Exploring Vision Transformers for 3D Human Motion-Language Models with Motion Patches

Qing Yu Mikihiro Tanaka Kent Fujiwara
LY Corporation
{yu.qing, mikihiro.tanaka, kent.fujiwara}@lycorp.co.jp

Abstract

To build a cross-modal latent space between 3D human motion and language, acquiring large-scale and high-quality human motion data is crucial. However, unlike the abundance of image data, the scarcity of motion data has limited the performance of existing motion-language models. To counter this, we introduce "motion patches", a new representation of motion sequences, and propose using Vision Transformers (ViT) as motion encoders via transfer learning, aiming to extract useful knowledge from the image domain and apply it to the motion domain. These motion patches, created by dividing and sorting skeleton joints based on body parts in motion sequences, are robust to varying skeleton structures, and can be regarded as color image patches in ViT. We find that transfer learning with pre-trained weights of ViT obtained through training with 2D image data can boost the performance of motion analysis, presenting a promising direction for addressing the issue of limited motion data. Our extensive experiments show that the proposed motion patches, used jointly with ViT, achieve state-of-the-art performance in the benchmarks of text-to-motion retrieval, and other novel challenging tasks, such as cross-skeleton recognition, zero-shot motion classification, and human interaction recognition, which are currently impeded by the lack of data.

1. Introduction

The cross-modal analysis of 3D human motion and natural language has opened up new avenues for tasks such as motion recognition [10, 38, 46, 49, 50, 52, 54, 60] and text-to-motion synthesis [3, 36, 37], which can benefit the applications like animating avatars or humans [18, 53]. The key to these tasks is constructing a cross-modal latent space that captures the intricate relationship between human motions and language semantics, allowing systems to interpret and generate human-like motions based on textual descriptions.

Despite the promising advancements in this area, one of the most challenging aspects is the scarcity of data, because the process of collecting and annotating 3D human motion data is labor-intensive and time-consuming. While recent years have seen an increase of motion capture datasets [31], some of which are categorized by classes [28] or even labeled with free text [39, 40], these resources are still not sufficient for deep learning algorithms to fully understand human motions. Moreover, because various motion capture systems and skeleton structures are used in different datasets, it has been difficult to build a large-scale dataset with a unified representation.

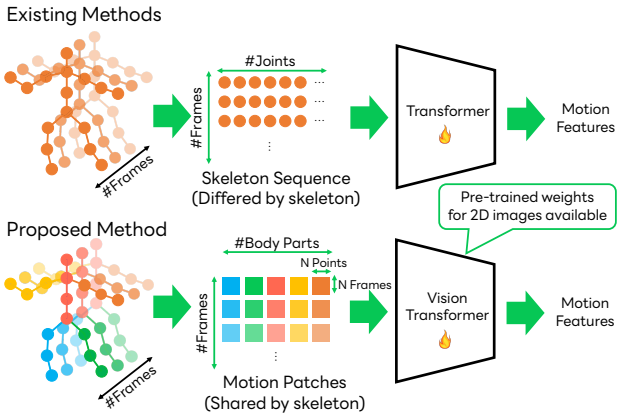

Figure 1. Overview of the existing methods and the proposed method. The existing methods train an original Transformer with the joint information from the motion sequences directly, while the proposed method converts them into motion patches and then trains the ViT, which can be initialized with pre-trained weights.

To build a motion-language model, existing methods [38, 50] attempt to incorporate text embeddings into motion autoencoders. Due to the lack of large-scale data, they train the motion encoder from scratch on each dataset and try to use motion synthesis in autoencoders to improve the motion features. Tevet et al. [50] attempt to apply pre-trained image-language models [41] to motion data, but they only render a single frame as the input image. Consequently, these methods are not yet robust enough to handle the vari-

ations and subtleties present in 3D human motions. They also need to deal with different skeleton structures by training an individual model for each dataset separately, despite all these data representing human motion. This leads to low performance on the small-scale datasets.

To overcome these challenges, we introduce an approach to building motion-language models by leveraging Vision Transformers (ViT) [9] as the motion encoder. This approach extends the conventional use of ViTs, which were originally designed for 2D image classification, to the more complex domain of 3D human motion analysis. With the transfer learning of ViT pretrained on ImageNet [43] to motion data, the training process of the motion encoder can be accelerated, while at the same time, overcoming the issue of limited data scale and achieving better correspondence between the motion features and the language features.

To efficiently transfer knowledge of the image domain to the motion domain, we also propose "motion patches", a novel unified representation for various skeleton structures in motion sequences. We design the motion patches to be likened to image patches in ViT, with joint positions in xyz coordinates simply converted to image colors in rgb space. We first partition the joints of the skeleton into five body parts: torso, left arm, right arm, left leg, and right leg. Then, motion patches are formed by sampling N points from each body part through linear interpolation, and stacking these points for each part across N frames by sliding window. This results in a patch for each part with a size of $N \times N$. We then train a motion-language model with a contrastive learning framework [41]. The comparison between existing methods and the proposed method is shown in Fig. 1.

We evaluate the versatility and effectiveness of our proposed method through comprehensive experiments and applications of motion-language tasks. This study makes the following contributions:

- We propose a new framework for building motion-language models using ViT, which extends the application of ViT to obtain a cross-modal latent space between motions and language.
- We introduce a novel method of representing 3D human motion data as "motion patches", which can be processed by the ViT architecture with its pre-trained weights for transfer learning, and are also resilient to variations in human skeleton structures.
- Our approach not only significantly improves the performance of text-to-motion retrieval, but also illustrates the potential for other novel applications, such as cross-skeleton recognition, zero-shot motion classification, and human interaction recognition.

2. Related Works

Motion-Language Datasets. While human motion modeling has gained interest in linking language and 3D body motions, there is a scarcity of motion-language datasets. Although datasets do exist for action recognition [45] and pose estimation [20], they lack detailed textual descriptions for each motion. Notably, the KIT dataset [39] offers 11 hours of motion capture sequences, each paired with descriptive sentences. The recently released HumanML3D dataset [15] provides around 29 hours of motion data with natural language labels for AMASS [31] and HumanAct12 [14] collections. Compared to the scale of image-text pairs used for training image-language models like CLIP (*e.g.*, datasets with 400 million images), motion-text pairs remain notably limited in scale (*e.g.*, 14,616 motions in HumanML3D). Creating motion-language datasets presents challenges, including the need for expensive motion capture and annotation systems, as well as issues related to variations in skeleton structures across different datasets.

Method	Input Type	Motion encoder	Unified Representation
MotionCLIP [50]	Motion + Image	Scratch Transformer + Pre-trained CLIP	✗
TMR [38]	Motion	Scratch Transformer	✗
Proposed	Motion Patch	Pre-trained ViT	✓

Table 1. Summary of recent related methods for motion-language models. Only our proposed method utilizes pre-trained motion encoders and a unified representation for various skeleton structures.

Motion-Language Models. In recent years, vision-language foundation models have garnered significant attention, driven by the availability of vast collections of image-text pairs gathered from the internet. These models have adopted various pre-training schemes [6, 26, 30, 47]. A recent representative in this field is CLIP [41], which aims to learn joint representations of vision and language by training on a large-scale dataset of image-text pairs. However, the field of motion-language models is relatively less explored. Petrovich et al. [38] use contrastive training during motion generation to align text features with motion features. Because of the limited scale of motion-text pairs, these methods are trained from scratch and cannot capture the differences between similar motions. Moreover, these methods cannot be applied to cross-skeleton recognition, as different motion encoders need to be trained for each dataset. There are some attempts to utilize external knowledge from different modalities to analyze human motion. Tevet et al. [50] attempt to render a single frame as a static image to be used as input to CLIP, in order to obtain visual features and align them with motion features, but the performance is limited by the use of a single frame.

Motion Generation and Retrieval. Motion-language models find valuable application in text-conditioned motion generation. Unlike unconstrained motion generation [55, 59], action-conditioned [14, 36] or text-conditioned models [1, 3, 5, 11, 16, 22, 23, 37, 48, 51, 57] introduce semantic controls to generate motion sequences correspond-

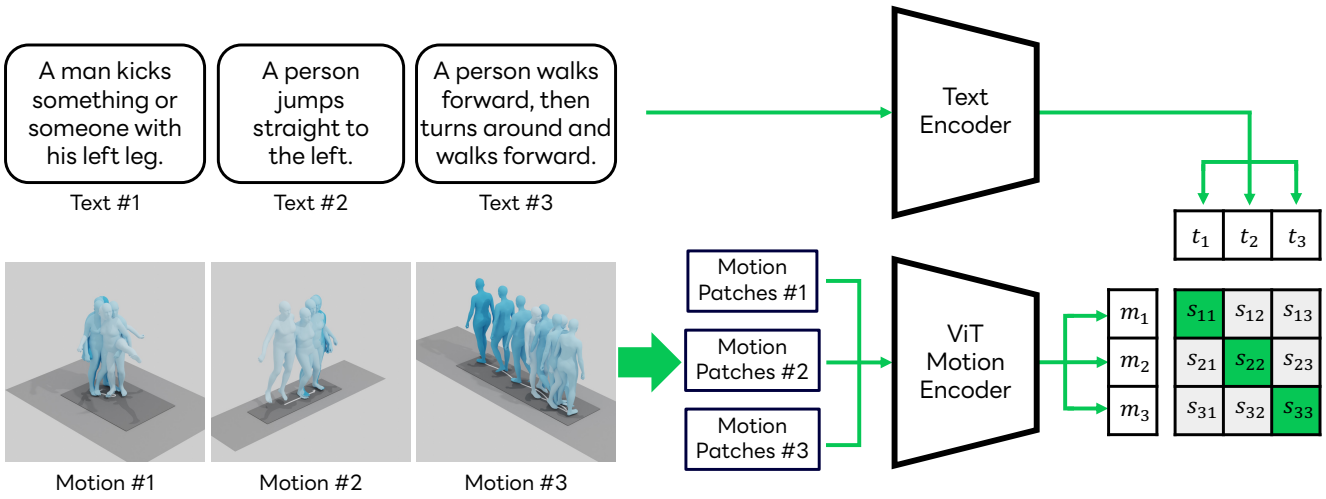

Figure 2. Overview of the proposed framework, which consists of a motion encoder and a text encoder. We transform the raw motion sequences into motion patches as the input of the ViT-based motion encoder. We calculate the similarity matrix between text-motion pairs within a batch to train the model. To illustrate this concept, we provide an example batch containing three samples for clarity.

ing to input textual descriptions. Recent advancements in text-to-image generation through diffusion generative models [8, 42] have led to methods such as [5, 51, 58], aiming to apply diffusion models to text-to-motion generation using the HumanML3D dataset, along with other approaches based on Large Language Models [22, 57]. Text-to-motion retrieval [38] is an alternative and potentially complementary approach to generating motions corresponding to a given textual description, as a retrieval model can always return a realistic motion. Text-to-motion retrieval is also used as a tool to evaluate the performance of text-to-motion generation [7, 15, 22, 58]. We also use the task of text-to-motion retrieval as an important evaluation of the proposed method.

Transfer Learning. Transfer learning involves taking models trained for a specific task using a large dataset, and extending their capabilities to address new tasks by leveraging prior knowledge to extract relevant features. Some examples of these attempts can be found in image segmentation [19] and medical image analysis [13, 32]. Besides the image domain, transfer learning using ImageNet [43] pre-trained weights also performs well in video recognition [4] and even in audio classification [12, 34], where time sequences are transformed into images as the input of the model. Some methods [2, 25] attempt to use convolutional neural networks for action recognition. However, our focus is on leveraging the pre-trained knowledge from the image domain to construct motion-language models.

To address the challenges of motion-language models, we aim to transfer the ViT pre-trained in the image domain to the motion domain with a unified representation of motion sequences, to overcome the data scale problem. We summarize the differences between the proposed method and related motion-language models in Table 1.

3. Method

3.1. Problem Statement

Given a set of motion sequences $\mathcal{M}$ and a set of captions $\mathcal{T}$, our target is to learn a function $s(m_i, t_j)$ to calculate the similarity between the motion $m_i \in \mathcal{M}$ and the caption $t_j \in \mathcal{T}$. The objective of the $s(m_i, t_j)$ is to calculate a high similarity for relevant motion-text pairs and a low similarity score for irrelevant ones. The motion sequence $m_i \in \mathcal{M}$ is represented as a sequence of skeleton joints in this paper. Formally, the motion sequence is denoted by $m_i \in \mathbb{R}^{T \times J \times 3}$, where T represents the length of the sequence, $J \times 3$ represents the position of the skeleton joints in Cartesian coordinates, (x, y, z).

To build a motion-language model, we adopt the CLIP framework [41], which consists of a motion encoder $\mathcal{F}_M$ and a language model $\mathcal{F}_T$. Using these encoders, we encode the motion sequence m_i as $\mathcal{F}_M(m_i)$ and the caption t_j, and then calculate the similarity as follows:

$$s(m_i, t_j) = \frac{\mathcal{F}_M(m_i) \cdot \mathcal{F}_T(t_i)}{\|\mathcal{F}_M(m_i)\| \|\mathcal{F}_T(t_i)\|}. \quad (1)$$

The overall architecture of the proposed framework is shown in Fig. 2.

3.2. Motion Patches

To extract spatial-temporal information in motion sequences as motion features, and to enable effective transfer

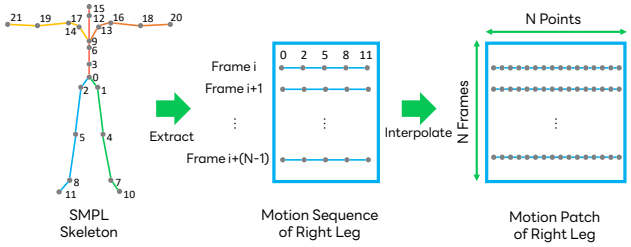

Figure 3. Process of building the motion patches for each motion sequence. Given a skeleton, we mark different body parts in different colors. We show the method to construct the motion patch of the right leg. The same process is applied to other body parts.

of knowledge from the image domain via pre-trained models, motions need to be represented in a similar manner as images. However, in contrast to image data usually having a unified representation of size $224 \times 224 \times 3$ for deep learning models, the size of motion sequences is $T \times J \times D$ as mentioned in Section 3.1. Because the number of the frames T differs for each sequence and the number of the joints J depends on the skeleton structure in the dataset, there is no consensus on how to obtain a unified representation for motion data in different skeleton structures.

We propose "motion patches" as a new representation, which can be further used as input to ViT [9] for motion feature extraction. To build motion patches similar to image patches with size $N \times N$ in ViT, we divide the joints of 3D motion skeletons into body parts, interpolate between joints in each body part to obtain N sample points, and use the N consecutive frames in the sequence as shown in Fig. 3. First, the joints are partitioned into five body parts: the torso (including the head), left arm, right arm, left leg, and right leg. This type of partitioning is commonly used [21] and can be implemented on any human skeleton. Each body part comprises a subset of joints corresponding to that part of the body according to the kinematic chain of the skeleton.

Next, within each body part, we arrange the joints based on their distance from the torso. For example, in the case of arms, we order the joints as the upper chest $\rightarrow$ the shoulder $\rightarrow$ upper arm $\rightarrow$ lower arm $\rightarrow$ hand. This sequence maintains the spatial structure of the skeleton. We standardize the number of sample points in each body part to N using linear interpolation. To normalize these sample points as image data, we calculate the mean and variance of each point across the dataset and perform the z-score normalization using these mean and variance values.

Finally, we form "motion patches" by stacking sequences of sample point positions across N frames. We repeat this process for every sequence of N frames using a sliding window, creating a series of motion patches. These patches, which are robust to variations in skeleton structures, can be analogized to image patches in ViT, and allow us to represent skeletons from various structures in a unified

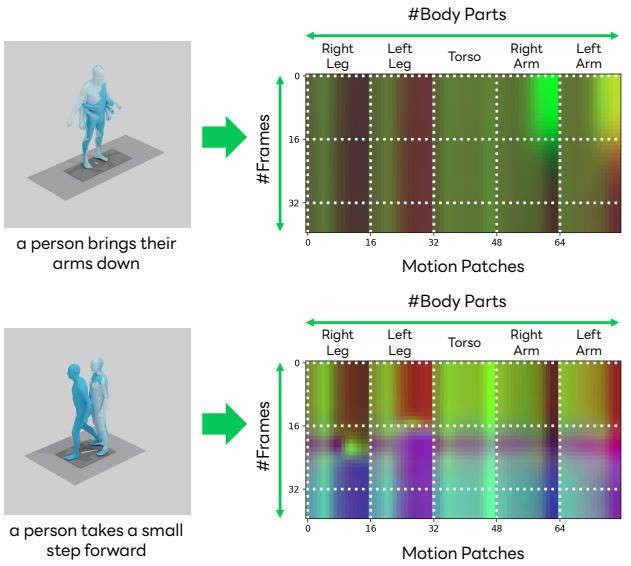

Figure 4. Visualization of motion patches by regarding the joint coordinates as RGB pixels. We show the rendered motions and their text label on the left and the processed motion patches on the right. We can observe different motions reflected in different motion patches.

format.

We provide a visualization of motion patches by depicting them as RGB images in Fig. 4. We interpret joint coordinates as RGB pixels to create a visual representation of each motion patch. The figure displays the rendered motions and their corresponding text labels on the left, and the processed motion patches on the right. We can observe different motions resulting in distinct motion patches, demonstrating the capacity of our method to capture unique characteristics of each motion in the form of motion patches.

3.3. Motion Encoder

For image data, many well-established architectures and pre-trained models can be used for CLIP. Meanwhile, for motion data, there is no standard architecture and no large-scale pre-trained model. Existing methods [38, 50] use their original motion encoders and train them from scratch to extract the motion representations. However, with the novel motion patches proposed in Section 3.2, we are able to encode motions by extending ViT for 2D images to 3D motion data to overcome the limited scale of motion data.

ViT first extracts non-overlapping image patches from the image data. Then, a projection head is used to project these patches onto 1D tokens. The transformer architecture is used to model the interaction between each image patch to obtain the final representation. To apply ViT to motion sequences, we first transfer the motion sequences into motion patches and then regard these motion patches as image patches. In this paper, we adopt the ViT-B/16 with 12 lay-

ers and the patch size 16 pre-trained on ImageNet-21k [43] as our motion encoder. Hence, we set $N = 16$ to obtain the motion patches with size 16×16. We have additionally included an investigation of the choices related to the ViT backbone and patch sizes in the supplementary material.

Following ViT and CLIP, the [class] token is added to the inputs and we resize the position embedding to match the number of patches. The output from the [class] token is projected onto a multi-modal embedding space as the motion representation.

3.4. Text Encoder

In the context of text encoding, it is crucial to extract features related to motion. Following TMR [38], we adopt DistilBERT [44] for this purpose, utilizing a pre-trained model with a projection head. The output from the [class] token is used as the text representation. An alternative is utilizing the text encoder of CLIP [41], commonly used in motion generation methods [51, 58]. However, vision-language models, including CLIP, face challenges in distinguishing between entities and verbs [17, 35, 56]. Despite exploring this option, our experiments showed that DistilBERT outperformed CLIP, with detailed comparisons available in the supplementary material.

3.5. Training Strategy

Given a batch of B (motion sequence, text) pairs, the model needs to generate and optimize $B \times B$ similarities. We use a symmetric cross-entropy loss over these similarity scores to train the parameters of the model as follows:

$$\mathcal{L}_{m2t} = -\frac{1}{B}\sum_{i}^{B} \log \frac{\exp(s(m_i, t_i)/\tau)}{\sum_{j=1}^{B} \exp(s(m_i, t_j)/\tau)}, \quad (2)$$

$$\mathcal{L}_{t2m} = -\frac{1}{B}\sum_{i}^{B} \log \frac{\exp(s(m_i, t_i)/\tau)}{\sum_{j=1}^{B} \exp(s(m_j, t_i)/\tau)}, \quad (3)$$

$$\mathcal{L} = \mathcal{L}_{m2t} + \mathcal{L}_{t2m}, \quad (4)$$

where τ is the temperature parameter. The loss $\mathcal{L}$ is the sum of motion-to-text loss $\mathcal{L}_{m2t}$ and text-to-motion loss $\mathcal{L}_{t2m}$.

4. Experiments

4.1. Datasets

We utilize two standard datasets in our experiments: the HumanML3D dataset [15] and the KIT Motion-Language dataset [39].

HumanML3D Dataset: The HumanML3D dataset enriches the AMASS [31] and HumanAct12 [14] motion capture collections with natural language labels describing the motions. We follow the same motion pre-processing method proposed in [15]. Furthermore, the dataset is augmented through the mirroring of both left and right motions, along with their corresponding textual descriptions. Subsequently, following the official dataset split, we acquire a total of 23,384, 1,460, and 4,380 motions for the training, validation, and test sets, respectively. On average, each motion receives 3.0 distinct textual annotations. During the training phase, we randomly choose one annotation as the matching text, while for testing, we only use the first one.

KIT Motion-Language Dataset (KIT-ML): The KIT-ML dataset, which primarily focuses on locomotion, is also derived from motion capture data. To prepare the motion data for analysis, we apply the identical pre-processing procedure as employed in the HumanML3D dataset. The dataset is partitioned into training, validation, and test sets, consisting of 4,888, 300, and 830 motions, respectively. On average, each motion is annotated 2.1 times.

4.2. Evaluation Protocol

To evaluate the performance of the motion-language model, we adopt the retrieval task between the motion sequence and the text description. Following [38], our evaluation of retrieval performance employs standard metrics, specifically Recall at various ranks (R@1, R@2, etc.), for both text-to-motion and motion-to-text tasks. Recall at rank k indicates the percentage of instances where the correct label appears within the top k results [1], with higher values indicating better performance. Additionally, we calculate the median rank (MedR), where a lower value shows better performance. It is important to note that the evaluation of retrieval performance is conducted using an unseen gallery of real motions, specifically the test set.

We used several evaluation protocols to calculate Recall, primarily altering the composition of the gallery set:

All: In this protocol, the entire test set is used without any modifications. However, repetitive texts across motions or minor textual differences (*e.g.*, "person" vs. "human", "walk" vs. "walking") will affect the results. We use this protocol as the default protocol in this paper.

Small Batches: This protocol is designed by Guo et al. [15]. It involves randomly selecting batches of 32 motion-text pairs and then reporting the average performance. While this approach introduces randomness, it serves as a benchmark for comparison. It is worth noting that a gallery size of 32 is relatively manageable compared to the other protocols, making it a less challenging scenario.

4.3. Implementation Details

In our experiments, we employ the Adam optimizer [24] with a learning rate of 10^{-5} for the text encoder, 10^{-4} for the motion encoder, and 10^{-3} for the projection head. A

[1] Due to the existence of mirroring augmented samples in the test data, some samples have two correct answers in the gallery, *i.e.*, the original motion and its mirrored counterpart may share the same text description. We accounted for this factor, whereas TMR [38] metrics overlooked it.

Protocol	Methods	Text-motion retrieval						Motion-text retrieval					
		R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓
All	TEMOS† [37]	2.12	4.09	5.87	8.26	13.52	173.0	3.86	4.54	6.94	9.38	14.00	183.25
	T2M† [15]	1.80	3.42	4.79	7.12	12.47	81.00	2.92	3.74	6.00	8.36	12.95	81.50
	TMR [38]	8.92	12.04	16.33	22.06	33.37	25.00	9.44	11.84	16.90	22.92	32.21	26.00
	Ours (scratch)	8.46	12.76	16.22	23.56	35.27	23.00	9.63	11.78	16.58	22.87	33.57	25.00
	Ours	**10.80**	**14.98**	**20.00**	**26.72**	**38.02**	**19.00**	**11.25**	**13.86**	**19.98**	**26.86**	**37.40**	**20.50**
Small batches	TEMOS† [37]	40.49	53.52	61.14	70.96	84.15	2.33	39.96	53.49	61.79	72.40	85.89	2.33
	T2M† [15]	52.48	71.05	80.65	89.66	96.58	1.39	52.00	71.21	81.11	89.87	96.78	1.38
	TMR [38]	67.45	80.98	86.22	91.56	95.46	1.03	68.59	81.73	86.75	91.10	95.39	1.02
	Ours (scratch)	67.61	82.40	86.79	91.75	95.97	1.01	67.11	80.04	85.86	91.86	95.98	1.00
	Ours	**71.61**	**85.81**	**90.02**	**94.35**	**97.69**	**1.00**	**72.11**	**85.26**	**90.21**	**94.44**	**97.76**	**1.00**

Table 2. Results of text-to-motion and motion-to-text retrieval benchmark on HumanML3D. The results of methods marked with † are sourced from TMR [38]. Ours (scratch) denotes the proposed method trained from scratch without using pre-trained ViT weights.

Protocol	Methods	Text-motion retrieval						Motion-text retrieval					
		R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@2 ↑	R@3 ↑	R@5 ↑	R@10 ↑	MedR ↓
All	TEMOS† [37]	7.11	13.25	17.59	24.10	35.66	24.00	11.69	15.30	20.12	26.63	36.39	26.50
	T2M† [15]	3.37	6.99	10.84	16.87	27.71	28.00	4.94	6.51	10.72	16.14	25.30	28.50
	TMR [38]	10.05	13.87	20.74	30.03	44.66	14.00	11.83	13.74	22.14	29.39	38.55	16.00
	Ours (scratch)	10.41	17.01	24.43	31.55	46.50	13.00	12.13	14.83	22.54	30.83	40.12	15.00
	Ours	**14.02**	**21.08**	**28.91**	**34.10**	**50.00**	**10.50**	**13.61**	**17.26**	**27.54**	**33.33**	**44.77**	**13.00**
Small batches	TEMOS† [37]	43.88	58.25	67.00	74.00	84.75	2.06	41.88	55.88	65.62	75.25	85.75	2.25
	T2M† [15]	42.25	62.62	75.12	87.50	96.12	1.88	39.75	62.75	73.62	86.88	95.88	1.95
	TMR [38]	50.00	69.14	78.02	87.97	94.87	1.50	51.21	69.53	78.64	89.00	95.31	1.50
	Ours (scratch)	51.13	70.15	78.96	88.06	95.74	1.43	53.12	70.31	79.40	89.34	95.59	1.33
	Ours	**53.55**	**71.30**	**79.82**	**88.92**	**96.29**	**1.36**	**54.54**	**72.15**	**79.68**	**89.35**	**96.11**	**1.31**

Table 3. Results of text-to-motion and motion-to-text retrieval benchmark on KIT-ML.

batch size of 256 is used during the training. The latent dimension of the embeddings after the projection is set to 256. We set the temperature parameter to 0.07 following CLIP. The number of frames in each motion sequence is limited to 224, following the existing methods [38, 50], which means $14 \times 5 = 70$ motion patches are used as the input of ViT.

4.4. Results

In our evaluation of text-to-motion and motion-to-text retrieval benchmark across HumanML3D (Table 2) and KIT-ML (Table 3) datasets, encompassing all evaluation protocols, we provide comparisons against prior works, specifically TEMOS [37], T2M [15], TMR [38] and the proposed method trained from scratch without using pre-trained ViT weights. The experimental results of TEMOS [37] and T2M [15] are sourced from the TMR [38] paper. Meanwhile, we re-evaluate the official models of TMR [38] using our evaluation code to ensure a fair comparison.

It is important to note that TEMOS [37] is not explicitly designed for retrieval tasks. The cross-modal embedding space of TEMOS [37] is primarily trained with positive pairs. In contrast, T2M [15] applied their method to retrieval by employing contrastive learning, which includes negative pairs as well. TMR [38] is the state-of-the-art method for text-to-motion retrieval, which extends TEMOS by incorporating a contrastive loss [33] between the motion features and the text features in the latent space.

Remarkably, our model consistently outperforms prior work across all evaluation sets in various degrees of difficulty. This indicates that our model can capture the nuanced nature of motion descriptions. The substantial performance enhancements we achieve over the state-of-the-art can be attributed to several factors: (1) the design of motion patches to capture the temporal-spatial motion representation and (2) the utilization of ViT and transferring its pre-trained weights to the motion domain. In the subsequent sections, we conduct controlled experiments to analyze the impact of these components on our results.

4.5. Ablation Studies

In this section, we explore various settings to better understand the factors influencing the performance of our model.

Pre-trained ViT: We conduct experiments to compare the performance of our model when utilizing pre-trained ViT representations against a setting where ViT pre-training is not employed.

Motion Representation: Another aspect we investigate is the use of motion representations. We investigate differ-

Dataset: HumanML3D

Pre-trained ViT	Motion Patches	Text-motion retrieval				Motion-text retrieval			
		R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
✓	✓	**10.80**	**26.72**	**38.02**	**19.00**	**11.25**	**26.86**	**37.40**	**20.50**
✗	✓	8.46	23.56	35.37	23.00	9.63	22.87	33.57	25.00
✓	✗	8.36	22.84	33.62	24.00	8.81	21.67	31.35	29.00
✗	✗	8.58	21.54	32.87	25.00	8.46	21.96	30.79	30.00

Dataset: KIT-ML

Pre-trained ViT	Motion Patches	Text-motion retrieval				Motion-text retrieval			
		R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
✓	✓	**14.02**	**34.10**	**50.00**	**10.50**	**13.61**	**33.33**	**44.77**	**13.00**
✗	✓	10.41	31.55	46.50	13.00	12.13	30.83	40.12	15.00
✓	✗	9.54	31.72	45.86	14.00	11.01	27.31	37.41	18.00
✗	✗	9.87	30.37	44.21	16.00	10.14	26.86	36.43	21.00

Table 4. Results of ablation studies. We experiment with different settings (1) with/without the pre-trained ViT and (2) whether to use motion patches as the representation of the motion.

ent scenarios where we either employ the proposed motion patches as the input for ViT or directly feed the raw motion sequences into the Transformer component, along with positional encodings.

The results of HumanML3D and KIT-ML are shown in Table 4. It is noticeable that when the motion patches are used, training the model from pre-trained weights leads to much better results than training from scratch. Compared with using motion patches as input, using motion sequences without preprocessing leads to worse performance. This analysis shows the impact of ViT pre-training on the capabilities of our model and the advantages that our motion patches bring to retrieval tasks.

4.6. Qualitative results

In Fig. 5, we present the qualitative results of text-to-motion retrieval on the entire test set of HumanML3D. Each query text is displayed on the left, and on the right, we showcase the top-3 retrieved motions along with their corresponding ground-truth text labels. The gallery of motions for retrieval remains unseen during training.

In the first two examples, we successfully retrieve the ground-truth motion in the top-2 results. Note that in the second example, the differences between the motions are very small and they all present motions similar to the text query. For the free-form prompt in the last example, where the exact text is not present in the gallery, our method also succeeds in retrieving correct motions.

5. Applications

5.1. Cross-skeleton Recognition

One advantage of the proposed motion patches is that the motion sequences from different skeleton structures can be transferred into a unified representation. For example, motion in HumanML3D [15] follows the skeleton structure of SMPL [29] with 22 joints as shown in Fig. 3, while poses

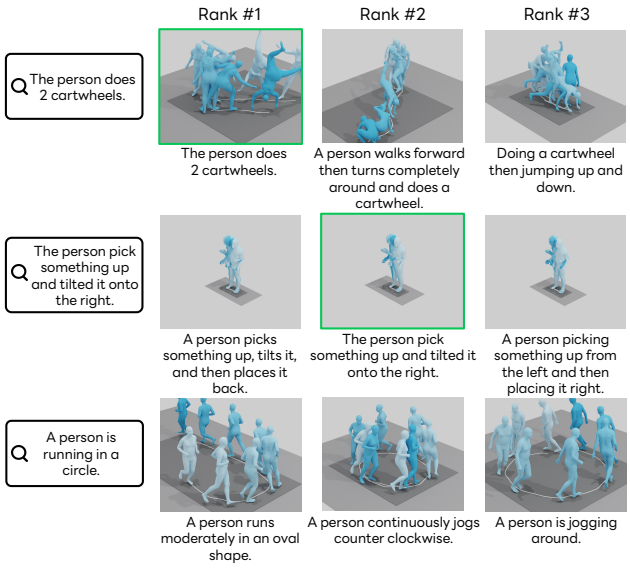

Figure 5. Qualitative results of text-to-motion retrieval. For each query, we show the retrieved motions ranked by text-motion similarity and their accompanying ground-truth text labels. Note that these descriptions are not used in the retrieval process. All motions in the gallery are from the test set and were unseen during training. For the first two examples, the text queries are sampled from the data. For the last example, we query with a free-form text.

have 21 joints in KIT-ML, and the skeleton structure of KIT-ML is different from SMPL (detailed in the supplementary). Existing methods cannot deal with these two datasets simultaneously because the dimension of the input vector varies according to the dimension of the pose vector. However, our method is able to convert the motion sequence to 16×16 motion patches according to the kinematic chain of each body part, which means the motion features learned on a dataset can be transferred to another dataset even when the skeleton structure is different.

To evaluate the performance of the proposed method for cross-skeleton recognition, we prepare two scenarios. The first one is a zero-shot setting, where we directly apply the motion-language model trained on the HumanML3D dataset to the text-to-motion retrieval task of the KIT-ML dataset. The other one is a transfer learning setting, where we further fine-tune the HumanML3D model with the KIT-ML dataset, because the scale of the KIT-ML dataset is smaller than that of the HumanML3D dataset.

The results are shown in Table 5, which compares the zero-shot setting and the transfer learning setting with other existing methods and the proposed method trained on KIT-ML. The performance of zero-shot prediction with the HumanML3D model is lower than the models trained on KIT-ML, because the language domains (*i.e.*, KIT-ML is more focused on locomotion descriptions) and the skeleton struc-

Dataset: KIT-ML

Method	Training Dataset	Text-motion retrieval				Motion-text retrieval			
		R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
TEMOS† [37]	KIT-ML	7.11	24.10	35.66	24.00	11.69	26.63	36.39	26.50
T2M† [15]	KIT-ML	3.37	16.87	27.71	28.00	4.94	16.14	25.30	28.50
TMR [38]	KIT-ML	10.05	30.03	44.66	14.00	11.83	29.39	38.55	16.00
Ours	KIT-ML	14.02	34.10	50.00	10.50	13.61	33.33	44.77	13.00
Ours	HumanML3D	7.35	20.98	34.33	23.50	7.90	17.71	25.88	31.00
Ours	Transferred	**15.28**	**38.71**	**52.65**	**9.00**	**16.51**	**35.78**	**47.47**	**11.00**

Table 5. Results of cross-skeleton recognition. We evaluate the text-to-motion and motion-to-text retrieval on the KIT-ML dataset with the HumanML3D model and the transferred model. The transfer learning method achieves better performance than the method of training only on KIT-ML.

tures of these two datasets are different. However, it still achieves acceptable performance compared to TEMOS [37] and T2M [15], especially on the task of text-motion retrieval. It is noticeable that the transfer learning method obtained the best results, which shows that training the model with HumanML3D is helpful in recognizing the motion sequences of KIT-ML. This highlights the potential of our approach to improve the performance on small-scale datasets by pre-training the model on large-scale datasets.

5.2. Zero-shot Motion Classification

Furthermore, we demonstrate the effectiveness of the semantically structured latent spaces generated by our motion-language model via action recognition. We follow the BABEL 60-classes benchmark [40], containing 10,892 sequences, and 20% of them are used as test sets. We pre-processed the motion sequences with the same procedure as HumanML3D [15]. Because this is a zero-shot classification setting, we did not train the model with BABEL and only applied the model trained on HumanML3D to the test data. For the text prompts, the action names in BABEL are used as "A person {action}". We calculate the cosine distance between a given motion and all 60 text prompts.

In Table 6, we show a comparison of the Top-1 and Top-5 accuracy achieved by our zero-shot classifier with that of the 2s-AGCN classifier [40] and MotionCLIP [50]. As evident from the results, our framework performs comparably to the state-of-the-art supervised methods, despite the fact that our method was not initially designed for action recognition tasks nor trained on the action label set of BABEL. This highlights the versatility and adaptability of our approach across various applications.

5.3. Human Interaction Recognition

Besides the recognition of single-person motion, the proposed method can also be applied to multi-person motion recognition. We conduct the experiments using InterHuman [27], a motion-language dataset that comprises a diverse set of 3D motions involving interactions between two individuals. The dataset is split into 6,222 sequences for training and 1,557 sequences for testing. We processed the motion sequences of each individual with the same procedure as HumanML3D [15]. To obtain the features of the interactions, we apply a shared motion encoder to each person and simply concatenate the motion features before the projection head. We evaluate the performance of the proposed method via text-to-motion retrieval. The results are shown in Table 7 and our method outperforms TMR [38].

Method	Training Dataset	Zero-shot	Modality	Top-1 Acc.	Top-5 Acc.
2s-AGCN [46]	BABEL	✗	M	41.14	**73.18**
MotionCLIP [50]	BABEL	✗	M+L	40.90	57.71
TMR [38]	HumanML3D	✓	M+L	30.13	41.52
Ours	HumanML3D	✓	M+L	**41.33**	68.97

Table 6. Results of zero-shot motion classification. Modality with motion only and motion language are denoted as M and M+L, respectively. When applying our proposed method for zero-shot classification, we achieve performance results that are closely aligned with those of the 2s-AGCN classifier trained with supervision on the BABEL-60 benchmark.

Method	Text-motion retrieval				Motion-text retrieval			
	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓	R@1 ↑	R@5 ↑	R@10 ↑	MedR ↓
TMR [38]	5.38	15.64	24.40	34.00	5.13	15.26	25.65	33.00
Ours	**9.51**	**21.27**	**32.41**	**27.00**	**8.26**	**22.65**	**32.66**	**24.00**

Table 7. Results of human interaction recognition. For TMR [38] and our method, we concatenate the motion features of each person and get the multi-person motion feature through a projection head of the concatenated feature.

6. Limitations

In this paper, we primarily evaluate our method on motion recognition, focusing on text-to-motion retrieval. Future work includes applying our method to text-to-motion generation. Despite leveraging pre-trained vision models to handle small-scale motion datasets, the generalization performance of our method may be limited due to the comparatively smaller size of motion-text data versus image-text data. However, our proposed motion patch, a skeleton-robust representation, aids in constructing large-scale motion datasets from diverse motion capture systems.

7. Conclusion

In this paper, we introduced a novel unified motion representation called "motion patches" and applied the ViT architecture with its pre-trained weights to build motion-language models. Our approach effectively addresses challenges related to limited data scales in 3D human motion data and diverse skeleton structures, characterized by complex spatial-temporal dependencies. As a result, we have made significant advancements in motion recognition, including text-to-motion retrieval and other applications.

References

[1] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *3DV*, 2019. 2

[2] Ayman Ali, Ekkasit Pinyoanuntapong, Pu Wang, and Mohsen Dorodchi. Skeleton-based human action recognition via convolutional neural networks (cnn). *arXiv preprint arXiv:2301.13360*, 2023. 3

[3] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. Teach: Temporal action composition for 3d humans. In *3DV*, 2022. 1, 2

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 3

[5] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *CVPR*, 2023. 2, 3

[6] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. In *ECCV*, 2020. 2

[7] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *CVPR*, 2023. 3

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 3

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 2, 4

[10] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015. 1

[11] Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *ICCV*, 2021. 2

[12] Yuan Gong, Yu-An Chung, and James Glass. Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM T-ASLP*, 2021. 3

[13] Varun Gulshan, Lily H. Peng, Marc Coram, Martin C. Stumpe, Derek J. Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge A Cuadros, Ramasamy Kim, Rajiv Raman, Philip Nelson, Jessica L Mega, and Dale R. Webster. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 2016. 3

[14] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *ACM MM*, 2020. 2, 5

[15] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, 2022. 2, 3, 5, 6, 7, 8

[16] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *ECCV*, 2022. 2

[17] Lisa Anne Hendricks and Aida Nematzadeh. Probing image-language transformers for verb understanding. In *ACL*, 2021. 5

[18] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint arXiv:2311.17117*, 2023. 1

[19] Vladimir Iglovikov and Alexey Shvets. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*, 2018. 3

[20] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*, 2013. 2

[21] Deok-Kyeong Jang, Soomin Park, and Sung-Hee Lee. Motion puzzle: Arbitrary motion style transfer by body part. *ACM TOG*, 2022. 4

[22] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. In *NeurIPS*, 2023. 2, 3

[23] Sai Shashank Kalakonda, Shubh Maheshwari, and Ravi Kiran Sarvadevabhatla. Action-gpt: Leveraging large-scale language models for improved and generalized zero shot action generation. *arXiv preprint arXiv:2211.15603*, 2022. 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[25] Sohaib Laraba, Mohammed Brahimi, Joëlle Tilmanne, and Thierry Dutoit. 3d skeleton-based action recognition by representing motion capture sequences as 2d-rgb images. *Computer Animation and Virtual Worlds*, 2017. 3

[26] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2

[27] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *arXiv preprint arXiv:2304.05684*, 2023. 8

[28] 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 TPAMI*, 2019. 1

[29] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM TOG*, 2015. 7

[30] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 2

[31] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *CVPR*, 2019. 1, 2, 5

[32] Anna Majkowska, Sid Mittal, David F. Steiner, Joshua Jay Reicher, Scott Mayer McKinney, Gavin E Duggan, Krish Eswaran, Po-Hsuan Cameron Chen, Yun Liu, Sreenivasa Raju Kalidindi, Alexander Ding, Greg S Corrado,

Daniel Tse, and Shravya Shetty. Chest radiograph interpretation with deep learning models: Assessment with radiologist-adjudicated reference standards and population-adjusted evaluation. *Radiology*, 2019. 3

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 6

[34] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao. Rethinking cnn models for audio classification. *arXiv preprint arXiv:2007.11154*, 2020. 3

[35] Jae Sung Park, Sheng Shen, Ali Farhadi, Trevor Darrell, Yejin Choi, and Anna Rohrbach. Exposing the limits of video-text models through contrast sets. In *NAACL*, 2022. 5

[36] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *ICCV*, 2021. 1, 2

[37] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *ECCV*, 2022. 1, 2, 6, 8

[38] Mathis Petrovich, Michael J Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *ICCV*, 2023. 1, 2, 3, 4, 5, 6, 8

[39] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 2016. 1, 2, 5

[40] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. Babel: Bodies, action and behavior with english labels. In *CVPR*, 2021. 1, 8

[41] 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 *ICML*, 2021. 1, 2, 3, 5

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 2, 3, 5

[44] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS-W*, 2019. 5

[45] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, 2016. 2

[46] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019. 1, 8

[47] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 2

[48] Mikihiro Tanaka and Kent Fujiwara. Role-aware interaction generation from textual description. In *ICCV*, 2023. 2

[49] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *CVPR*, 2018. 1

[50] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *ECCV*, 2022. 1, 2, 4, 6, 8

[51] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. In *ICLR*, 2023. 2, 3, 5

[52] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, 2014. 1

[53] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. *arXiv preprint arXiv:2311.16498*, 2023. 1

[54] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 1

[55] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *CVPR*, 2019. 2

[56] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bag-of-words models, and what to do about it? In *ICLR*, 2023. 5

[57] 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. In *CVPR*, 2023. 2, 3

[58] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 3, 5

[59] Rui Zhao, Hui Su, and Qiang Ji. Bayesian adversarial human motion synthesis. In *CVPR*, 2020. 2

[60] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *ICCV*, 2023. 1