shi. Skeleton mixformer: Multivariate topology representation for skeleton-based action recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2211–2220, 2023.

[67] Haojun Xu, Yan Gao, Zheng Hui, Jie Li, and Xinbo Gao. Language knowledge-assisted representation learning for skeleton-based action recognition. *arXiv preprint arXiv:2305.12398*, 2023. 3

[68] Kailin Xu, Fanfan Ye, Qiaoyong Zhong, and Di Xie. Topology-aware convolutional neural network for efficient skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2866–2874, 2022. 3

[69] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2272–2281, 2017. 7

[70] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 1, 2, 3, 7, 8

[71] Di Yang, Rui Dai, Yaohui Wang, Rupayan Mallick, Luca Minciullo, Gianpiero Francesca, and Francois Bremond. Selective spatio-temporal aggregation based pose refinement system: Towards understanding human activities in real-world videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2363–2372, 2021. 7

[72] Di Yang, Yaohui Wang, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, and François Brémond. Unik: A unified framework for real-world skeleton-based action recognition. *arXiv preprint arXiv:2107.08580*, 2021. 1, 7

[73] Jiahang Zhang, Lilang Lin, and Jiaying Liu. Hierarchical consistent contrastive learning for skeleton-based action recognition with growing augmentations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3427–3435, 2023. 3

[74] Jiahui Zhang, Fangneng Zhan, Christian Theobalt, and Shijian Lu. Regularized vector quantization for tokenized image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18467–18476, 2023. 3

[75] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. *arXiv preprint arXiv:2301.06052*, 2023. 2, 3, 6, 7

[76] Yuhan Zhang, Bo Wu, Wen Li, Lixin Duan, and Chuang Gan. Stst: Spatial-temporal specialized transformer for skeleton-based action recognition. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3229–3237, 2021. 3

[77] Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators. *arXiv preprint arXiv:2306.10900*, 2023. 2, 3, 6

[78] Huanyu Zhou, Qingjie Liu, and Yunhong Wang. Learning discriminative representations for skeleton based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 1, 7

[79] Lanyun Zhu, Tianrun Chen, Deyi Ji, Jieping Ye, and Jun Liu. Llafs: When large-language models meet few-shot segmentation. *arXiv preprint arXiv:2311.16926*, 2023. 3

[80] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *Proceedings*

of the IEEE/CVF International Conference on Computer Vision, 2023. 1, 3, 7

[81] Yisheng Zhu, Hui Shuai, Guangcan Liu, and Qingshan Liu. Multilevel spatial–temporal excited graph network for skeleton-based action recognition. *IEEE Transactions on Image Processing*, 32:496–508, 2022. 7

[82] G. K. Zipf. The psycho-biology of language, 1935. 2, 4