Advancing Motion with LLMs: Leveraging Large Language Models for Enhanced Text-Conditioned Motion Generation and Retrieval

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
***Computer Science and Engineering** by Research*

by

Sai Shashank Kalakonda
2018111016
sai.shashank@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)
Hyderabad - 500 032, INDIA
February, 2025

International Institute of Information Technology Hyderabad

(Deemed to be University)

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled ***"Advancing Motion with LLMs: Leveraging Large Language Models for Enhanced Text-Conditioned Motion Generation and Retrieval"*** by Sai Shashank Kalakonda, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Ravi Kiran Sarvadevabhatla

"To my family and friends, with deepest gratitude to the Almighty."

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Ravi Kiran Sarvadevabhatla, for his invaluable mentorship and unwavering support throughout the course of this research. His profound knowledge, insightful advice, and dedication to pushing the boundaries of research have been a source of immense inspiration for me. He is the reason behind many of my significant milestones—my first research experience, my first publication, and my first opportunity to travel abroad for academic work. His ability to push me beyond limits I never imagined or believed possible has played a crucial role in shaping the person I have become today. His pursuit of perfection has also instilled in me a deep sense of striving for excellence, which I will carry forward. I am truly fortunate to have had him as my guide.

I am deeply grateful to my family, whose unconditional love, encouragement, and belief in me have always been my strongest pillars of support. Their understanding during my long hours of work and constant motivation has been invaluable. Without their sacrifices and unwavering faith, this journey would not have been possible. A special mention goes to my sister, Sowmya, who was always there for me and supported our family during the times when I was completely absorbed in my work.

A special acknowledgement goes to my research colleague and friend, Shubh Maheshwari. Shubh has been a constant companion throughout the highs and lows of my research journey. Whether it was troubleshooting technical challenges, discussing complex ideas late into the night, or simply offering guidance when I was unsure of my direction, Shubh was always there. His insight and collaboration were instrumental in shaping the trajectory of this work, and I cannot thank him enough for his encouragement and friendship throughout the process.

I also owe a great deal to Debtanu Gupta, who was the first person to guide me into the realm of research. Debtanu not only introduced me to the world of academic inquiry, but he also provided me the opportunity to be a part of my very first publication. His early mentorship and willingness to share his knowledge helped lay the foundation for much of what I have achieved in this work. For this and his continued support, I will always be grateful.

My deepest appreciation goes to my friend, Guru Ravi Shanker, who has been an unwavering source of both moral and technical support throughout my research journey. He has stood by me not just as a friend, but as a mentor and confidant, always there to pull me out of difficult situations whenever I found myself stuck—whether it was in moments of self-doubt, technical challenges, or simply the

mental exhaustion that comes with research. His deep understanding of both the personal and academic challenges I faced meant that he was able to offer precisely the support I needed at any given moment.

Heartfelt thanks to Khadiravana Belagavi and Sai Madhusudan Gunda for their companionship and for keeping me connected to key aspects of my research journey.

A special mention to Sriharshitha, who was my go-to person for sharing queries and discussions during my extended research journey.

I also want to acknowledge my wonderful circle of friends, the "*Cult Boys*." You all have been a source of laughter, joy, and strength throughout this journey. The bond we share has provided much-needed relief from the rigors of research, and the memories we've created together will be cherished forever. Thank you for your constant support and for reminding me that there is life beyond research.

Lastly, I am grateful to my institute, which shaped me into the person I am today. The knowledge, skills, and experiences I gained here have been invaluable in my academic journey. The supportive environment and opportunities for growth provided by the faculty and staff have significantly contributed to my personal and professional development. I appreciate the rich learning experiences and lasting friendships I have formed during my time at this institution.

To all of you, I extend my deepest thanks. Your contributions, in ways big and small, have played an integral role in making this thesis a reality.

Abstract

In the field of artificial intelligence, the generation of human-like motion from natural language descriptions has garnered increasing attention across various research domains. Computer vision focuses on understanding and replicating visual cues for motion, while computer graphics aims to create and edit visually realistic animations. Similarly, multimedia research explores the intersection of data modalities, such as text, motion, and image, to enhance user experiences. Robotics and human-computer interaction are pivotal areas where language-driven motion systems improve the autonomy and responsiveness of machines, facilitating more efficient and meaningful human-robot interactions. Despite its significance, existing approaches still encounter significant difficulties, particularly when generating motions from unseen or novel text descriptions. These models often lack the ability to fully capture intricate, low-level motion nuances that go beyond basic action labels. This limitation arises from the reliance on brief and simplistic textual descriptions, which fail to convey the complex and fine-grained characteristics of human motion, resulting in less diverse and realistic outputs. As a result, the generated motions frequently lack the subtlety and depth required for more dynamic and context-specific applications.

This thesis introduces two key contributions to overcome these limitations and advance text-conditioned human motion generation. First, we present **Action-GPT**, a novel framework aimed at significantly enhancing text-based action generation models by incorporating Large Language Models (LLMs). Traditional motion capture datasets tend to provide action descriptions that are brief and minimalistic, often failing to convey the full range of complexities involved in human movement. Such sparse descriptions limit the ability of models to generate diverse and nuanced motion sequences. Action-GPT leverages LLMs to create richer, more detailed descriptions of actions, capturing finer aspects of movement. By doing so, it improves the alignment between text and motion spaces, enabling models to generate more precise and contextually accurate motion sequences. This framework is designed to work with both stochastic models (e.g., VAE-based) and deterministic models offering flexibility across different types of motion generation architectures. Experimental results demonstrate that Action-GPT not only enhances the quality of synthesized motions—both in terms of realism and diversity—but also excels in zero-shot generation, effectively handling previously unseen text descriptions.

Second, we introduce **MoRAG**, a sophisticated retrieval-augmented generation strategy tailored to enhance the performance of motion diffusion models. MoRAG adopts a multi-part fusion retrieval mechanism that allows for improved generalization of motion retrieval across a wide range of language inputs, addressing the limitations of current retrieval methods that struggle with unseen or atypical de-

scriptions. By incorporating low-level, part-specific motion details into the retrieval process, MoRAG constructs more accurate and varied motion sequences. The retrieval process is refined by prompting LLMs to handle issues like spelling errors, rephrasing, and ambiguous language, ensuring that the retrieved motions are contextually relevant and diverse. These augmented motion samples are then used as additional knowledge within the motion generation pipeline, enhancing the system's ability to generate complex and realistic motions from diverse textual inputs. This retrieval-augmented approach increases both the robustness and generalization capacity of the motion generation models, making them more adaptable to complex and unseen scenarios.

Together, these contributions represent a substantial advancement in text-conditioned human motion generation. By refining both the action description process and the motion retrieval strategy, this work enhances the ability of models to generate diverse, realistic motions from natural language inputs, particularly in zero-shot settings and when handling detailed, complex descriptions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent advancements in computer vision have unlocked a variety of new applications in fields such as virtual reality, human-robot interaction, gaming, and entertainment. One of the emerging challenges in this space is the generation of human-like motion from text descriptions, which has the potential to impact industries ranging from fashion to entertainment and beyond. However, this task is far from straightforward due to the complexity and variability of natural language descriptions, the limited availability of motion capture datasets with text annotations, and the challenge of consistently translating diverse text inputs into accurate motion outputs. In this chapter, we discuss our motivation for delving into this problem domain, outline the specific problem statements, and highlight the associated research challenges. We also provide an overview of the existing literature, noting its limitations, and conclude with a summary of our major contributions to this area.

1.1 Motivation

"Language shapes the way we think, and determines what we can think about." — Benjamin Lee Whorf. This insight emphasizes the fundamental connection between language and human behavior. Humans naturally translate verbal instructions into actions, and conversely, use language to interpret and describe observed behaviors. This seamless integration of language and action is central to human interaction and communication.

In the field of artificial intelligence, the generation of human-like motion from natural language descriptions has attracted significant attention from diverse research domains. These include computer vision [1, 18, 38, 4, 25], which focuses on understanding and replicating visual cues for motion, and computer graphics [9, 3, 57, 40, 6], which aims to create and edit visually realistic and dynamic animations. Similarly, multimedia research [49, 55] explores the intersection of various data modalities, like text, motion and image, to enhance user experiences. Robotics [60] and human-computer interaction [13, 32, 23] are also pivotal areas where language-driven motion systems aim to improve the autonomy and responsiveness of machines, enabling more efficient and meaningful interactions between humans and robots.

The ultimate goal of human motion generation is to create natural, realistic, and diverse motions that can be applied across a wide range of industries. As shown in Figure 1.1, this technology holds significant potential in various fields. Augmented and virtual reality (AR/VR) can greatly benefit from text-conditioned motion, as illustrated in (A), where user actions in virtual environments are simulated in real-time. In video games, motion generation based on textual descriptions enhances the flexibility and creativity of character animations, such as the fighting sequences depicted in (B). In film production, text-driven motion can streamline animation workflows, enabling more efficient creation of complex scenes.

Moreover, digital humans (C) can be animated to perform various tasks in 3D environments, such as sitting or grabbing objects, adding depth to simulations and virtual experiences. Full-body motion sequences, like the example in (D) where, *a man picks up an object with a horrified expression*, showcase the ability to generate highly expressive and context-aware animations from descriptive text. Finally, robotics applications (E) allow humanoid robots to perform human-like actions, such as *exaggerated movements while eating popcorn*, enabling more natural human-robot interactions. These examples illustrate the broad potential of text-driven human motion generation across entertainment, virtual environments, and robotics.

By empowering machines to comprehend and execute actions based on language instructions, these technological advancements pave the way for more intuitive, seamless, and human-like behaviors in both virtual and physical environments. This progress effectively bridges the gap between language, cognition, and action, broadening the scope and impact of artificial intelligence across multiple fields, thus pushing forward the frontiers of human-machine collaboration.

1.2 Problem Statement

The focus of this thesis is to address the limitations of existing text-conditioned human motion generation systems in their ability to produce accurate, diverse, and contextually relevant human motions based on textual descriptions. While these systems can generate motions from text inputs, they often lack precision, resulting in outputs that do not adequately align with the specific details or nuances of the input text. This misalignment leads to motions that are either too generic or inaccurate, which poses significant challenges in scenarios requiring fine-grained control to capture the subtleties of human actions.

In addition to enhancing motion generation, this research also tackles the sub-task of text-to-human motion retrieval. This involves developing methods to retrieve the most relevant motion samples from a dataset in response to a given textual description.

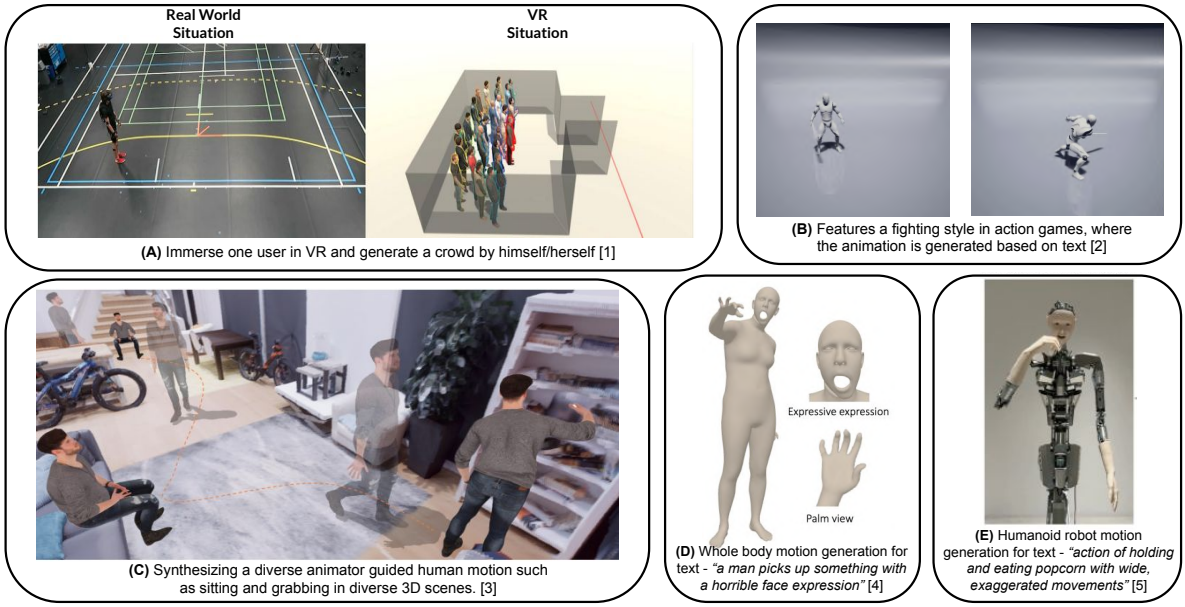


Figure 1.1 Examples of diverse domains where text-conditioned motion generation is beneficial include: (A) A virtual reality situation which is being simulated by a real world user; (B) An action video game featuring fighting animations generated from text descriptions; (C) A digital human performing tasks such as sitting and grabbing in diverse 3D environments; (D) A full-body motion sequence generated for the text - *"a man picks up something with horrible face expression"*. (E) A humanoid robot executing actions based on text, such as - *"action of holding and eating popcorn with wide, exaggerated movements"*.

1.3 Research Challenges

Both of the aforementioned problem statements present a wide variety of research challenges, detailed as follows:

- **Complexity in Human Motion:** Human motion is a complex process made up of various patterns and behaviors influenced by physical, biomechanical, and neurological factors [17, 52]. A key characteristic of human movement is its non-linearity, which means that even minor changes can result in significant differences in how we move. This complexity is further compounded by the articulated nature of human anatomy, where multiple joints and segments work in coordination to produce a wide range of motions. [68]
- **Semantic Coherence:** Generating semantically correct human motion is challenging because models must capture not just the physical action, but the underlying meaning—such as emotions, intentions, and cultural context. Small variations in motion can change the meaning entirely, making it difficult to align the generated motion with both the text and its implied nuances.
- **Temporal Coherence:** Human motion sequences require maintaining temporal consistency across frames to ensure realistic and plausible movements. Achieving this is challenging due to the complexity of human motion, making it difficult to seamlessly integrate information across multiple frames. As a result, ensuring smooth and coherent transitions in generated motion is a key challenge.
- **Text Interpretation Challenges:** Textual descriptions can encode rich information about actions, movement dynamics, and directions with varying levels of clarity. This variability and ambiguity, makes it challenging for motion generation models to accurately interpret and translate these descriptions into coherent, realistic human motions.
- **Stochasticity:** Actions can be executed in various ways, making it important for the model to incorporate stochasticity to reflect this variability. This allows for a more diverse range of generated motions, enhancing realism and adaptability in dynamic scenarios. By accounting for this randomness, the model can better mimic the unpredictability inherent in human behavior.
- **Scarcity in text-annotated datasets:** The vastness of the language space contrasts with the limited availability of text-annotated datasets for human motion. This scarcity restricts models from learning diverse and varied descriptions, making it difficult to generalize well to unseen or complex text inputs. As a result, generating accurate and contextually rich motions becomes a challenge.
- **Zero-shot:** Given the limited availability of datasets, it is crucial for the model to possess zero-shot capabilities. This allows it to effectively handle and generalize to unseen text conditions. By developing this ability, the model can generate appropriate motions even with unfamiliar input descriptions.

1.4 Research Landscape

Human motion generation focuses on producing sequences of human poses in response to a given input signal. Early research in this domain primarily concentrated on *"action-class"* conditioned motion generation, where predefined action categories like walking, running, or sitting were used as inputs. Datasets such as NTU-RGBD-120 [28], HumanAct12 [69], and UESTC [22] provided extensive pose sequences corresponding to a wide variety of action categories, serving as critical resources for advancing research in this area. However, this approach was limited by the finite set of predefined action labels, which restricted the diversity and granularity of the generated motions.

The introduction of AMASS [34], a large and varied motion capture dataset, marked a significant shift in the field. With the availability of text annotations from datasets like BABEL [42] and HumanML3D [18], researchers began exploring text-conditioned motion generation, which allows for a more dynamic and versatile approach. This method opens up new possibilities by enabling models to generate human motion based on detailed and natural language descriptions, far beyond the constraints of fixed action categories, thus leading to richer, more nuanced, and context-specific motion sequences. However, the task of text-to-motion is quite challenging, requiring a deep understanding of both linguistic subtleties and the dynamics of physical motion.

Early research in text-to-motion primarily focused on GAN and GRU-based models [1, 2, 16], which aimed to learn a joint embedding space for language and motion. Subsequently, VAE-based approaches [18, 38, 4] were introduced to generate natural and diverse motion sequences. However, these methods are limited to generating motions based on the texts within their training datasets and struggle to produce results for unseen text inputs, known as the zero-shot setting. To address this limitation, recent works [49, 21] have focused on aligning the motion latent space with the text and image spaces of the pre-trained vision-language model CLIP [44]. It is important to note that while these methods incorporate text as input, they primarily rely on brief descriptions that mostly consist of action class names. Such brief descriptions fail to adequately capture the complexities of motion characteristics. Therefore, there is a significant necessity to improve how these text descriptions are handled, aiming to provide more detailed, low-level motion descriptions. This enhancement would enable the models to perform better and generalize more effectively.

Recent advancements in VQ-VAE and diffusion models within the text and image domains have sparked interest in applying similar techniques to text-to-motion generation. Rather than simply predicting noise, these models focus on generating motion samples, which allows for the application of geometric losses as training constraints [62, 63, 11]. However, these models often struggle with unusual or atypical inputs. To address this limitation, motion generation approaches have begun to adopt retrieval augmentation techniques, incorporating additional knowledge through retrieval strategies within the training pipeline [64]. This integration not only enhances the models' ability to handle a wider range of inputs but also improves the overall robustness and versatility of motion generation systems.

Lately, the task of text-to-motion retrieval has garnered significant attention, focusing on retrieving relevant motion sequences from databases based on given text descriptions. Drawing inspiration from

the CLIP model [44], these motion retrieval methods employ contrastive learning techniques [39, 7] to enhance their performance. By assessing the similarities between text inputs and motion sequences, contrastive learning boosts the accuracy of retrieving motions that align with the semantic meaning of the provided descriptions. As a result, text-to-motion retrieval shows promise in improving the effectiveness and adaptability of motion generation technologies.

However, the current retrieval methods struggle to generalize effectively and fall short in delivering diverse motion sequences due to the limited availability of text-annotated motion datasets. This limitation hinders the training of Retrieval-Augmented Generation (RAG) architectures in the motion generation domain. Therefore, it is crucial to develop a more effective retrieval strategy that can construct relevant and diverse motion sequences while also being capable of handling unseen text descriptions to some degree.

1.5 Thesis Contributions

As part of this thesis, we present solutions aimed at enhancing the performance of existing text-conditioned motion generation models through the integration of large language models (LLMs) and retrieval-augmented strategies. Below, we outline our key contributions associated with these efforts.

1. **Action-GPT: Leveraging Large-scale Language Models for Improved and Generalized Action Generation:** In this work, we propose a plug-and-play framework for incorporating Large Language Models (LLMs) into text conditioned motion generation methods, where our key contributions are:

   a. Usage of LLM-generated detailed body movement descriptions instead of existing concise text annotations enhances the motion generation pipeline.

   b. Novel framework which is compatible with deterministic and stochastic text-to-motion models.

   c. We present a meticulously designed prompt function that facilitates the generation of meaningful descriptions for specific action phrases, along with an assessment of its suitability.

2. **MoRAG - Multi-Fusion Retrieval Augmented Generation for Human Motion:** In this work, we propose a novel multi-part fusion based retrieval-augmented generation strategy for text-based human motion generation, where our key contributions are:

   a. We adapt part-wise motion retrieval approach that utilizes generative prompts to construct motion sequences that align with the provided text description.

   b. Motion sequences constructed using our retrieval strategy exhibit enhanced generalization and diversity, which further improves the performance of motion generation models when conditioned.

1.6 Organization of the Thesis

In this chapter, we have outlined our target problem domain, our motivation for pursuing it, and discussed the related research challenges, along with a brief overview of the existing literature and its limitations.

The remainder of this thesis is organized as follows: Chapter 2 presents the key methods that underpin our work, providing essential background information. Chapter 3 introduces Action-GPT, a framework designed to incorporate Large Language Models (LLMs) into text-conditioned motion generation models. This chapter provides a comprehensive description of our methodology, along with the experimental setups and results. Chapter 4 focuses on MoRAG, a novel multi-part fusion-based retrieval-augmented generation strategy aimed at enhancing the semantic coherence and diversity of existing diffusion-based motion generation models. It also presents an augmented retrieval strategy that surpasses current motion retrieval methods. This chapter concludes with a detailed account of our approach and its experimental results. Finally, Chapter 5 summarizes our key contributions and discusses potential future research directions in this domain.

Chapter 2

Background

In this chapter, we review the foundational literature relevant to this thesis. Specifically, we explore motion data representation and overview of large language models (LLMs) and prompting mechanism.

2.1 Motion Data Representation

Human motion, when represented through poses, is captured as a series of poses, each of which defines the structure of the human body's kinematic chain at a particular moment in time. More specifically, an action sequence $\mathcal{X}$, spans over T timesteps. This sequence is denoted as $\mathcal{X} = \{[X_1, X_2, \dots X_T]\}$, $1 \leqslant t \leqslant T$ indicates the time steps. Each pose, $[X]_t$, captures the joint configuration at time t within a global coordinate frame. The time steps t can vary depending on the specific activity being performed. Each pose $[X]_t$ can either be represented through 3D joint positions or 3D rotations, which are combined using forward kinematics. However, the latest methods adapt 3D rotation based representation as it avoids problems such as inconsistent bone lengths.

2.1.1 3D rotation based representation

In this approach, each joint's rotation is represented in 3D relative to a fixed rest pose. These rotations are then processed through a forward kinematics module to compute the corresponding 3D joint positions. Current methods, such as [35], do not regress directly on the predicted rotations. Instead, they apply forward kinematics to generate 3D joint positions and perform regression on these generated positions. There are various ways to represent rotations for regression tasks, each with its own strengths and limitations. The following section explores some of the most commonly used rotation representations in this context.

Rotation Matrix: The simplest way to represent a rotation is through a rotation matrix, which uniquely defines any rotation. For human pose representation in 3D space, a 3D rotation is described by a 3×3 rotation matrix A, a special orthogonal matrix with a determinant always equal to one.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{2.1}$$

where

$$A^T A = A A^T = 1, \tag{2.2}$$

$$det(A) = 1 \tag{2.3}$$

6D rotation representation: When using 3×3 rotation matrices for regression, it's necessary to enforce the orthogonality constraint. This can be achieved by applying the Gram-Schmidt orthogonalization process, which constructs an orthonormal basis from two vectors. During this process, one row or column of the predicted noisy matrix is discarded. Since three parameters of the rotation matrix become redundant, we can predict only six parameters and reconstruct the full rotation matrix. The key idea is that, because a rotation matrix is a 3×3 orthonormal matrix, we can omit the third column and later recover it by computing the cross product of the first two columns. Thus, by predicting only two columns of the matrix and applying the Gram-Schmidt orthogonalization, we arrive at a continuous 6D representation for rotation.

Let $\vec{a_1}$ and $\vec{a_2}$ be the two vectors used to derive the 3D rotation matrix. The rotation matrix 'A' can be constructed as follows:

$$\vec{c_1} = \frac{\vec{a_1}}{||\vec{a_1}||} \tag{2.4}$$

$$\vec{c_2} = \frac{\vec{b_2}}{||\vec{b_2}||}, \vec{b_2} = \vec{a_2} - (\vec{c_1}.\vec{a_2})\vec{c_1} \tag{2.5}$$

$$A = (c_1, c_2, c_1 \times c_2) \tag{2.6}$$

Many recent methods [33, 36] employ a 6D rotation representation for regression. Initially, the human motion synthesis model generates local human pose sequences using this 6D representation. These sequences are then converted into a 3×3 rotation matrix representation. Subsequently, this representation is processed through a forward kinematics module to derive the human pose sequences in the form of 3D joint positions.

Forward Kinematics: This computes 3-D positions of each joint from the 3D rotation based pose representation. The human pose sequence need to be represented using 3×3 rotation matrices, i.e. $\widetilde{\mathcal{X}}_l = \{[X_l]_t\}$ where $X_l \in \mathbb{R}^{J \times 3 \times 3}$. The forward kinematic function takes $\widetilde{\mathcal{X}}_l$, vertices of rest pose $\widetilde{\mathcal{R}_l}$ (which is constant across the sequence) and the kinematic tree as input. It applies transformation F_{fkt} which outputs 3-D positions of joints $\widetilde{\mathcal{X}}_e = \{[X_e]_t\}$, where pose instance $X_e \in \mathbb{R}^{J \times 3}$. The transformation F_{fkt} is given as:

Figure 2.1 SMPL Representation: This figure demonstrates SMPL base human pose representation. This is essentially represented by body pose (3D joint rotation of 23 joints) and body shape (beta parameters). (Image taken from Loper Matthew, SMPL: A Skinned Multi-Person Linear Model, SIGGRAPH Asia [30])

$$F_{fkt}(X_l^{(c)}, \widetilde{\mathcal{R}}) = \begin{cases} [0,0,0] & \text{if root joint.} \\ X_l^{(i)^{(c)}} \cdot (\widetilde{\mathcal{R}}^c - \widetilde{\mathcal{R}}^p) & \\ +F_{fkt}(X_l^{(p)}, \widetilde{\mathcal{R}}) & \text{otherwise.} \end{cases}$$

where $X_l^{(c)}$ and $X_l^{(p)}$ indicate 3×3 rotation matrix of child and parent joints respectively in the kinematic tree of pose instance X_l. $\widetilde{\mathcal{R}}^c$ and $\widetilde{\mathcal{R}}^p$ indicate 3D joint position of child and parent joints respectively in the kinematic tree of rest pose $\widetilde{\mathcal{R}}$. Finally, the 3-D joint positions of pose instance $X_e \in \mathbb{R}^{J\times 3}$ is given as, $X_e = [f_{fkt}(X_l^{(1)}, \widetilde{\mathcal{R}}), f_{fkt}(X_l^{(2)}, \widetilde{\mathcal{R}}), ..., f_{fkt}(X_l^{(J)}, \widetilde{\mathcal{R}})]$.

2.1.2 SMPL representation

The Skinned Multi-Person Linear Model (SMPL) is a learned model of human body shape and pose that integrates seamlessly with existing 3D mesh rendering tools. SMPL separates human body pose into two components: identity-specific shape and pose-dependent shape. It then employs a vertex-based skinning method to generate a 3D mesh representation of the human body. This 3D mesh consists of 6890 vertices and 23 joints. A key innovation of SMPL is that the pose-dependent blend shapes are

formulated as a linear function of pose rotation matrices, which define human body poses. This straightforward approach allows the model to be trained on a large dataset of aligned 3D meshes of individuals in various poses. By providing pose rotations (pose parameters) and body shape (beta parameters), the model can generate a corresponding SMPL-based 3D human mesh. Additionally, modifying the beta parameter allows for the creation of 3D meshes with different body shapes.

2.1.3 263 dimension representation

According to 263-dimensional motion representation, as described in [18], each human pose C_t^i is represented by $(\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, \mathbf{j}^p, \mathbf{j}^v, \mathbf{j}^r, \mathbf{c}^f)$, where $\dot{r}^a \in \mathbb{R}$ is root angular velocity along Y-axis; $\dot{r}^x \in \mathbb{R}$, $\dot{r}^z \in \mathbb{R}$ are global root velocities in X-Z plane; $\dot{r}^y \in \mathbb{R}$ is root height; $\mathbf{j}^p \in \mathbb{R}^{3\times n(J)}$, $\mathbf{j}^v \in \mathbb{R}^{3\times n(J)}$, $\mathbf{j}^r \in \mathbb{R}^{6\times n(J)}$ are the local pose positions, velocity and rotation respectively. $\mathbf{c}^f \in \mathbb{R}^4$ is the foot contact features calculated by the heel and toe joint velocity. $n(J)$ is the number of joints and T_i represents the number of timesteps for motion C^i.

2.2 Prompting Large Language Models

2.2.1 Large Language Models

Large Language Models (LLMs) are advanced types of artificial intelligence designed to process and generate human language. These models are based on deep learning, particularly using transformer architectures [54], which allow them to capture patterns in vast amounts of text data. LLMs are capable of understanding context, reasoning, and even generating text that closely mimics human writing. Examples of LLMs include OpenAI's GPT series, Google's BERT, and Meta's LLaMA.

LLMs have gained significant attention and importance in recent years due to advancements in several areas:

1. **Transformer Architecture:** The transformer architecture introduced in the work "Attention Is All You Need" [54], changed the way models process text. Unlike traditional RNNs (Recurrent Neural Networks) and CNNs (Convolutional Neural Networks), transformers handle sequences in parallel, which allows them to process long-range dependencies more effectively. This leads to better performance in understanding and generating text.

2. **Massive Data Availability:** The internet provides a vast resource of textual data, including books, articles, websites, and social media content. This data has been used to train LLMs on diverse linguistic styles, knowledge domains, and contexts, making them highly adaptable across various tasks.

3. **Increased Computational Power:** The rise of high-performance computing resources, such as GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units), has enabled the training

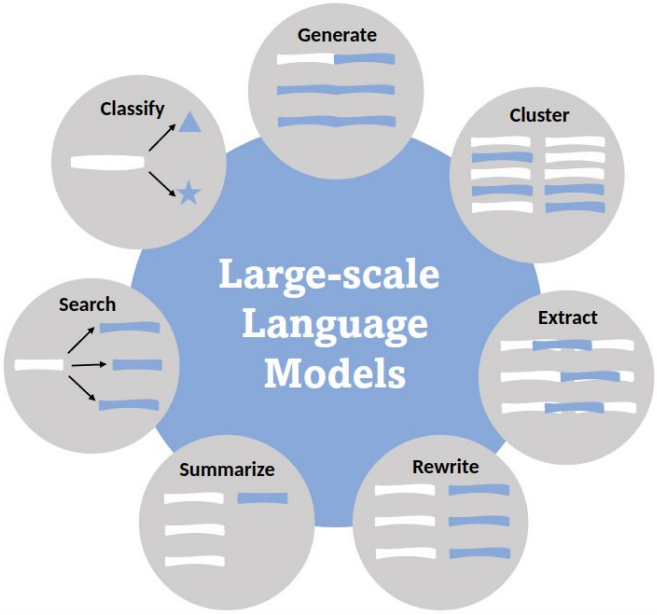


Figure 2.2 Illustration of various use cases of Large Language Models (LLMs), including text generation, summarization, classification, extraction etc. The versatility of LLMs allows them to excel in tasks ranging from creative writing to complex problem-solving across multiple domains. (Image Credits : Cohere)

of massive models with billions of parameters. This scale allows LLMs to better understand complex language patterns and generate more coherent and accurate outputs.

2.2.2 The Versatile Capabilities of Large Language Models (LLMs)

As illustrated in Figure 2.2, LLMs stand out because of their ability to generalize and excel at multiple tasks, often without needing task-specific training. Some of their most significant advantages include:

1. **Natural Language Understanding (NLU):** LLMs excel at interpreting and comprehending human language, recognizing patterns, and extracting meaning from text. This ability allows them to understand complex questions and respond in a manner that is contextually appropriate.

2. **Text Generation:** LLMs can produce coherent and contextually relevant text based on a given prompt or input. Whether it's writing essays, summarizing articles, generating reports, or creating dialogue, LLMs can generate high-quality content across diverse topics.

3. **Question Answering:** LLMs can provide answers to factual, technical, and open-ended questions by leveraging their trained knowledge across a wide variety of domains. This makes them useful for research, customer support, and general information retrieval.

4. **Generalization Across Tasks:** Unlike traditional models, which often require specific training for each task, LLMs are versatile. They can perform a wide range of functions such as question answering, language translation, text summarization, and more—often with just a few examples or prompts guiding them.

2.2.3 Prompting

Prompting is the method of creating specific inputs or queries that direct the LLM to generate relevant and accurate outputs. This process is particularly important for large models like GPT, as it helps the model grasp the type of response that is expected. By carefully crafting the input to utilize the LLM's language comprehension, you can obtain the desired output. This method is especially vital given the complexity and sensitivity of LLMs to variations in input. For example, when requesting text generation, the prompt may include detailed instructions, such as asking for a description, summary, or a particular type of content.

In the context of **text-to-motion generation**, prompting plays a key role in shaping the kind of motion description or action sequences you need from the model. The more detailed and specific the prompt is, the better the model can translate that input into coherent and semantically relevant output.

Chapter 3

Action-GPT: Leveraging Large-scale Language Models for Enhanced Human Motion Generation

3.1 Introduction

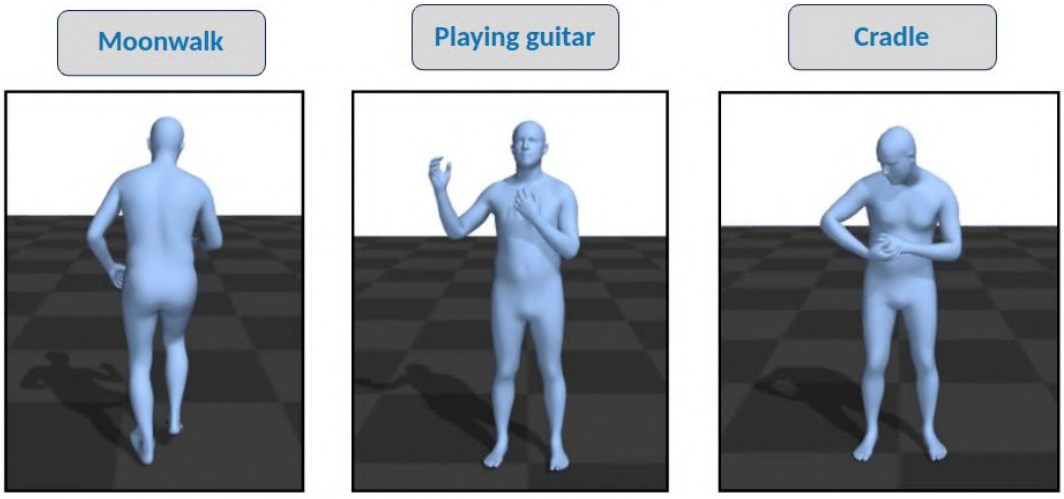


Figure 3.1 BABEL Dataset: Illustration of motion samples in the BABEL dataset [42]. The text annotations in BABEL often provide minimal descriptions of actions, lacking detailed information about specific body movements.

Human motion generation plays a critical role in a wide range of applications, from entertainment industries such as gaming and film production to virtual reality environments and robotics. This technology is integral to creating lifelike animations and interactions, and has the potential to revolutionize how virtual characters and robots perform tasks and engage with their environments. Historically, significant progress has been made in category-conditioned human motion generation, where motion is generated based on predefined action categories. Notable works like ACTOR [37] and models such as

MUGL [33] and DSAG [20] have enabled large-scale motion generation. However, these methods are limited in scope, as they rely on a finite set of action categories, restricting their generalizability and flexibility in diverse, real-world scenarios.

In response to these limitations, more recent approaches have shifted focus towards text-conditioned motion generation, which aligns motion generation with textual descriptions by constraining motion and language representations in a jointly optimized latent space [38, 49, 4]. These methods offer greater flexibility by enabling motion generation based on diverse and more descriptive inputs. However, current large-scale text-annotated motion capture datasets, such as BABEL [42], often contain minimal and overly simplistic textual descriptions that lack the detail necessary to fully describe the intricacies of body movements (Fig. 3.1). This absence of detailed annotations poses a challenge for motion generation models, as the lack of precise and informative text input hinders the generation of accurate and contextually relevant motions. Addressing this gap is essential to improving the quality and applicability of text-conditioned motion generation systems.

The recent development of large-scale language models (LLMs) [47, 8] has triggered a paradigm shift in the field of Natural Language Processing (NLP). These models, pre-trained on enormous amounts of text [29], have demonstrated impressive generalization capabilities for challenging zero-shot setting tasks such as text generation [59]. This exciting advance has also driven progress for various applications in computer vision [56, 41], including the related task of pose-based human action recognition [56].

The appeal of LLM models lies in their ability to generate task-relevant text when provided a so-called *prompt* - a small piece of text - as input. Motivated by this observation and the advances mentioned above, we introduce Action-GPT, an approach that utilizes the generative power of LLMs to improve the quality and generalization capabilities of action generative models. In particular, we demonstrate that our plug-and-play approach can be used to advance existing state-of-the-art motion generation architectures in a practical manner.

Our contributions are summarized below:

- To the best of our knowledge, we are the first to incorporate Large Language Models (LLMs) for text-conditioned motion generation.
- We introduce a carefully crafted prompt function that enables the generation of meaningful descriptions for a given action phrase.
- We introduce Action-GPT, a generic plug-and-play framework which is compatible with stochastic (e.g. VAE-based [4, 38]) and deterministic (e.g. MotionCLIP [49]) text-to-motion models. In addition, our framework enables multiple generated text descriptions to be utilized for action generation.
- Via qualitative and quantitative experiments, we demonstrate (i) noticeable improvement in the quality of synthesized motions, (ii) benefits of utilizing multiple LLM-generated descriptions, (iii) suitability of the prompt function, and (iv) zero-shot generation capabilities of the proposed approach.

Code, pretrained models and sample videos are available at https://actiongpt.github.io.

3.2 Related Works

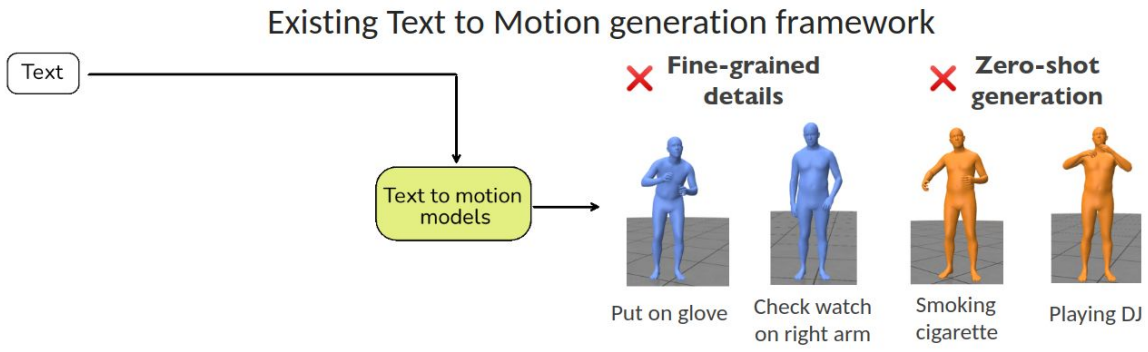


Figure 3.2 Existing Text-to-Motion Frameworks : In the existing motion generation frameworks, textual descriptions are directly passed to the motion generation model, leading to suboptimal results. The lack of detailed language modeling often results in misalignment between the text and the generated motion, producing outputs that are either too generic or fail to capture the full complexity and nuances of the described actions.

Early research on text-conditioned human motion generation focused on encoder-decoder models that employed a multimodal joint embedding space to represent both text and motion domains [1, 2]. Text2Action [1] introduced a GAN-based generative model using an RNN-based text encoder and an action decoder. JL2P [2] emphasized learning a joint embedding space for language and pose, targeting motion reconstruction through the integration of text and motion embeddings. Ghosh et al. [16] improved this approach by developing a hierarchical two-stream model that learned separate motion representations for the lower and upper body.

To improve generalizability in text-to-motion models, MotionCLIP [49] incorporated image embeddings of poses into the training process alongside text embeddings, both of which were generated using CLIP [44]. TEMOS [38] proposed a transformer-based VAE approach that enhanced the diversity of generated motion sequences by encoding both text and motion. Building on TEMOS, TEACH [4] introduced an additional past motion encoder, enabling the generation of long motion sequences by utilizing both text descriptions and prior motion data.

While state-of-the-art (SOTA) text-to-motion models have made significant progress, they still face notable challenges. These models typically map short action phrases to motion sequences within a latent space, but as mentioned earlier, such phrases often lack the detailed specification of limb movements or the sequencing of sub-actions (Fig. 3.2). This omission results in limited generalizability and scalability, as the models struggle to generate motions that accurately reflect complex or nuanced actions. Moreover,

these models often fail to capture fine-grained motion details, which are crucial for producing realistic and contextually appropriate motions. The lack of specificity in textual input also hinders the models' ability to perform well in zero-shot generation, where the task requires producing unseen actions based solely on text descriptions. Consequently, these issues restrict the overall performance and applicability of existing text-to-motion models in more diverse and real-world scenarios.

3.3 Action-GPT

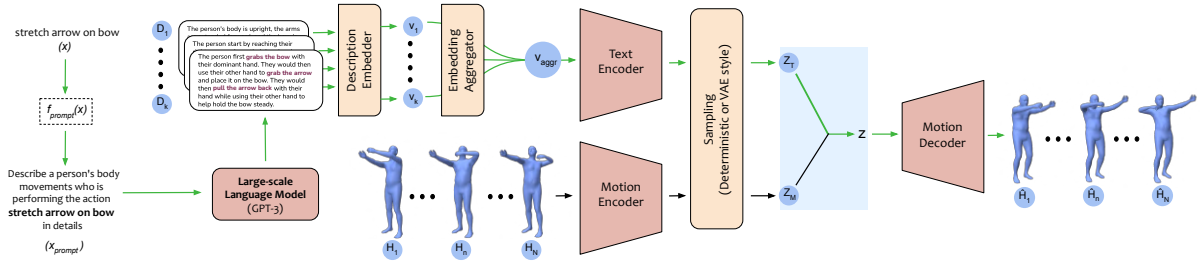


Figure 3.3 Action-GPT Overview: Given an action phrase (x), we first create a suitable prompt using an engineered prompt function $f_{\text{prompt}}(x)$. The result (x_{prompt}) is passed to a large-scale language model (GPT-3) to obtain multiple action descriptions (D_i) containing fine-grained body movement details. The corresponding deep text representations v_i are obtained using Description Embedder. The aggregated version of these embeddings v_{aggr} is processed by the Text Encoder. During training, the action pose sequence $H_1, ..., H_N$ is processed by a Motion Encoder. The encoders are associated with a deterministic sampler (autoencoder) [49] or a VAE style generative model [38, 4]. During training (shown with black), the latent text embedding Z_T and the latent motion embedding Z_M are aligned. During inference (shown in green), the sampled text embedding is provided to the Motion Decoder, which outputs the generated action sequence $\widehat{H}$.

Our objective is to generate an actor performing the motion conditioned on the given action phrase. The input action phrase is a natural language text that gives a high-level description of the action. It is denoted as a sequence of words $x = [w_1, w_2, ..., w_M]$. The action is represented as a sequence of human poses $H = \{H_1, \ldots, H_n, \ldots, H_N\}$ where N represents the number of timesteps. The human pose $H_n \in \mathcal{R}^{|J|\times 6}$, where J is the number of joints, is the parametric SMPL [30] representation wh ich encodes the global trajectory and the parent relative joint rotation using the 6D [67] rotation representation.

Our proposed framework Action-GPT (Fig. 3.3) can be incorporated in an autoencoder [49] or a Variational Auto Encoder [38, 4] based text-conditioned motion generation model. These motion generation

models aim to generate a motion sequence conditioned on the text input by learning a joint latent space between the text and motion modalities. The key components in these models are Text Encoder $\mathcal{T}_{enc}$, Motion Encoder $\mathcal{M}_{enc}$ and Motion Decoder $\mathcal{M}_{dec}$. The two text and motion encoders encode the text sequence and motion sequence to text Z_T and motion Z_M latent embeddings of the same dimension, respectively. In the case of autoencoders, the latent embeddings are obtained in a deterministic fashion, whereas in Variational Auto Encoders, the latent embeddings are sampled from the Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, where (μ, Σ) are the outputs of the encoder. The motion decoder, on the other hand, uses the latent embedding Z as input to generate a sequence of motion poses $\widehat{H} = \{\widehat{H}_1, \ldots, \widehat{H}_n, \ldots, \widehat{H}_N\}$

Fig. 3.3 provides an overview of our approach to incorporate LLM (GPT-3 in our case) into the text-conditioned motion generation models. In contrast to training directly using the action phrase x from the dataset, our framework uses carefully crafted GPT-3 generated text descriptions D_i, which provide low-level details about the movement of individual body parts. The proposed framework consists of three steps (1) Constructing a prompt function f_{prompt}, (2) Aggregating multiple GPT-3 generated text descriptions D_i, and finally, (3) utilizing the GPT-3 generated text descriptions D_i in T2M models.

3.3.1 Prompt strategy

For a given action phrase x, we generate low-level body movement details using GPT-3 [8]. GPT-3 is an autoregressive transformer model which generates human-like textual descriptions relevant to the small amount of input text provided. However, directly providing the action phrase as input to GPT-3 fails to output text containing the desired detail body movement information and leads to unrealistic motion generations (see Fig. 3.4). This necessitates the need for a suitable prompt function [29]. After multiple empirical trials, we determine the following prompting function f_{prompt}: `Describe a person's body movements who is performing the action [x] in detail.` Specifically, adding `Describe a person's` to the prompt restricts the description from generic information to character movement. The phrase `body movements` forces GPT-3 to explain the motion of individual body parts. Lastly, `in detail` forces the descriptions to provide low-level details. Fig. 3.5 showcases the importance of each component of our prompt function. We provide GPT-3 generated text description (D) and corresponding generated action sequence $\widehat{H}$ for the action phrase `act like a dog` along with the observations in the rightmost column.

3.3.2 Aggregating multiple descriptions

Given an action prompt x_{prompt}, GPT-3 is capable of generating multiple textual descriptions $D_1, \ldots, D_k$ describing the action-specific information. The randomly generated k descriptions contain common and description-specific text segments, which enhance the overall richness of action description (see Fig. 3.6). Therefore, we utilize multiple descriptions as part of the text-processing pipeline. The GPT-3 generated k text descriptions $D_1, \ldots, D_k$ are passed through a Description Embedder D_{emb} to obtain corresponding description embeddings $v_1, \ldots, v_k$. These k description embeddings are ag-

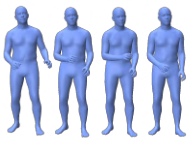
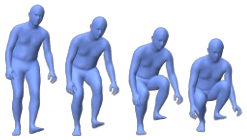
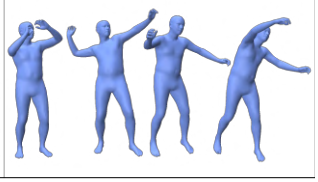
Action phrase(x)	GPT-3(x) output	Generated sequence	GPT-3(x_{prompt}) output	Generated sequence
Duck down	Duck down is an action typically used to avoid something coming towards you such as a ball or someone trying to hit you.		The person's body movements who is performing the action duck down would be to lower their body down closer to the ground, while bending their knees.	
Javelin throw	The javelin throw is an athletic event where the goal is to throw a javelin as far as possible. The javelin is a spear-like object that is thrown with the arm and hand.		The person stands with their feet shoulder-width apart and toes pointing forward. They grip the javelin with their dominant hand in the middle of the shaft and their other hand near the point. They bend their elbow and raise their arm back behind their head. As they throw, they extend their arm, release the javelin, and follow through with their arm and body.	

Figure 3.4 This figure highlights the importance of the prompt function. Observe that directly feeding the action phrase text (x) to GPT-3 results in poor-quality generations. In contrast, the fine-grained body movement details in the prompt-based text enable higher fidelity generations (last column). Note that the coloured text descriptions correspond to different body movement details.

Prompt Function (f_{prompt})	GPT-3 generated text description (D)	Generated Action Sequence ($\hat{H}$)	Observations
Describe the action [x]	The action of acting like a dog would involve someone imitating the behavior of a dog. This could involve barking, wagging their tail, playing fetch, or any other number of dog-like behaviors.		Generated text description contains generic information of act like a dog without any details of body movements, resulting in an unrealistic action generation.
Describe a person performing the action [x]	A person who is acting like a dog might crawl on the ground, bark, and wag their tail.		Generated text description contains high-level body movements information, resulting in an action generation which just captures minimal high-level details.
Describe a person's body movements who is performing the action [x]	The person would be down on all fours, with their hands and feet touching the ground. Their back would be arched and their head would be held high. They would be moving their legs and arms in a coordinated way, similar to how a dog would move		Generated text description contains required low-level body movements information, resulting in a realistic action generation, but suffers in capturing fine-grained details.
Describe a person's body movements who is performing the action [x] in detail	The person would be on all fours with their palms and feet flat on the ground. Their legs would alternate moving forward and their hips would sway from side to side. Their arms would move in rhythm with their legs. Their head would be up and their gaze would be forward.		Generated text description contains required low-level fine-grained body movements information, resulting in a realistic action generation. (Additional details, legs would alternate moving forward enabled in better generation)

Figure 3.5 The table showcases the descriptions generated by GPT-3 (D), generated action sequences ($\widehat{H}$) for the action phrase (x =) *act like a dog* using different prompt strategies along with the observations (right most column). Notice that our prompt function (bottom row) generates the highest amount of required body movement descriptions, generating the most realistic action sequence. Note that the coloured text descriptions correspond to the body movement details.

gregated into a single embedding v_{aggr} using an Embedding Aggregator E_{aggr}. We consider average operation as our Embedding Aggregator unless stated otherwise.

3.3.3 Utilizing GPT-3 generated text descriptions in T2M models

The text encoder $\mathcal{T}_{enc}$ inputs the aggregated embedding v_{aggr}, and the outputs are sampled to generate text latent embeddings Z_T. In a similar fashion, motion encoder $\mathcal{M}_{enc}$ inputs the sequence of motion poses $H = \{H_1, \ldots, H_N\}$, where $H_n \in \mathcal{R}^{|J|\times 6}$ and samples motion latent embeddings Z_M. In a deterministic approach [49], the latent embeddings are generated directly as outputs of the encoder, whereas in a VAE-based approach [38, 4], the encoders generate distribution parameters μ and Σ. The latent embeddings are then sampled from the Gaussian distributions $N(\mu, \Sigma)$. On the other hand, the generated text and motion embeddings are provided to the motion decoder, which generates the human motion sequence $\widehat{H} = \{\widehat{H}_1, \ldots, \widehat{H}_N\}$, where $\widehat{H}_i \in \mathcal{R}^{|J|\times 6}$ which is passed through forward kinematics to generate corresponding 3D mesh sequence.

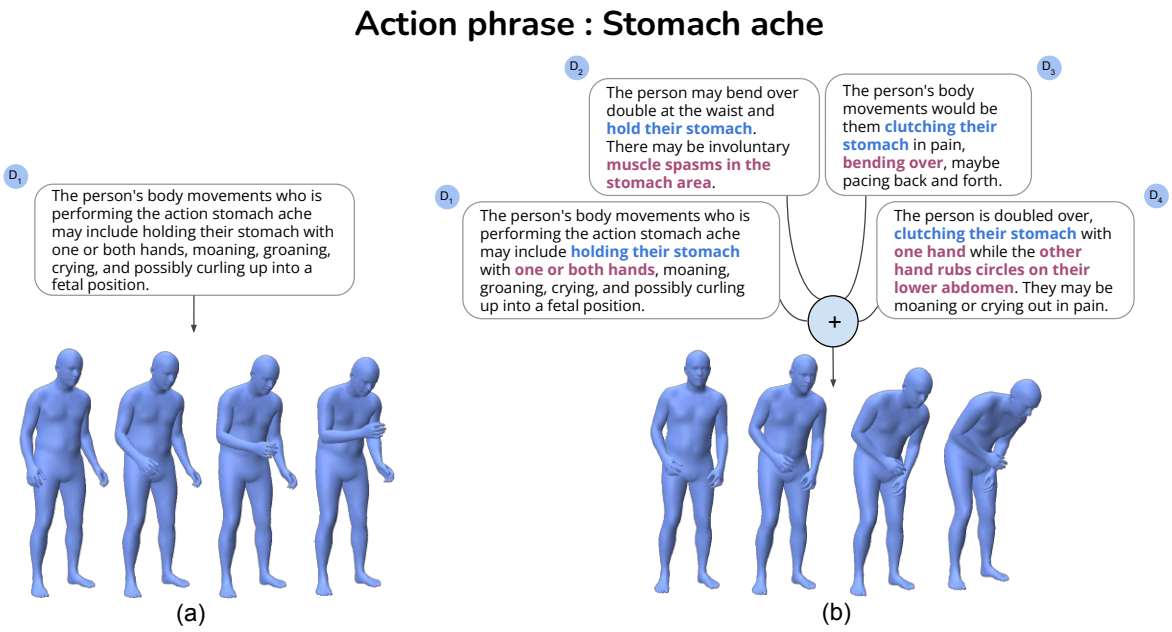


Figure 3.6 This figure highlights the importance of using multiple GPT-3 generated descriptions $(D_1, ..., D_k)$, $k = 4$ for each action phrase in Action-GPT framework for TEACH [4]. Notice the visibly improved generation quality when multiple prompted descriptions are used (right column). Body movement text common across descriptions is highlighted in blue. Movements unique to each description are highlighted in pink.

3.4 Experiments

BABEL [42] is a large dataset with language labels describing the actions being performed in motion capture sequences. It contains about 43 hours of mocap sequences, comprising over 65k textual

labels which belong to over 250 unique action categories. We primarily focus our results on BABEL, considering its vast and diverse set of motion sequences assigned to short text sequences, which contain an average of 3-4 words. The action phrases of the BABEL dataset are to the point and precise about the action information without any additional details about the actor.

3.4.1 Models

We demonstrate our framework on state-of-the-art text conditioned motion generation models - MotionCLIP [49], TEMOS [42] and TEACH [4], all trained on BABEL [42]. We name their LLM (i.e. GPT here) based variants as Action-GPT-[model].

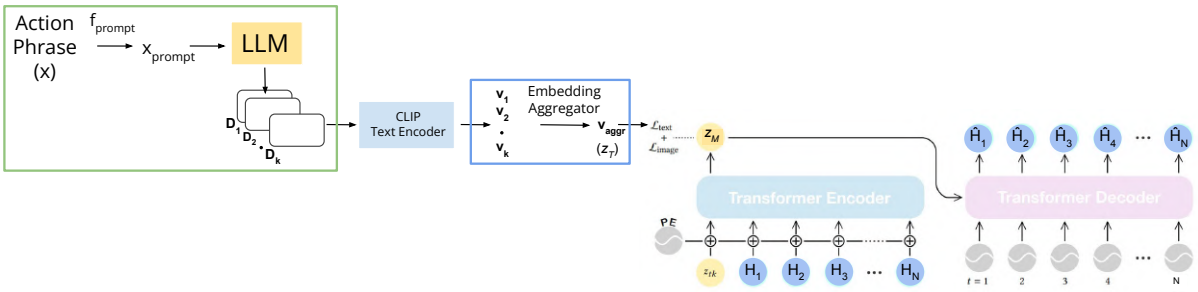


Figure 3.7 Action-GPT-MotionCLIP Overview: We extend MotionCLIP [49] by incorporating LLM (i.e. GPT-3). The box highlighted in green showcases the generation of k text descriptions D_i as the output of LLM on input x_{prompt}, which is constructed using the prompt function f_{prompt} and action phrase x. The box highlighted in blue showcases the aggregation of description embeddings v_i, outputs of the CLIP text encoder. The aggregated embedding v_{aggr} is considered as the latent text embedding Z_T, on which the text loss $\mathcal{L}_{text}$ is computed. All the other components apart from the highlighted boxes represent the original architecture of MotionCLIP [49].

Action-GPT-MotionCLIP: In MotionCLIP [49], for a given action phrase x, the CLIP text embedding of the phrase x is considered as its corresponding latent text embedding Z_T, whereas in our Action-GPT framework the aggregated vector embedding $v_{aggr} \in R^c$, where c is the CLIP text embedding dimesnion, constructed for the action phrase x is considered as its latent text embedding Z_T. In detail, we first construct x_{prompt} using the action phrase x and prompt function f_{prompt}. The x_{prompt} is then input to LLM (i.e. GPT-3) to generate k textual descriptions D_i. Using the CLIP text encoder, we then construct k corresponding CLIP text embeddings v_i. These k CLIP text embeddings v_i are then aggregated into a single embedding v_{aggr} using an Embedding aggregator (average operation here). The constructed v_{aggr} is the corresponding latent text embedding Z_T for the action phrase x. (see Fig. 3.7)

We trained this model using the same training configurations as mentioned in MotionCLIP [49] and provided the results on the test split. We generated the metrics for the baseline MotionCLIP [49] using the pre-trained model provided.

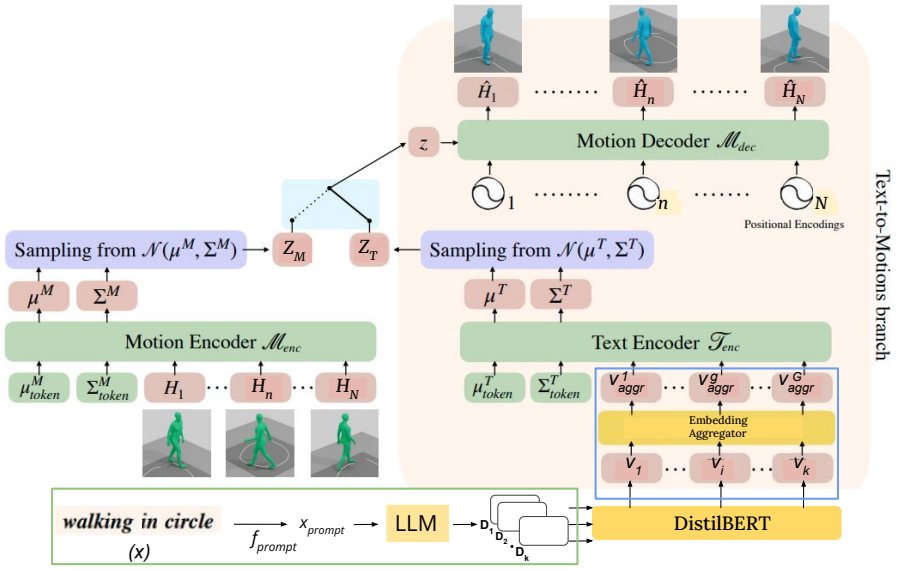


Figure 3.8 Action-GPT-TEMOS Overview: We extend TEMOS [38] by incorporating LLM (i.e. GPT-3). The box highlighted in green showcases the generation of k text descriptions D_i using LLM on input x_{prompt}, which is constructed using the prompt function f_{prompt} and action phrase x. The k text descriptions D_i are input to DistilBERT to generate corresponding description embeddings v_i. The box highlighted in blue showcases the aggregation of description embeddings v_i to $v_{aggr}^{1:G}$ using an embedding aggregator. All the other components apart from the highlighted boxes represent the original architecture of TEMOS [38].

Action-GPT-TEMOS: Instead of providing the action phrase x, directly to DistilBERT [45] as that in TEMOS [38], we first construct x_{prompt} and generate k textual descriptions D_i using LLM (i.e. GPT-3). These k textual descriptions are then input to DistilBERT [45] to obtain k corresponding description embeddings $v_i \in R^{n_i \times e}$, where n_i is the number of words in description D_i and e is the DistilBERT embedding dimension. The k description embeddings are aggregated to a single embedding $v_{aggr} \in R^{G \times e}$, where $G = max(n_i)$ using the average operation. This aggregated embedding v_{aggr} is then used as input to Text Encoder along with the text distribution tokens μ_{token}^T and $\sum_{token}^T$. The later training and inference process is carried out the same as that of in TEMOS [38]. (see Fig. 3.8)

We trained this model on a 4 GPU setup with a batch size of 4 on each GPU, keeping rest all the parameters same as provided in TEMOS [38]. We trained the baseline model of TEMOS [38] and its

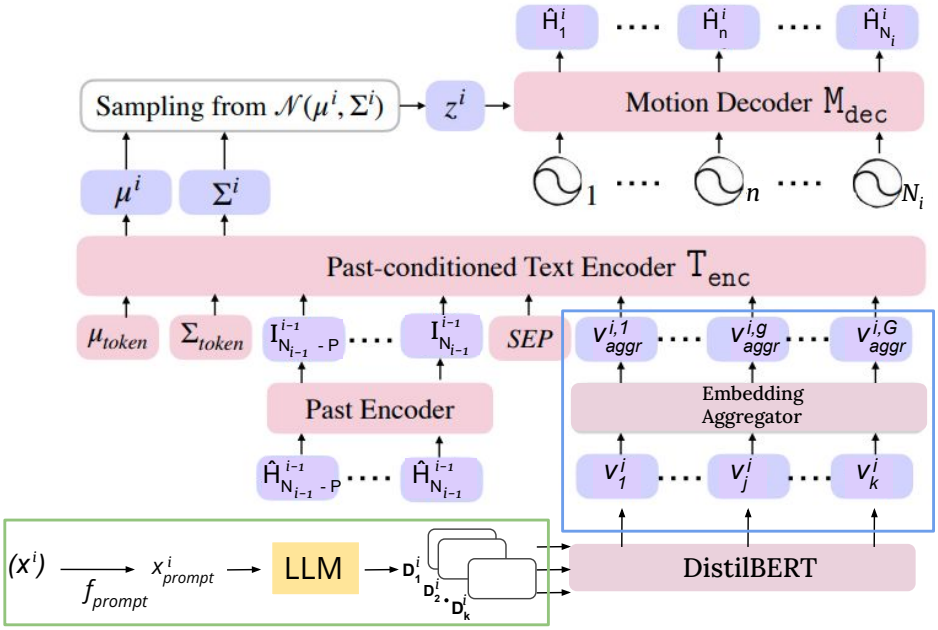


Figure 3.9 Action-GPT-TEACH Overview: We extend TEACH [4] by incorporating LLM (i.e. GPT-3). As TEACH can generate an action sequence for a series of action phrases, we use x^i to denote the i^{th} phrase. The box highlighted in green showcases the k generated text descriptions D^i_j using LLM on input x^i_{prompt}, which is constructed using the prompt function f_{prompt} and action phrase x^i. The k text descriptions D^i_j are input to DistilBERT to generate corresponding sentence embeddings v^i_j, which are aggregated into a single embedding $v^{i,1:G}_{aggr}$ using an embedding aggregator. All the other components apart from the highlighted boxes represent the original architecture of TEACH [4].

LLM-based variant on BABEL [42] using the single action data segments provided by TEACH [4] and generated the metrics on the corresponding test set.

Action-GPT-TEACH: Since TEACH [4] is an extension of TEMOS [38] the process of generating aggregated description embedding v_{aggr} is same as that of Action-GPT-TEMOS. In addition, TEACH can generate an action sequence for a series of action phrases as input $x = \{x^1, \ldots, x^i, \ldots, x^s\}$. So, we compute v_{aggr} s number of times, once for each phrase x^i. In detail, for an action phrase x^i we generate k text descriptions $D^i_1, .., D^i_j, .., D^i_k$. The k text descriptions are input to DistilBERT to generate corresponding description embeddings $v^i_1, .., v^i_j, .., v^i_k$, where $v^i_j \in R^{n^i_j \times e}$, where n^i_j is the number of words in description D^i_j and e is the DistilBERT embedding dimension. The k description embeddings v^i_j are aggregated to a single embedding $v^i_{aggr} \in R^{G \times e}$, where $G = max(n^i_j)$ using the average operation. The aggregated embedding v^i_{aggr} is input to Past-conditioned Text Encoder T_{enc} along with the learnable tokens $\mu_{token}, \sum_{token}$, SEP token and $I^{i-1}_{N_{i-1}-P:N_{i-1}}$, the motion features generated using Past

Encoder, corresponding to the last P frames of previous generated action sequence $\widehat{H}^{i-1}_{N_{i-1}-P:N_{i-1}}$. The later training and inference process followed is the same as that of TEACH.(see Fig. 3.9)

Similar to Action-GPT-TEMOS, we trained this model on a 4 GPU setup with a batch size of 4 on each GPU, keeping rest all the parameters same as provided in TEACH [4]. We generated the metrics for the baseline TEACH [4] using the pre-trained model provided.

3.4.2 Implementation details

We access GPT-3 via OpenAI API Beta Access program. Unless stated otherwise, we use the largest GPT-3 model available, `davinci-002`. The Action-GPT prompt strategy consumes a maximum of 140 tokens together for prompt and generation. We use the completions API endpoint with the parameters temperature and top-p set to 0.5 and 1, ensuring we have well-defined diverse descriptions. All the other parameters are set to default. We conduct all our experiments on cluster machines with Intel Xenon E5 2640 v4 and Nvidia GeForce GTX Ti 12GB GPUs with Ubuntu 16.04 OS.

3.5 Results

3.5.1 Quantitative analysis

We follow the metrics employed in TEACH [4] for quantitative evaluation, namely Average Positional Error (APE) and Average Variational Error (AVE), measured on the root joint and the rest of the body joints separately. Mean local correspond to the joint position in the local coordinate system (with respect to the root), whereas mean global corresponds to the joint position in the global coordinate system. For all the metrics, the smaller the score, the better the generative quality.

Average Positional Error (APE) : For a joint j, APE is calculated as the average of the L2 distances between the generated and ground truth joint positions over the timesteps (N) and the number of test samples (S).

$$\text{APE}[j] = \frac{1}{SN} \sum_{s \in S} \sum_{n \in N} \|\widehat{H}_{s,n}[j] - H_{s,n}[j]\|_2$$

Average Variational Error (AVE) : For a joint j, AVE is calculated as the average of the L2 distances between the generated and ground truth variances.

$$\text{AVE}[j] = \frac{1}{S} \sum_{s \in S} \|\widehat{\sigma}_s[j] - \sigma_s[j]\|_2$$

where $\sigma[j]$ denotes the variance of the joint j,

$$\sigma[j] = \frac{1}{N-1} \sum_{n \in N} (\widetilde{H}[j] - H_n[j])^2$$

$\widetilde{H}[j]$ is calculated as the mean of the joint j over N timesteps.
We calculate four variants of errors for both APE and AVE,

		Average Positional Error ↓				Average Variance Error ↓			
Model	Method	root joint	global traj	mean local	mean global	root joint	global traj	mean local	mean global
MotionCLIP	Default	-	-	0.556	0.541	-	-	0.056	0.02
	Action-GPT	-	-	0.590	0.571	-	-	**0.042**	**0.019**
TEMOS	Default	0.597	0.574	0.162	0.644	0.113	0.112	0.010	0.122
	Action-GPT	**0.561**	**0.540**	**0.151**	**0.605**	**0.101**	**0.100**	**0.010**	**0.109**
TEACH	Default	0.674	0.654	0.159	0.717	0.222	0.220	0.014	0.234
	Action-GPT	**0.606**	**0.586**	**0.159**	**0.650**	**0.204**	**0.202**	**0.014**	**0.216**

Table 3.1 Quantitative Results: We evaluate text-to-motion generation performance by comparing our method with baseline approaches on the BABEL [42] test set. The results demonstrate that integrating our method enhances the performance of baseline models in both APE and AVE metrics.

- *root joint* error is calculated on the root joint using all the 3 coordinates X, Y and Z.
- *global traj* error is calculated on the root joint using only the 2 coordinates X and Y.
- *mean local* error is calculated as the average of all the joint errors in the local coordinate system with respect to the root joint.
- *mean global* error is calculated as the average of all the joint errors in the global coordinate system.

Tab. 3.1 summarizes the results of using our framework in comparison with the default setup for each model. Incorporating detailed descriptions using GPT-3 shows an improvement over all the APE (except for MotionCLIP) and AVE metrics. The metrics of root joints for MotionCLIP are empty since it generates only local pose without any locomotion.

3.5.2 Qualitative analysis

In Fig. 3.10, we provide qualitative comparisons of the model generations. We observe that the generations from our framework are more realistic and well-aligned with the semantic information of the action phrases compared to the default approach. The generations are able to capture the low-level fine-grained details of the action suggested by the original text phrase input.

3.5.2.1 Diverse Generations

Our Action-GPT framework can generate diverse action sequences for a given action phrase x, utilizing the capability of LLMs to generate diverse text descriptions for a given prompt. We generate multiple text descriptions D_i for an action phrase x, and using them as input, multiple action sequences $\widehat{H}_i$ are generated.

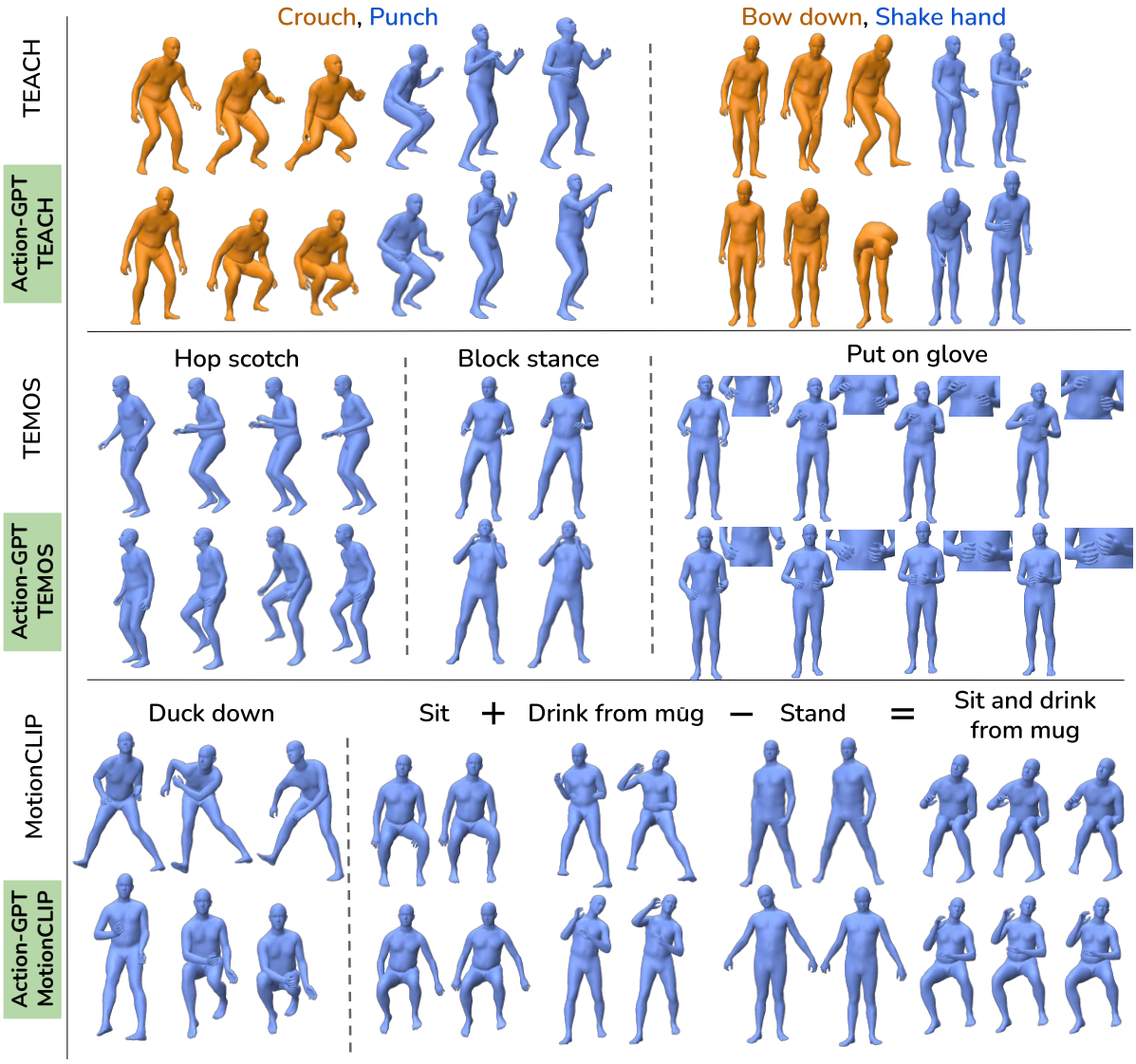


Figure 3.10 Visual comparison of generated motion sequences across models trained on Action-GPT framework on BABEL [42] dataset. Note that the generations using Action-GPT are well-aligned with the semantic information of action phrases. The example in the bottom right row shows latent space editing. Action-GPT is better able to transfer the *drink from mug* style from standing to sitting pose.

3.5.2.2 Zero-Shot Generations

Our approach enables the generation of action sequences $\widehat{H}$ for unseen action phrases (zero-shot). The intuition behind this is that our Action-GPT framework uses low-level detailed body movements textual information to align text and motion spaces instead of using the action phrase directly. Hence the

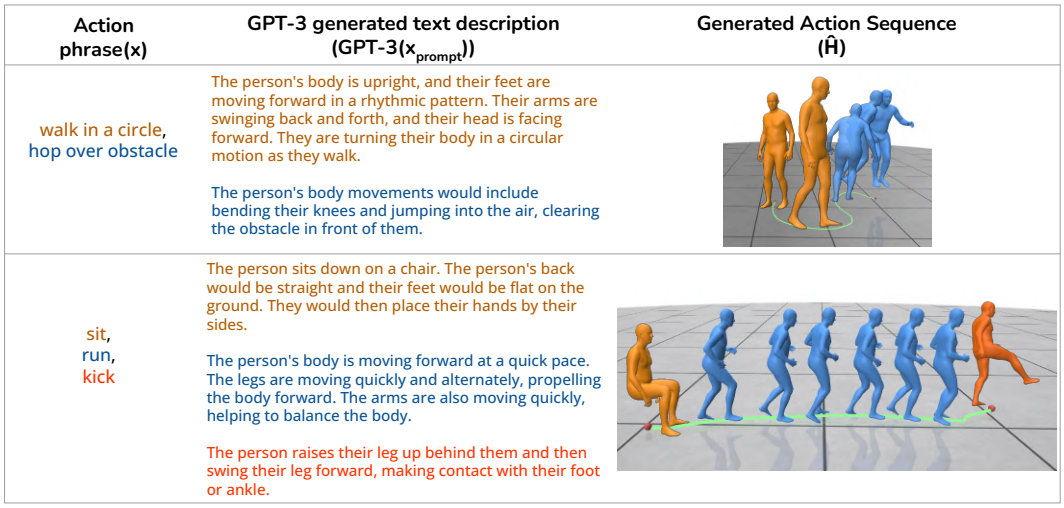


Figure 3.11 Actions with locomotion generated using Action-GPT-TEACH. The color of the text (column 2) represents the detailed text generated by GPT-3 and the same color of the mesh represents the pose sequence generated for the sub-action. The green curve shows the trajectory. The red points show the starting and end points of the motion. Action-GPT is able to generate diverse examples involving locomotion such as walk in a circle, run, hop over obstacle and kick.

action phrase might be unseen, but the low-level body movement details in the generated corresponding text descriptions D_i will not be completely unseen.

Results corresponding to comparison of the baseline models to their LLM-based variants, diverse and zero-shot generations can be found at https://actiongpt.github.io.

3.5.2.3 Locomotion and root movement

Fig. 3.11 shows the diverse actions **involving locomotion** generated using Action-GPT. Our prompt strategy provides detailed descriptions which include the direction of the locomotion along with the coordination among relevant body parts. We further verify this by quantitatively evaluating using Average Positional Error and Average Variance Error of the global trajectory metric – see Table. 3.1. We observe that the Action-GPT variants are better aligned with root trajectory than the baseline.

3.5.3 Ablations

We perform an ablation study to understand the underlying effects of the Action-GPT framework. All of the ablation experiments are carried out on the Action-GPT-TEACH model unless stated otherwise, as it is capable of handling a series of action phrases as input.

		Average Positional Error ↓				Average Variance Error ↓			
Architectural Component	Ablation Details	root joint	global traj	mean local	mean global	root joint	global traj	mean local	mean global
number of generated descriptions (K)	$k = 1$	0.655	0.635	0.159	0.698	0.216	0.214	0.015	0.228
	$k = 2$	0.637	0.617	0.158	0.680	0.211	0.209	0.015	0.223
	$k = 8$	0.632	0.613	**0.157**	0.674	0.212	0.210	0.015	0.224
GPT-3 Capacity	`curie`	0.642	0.622	0.159	0.680	0.216	0.214	0.015	0.228
Ours ($\mathbf{k = 4}$)	`davinci`	**0.606**	**0.586**	0.158	**0.650**	**0.204**	**0.202**	**0.014**	**0.216**

Table 3.2 Ablative Variants: We present performance scores for different ablative variants, examining the impact of (1) the number of GPT-generated descriptions k and (2) the capacity of the language model.

Number of GPT-3 Text Sequences: We analyzed the influence of number of generated descriptions in Action-GPT-TEACH framework by varying k in $1, 2, 4$ and 8. We observed that for all the values of k, Action-GPT-TEACH performs better than the default TEACH and the best results are obtained for $k = 4$ (see Tab. 3.2). Increasing the value of k up to a certain value improves performance. However, aggregating too many descriptions can lead to the injection of excessive noise, which dilutes the presence of text related to body movement details.

Language Model Capacity: Open AI provides GPT-3 in different model capacities, `davinci` being the largest. We analyzed the influence of `curie`, the second largest GPT-3 model, on the motion sequence generations of the Action-GPT-TEACH ($k = 4$) framework. Results show that having a larger model capacity helps in generating more realistic motion sequences, as the generated text descriptions provide much relevant and detailed information as required.

Model	Method	Avg. Training Time (secs)	Avg. Inference Time (secs)
MotionCLIP	Default	585 - 598	0.32 - 0.34
	Action-GPT	700 - 712	0.53 - 0.62
TEMOS	Default	225 - 230	0.8 - 0.96
	Action-GPT	255 - 260	1.44 - 1.76
TEACH	Default	234 - 240	1.6 - 2.1
	Action-GPT	315 - 320	3.4 - 3.8

Table 3.3 Computation Costs Analysis: Comparison of computation costs between baseline models and their Action-GPT variants ($k = 4$).

3.5.4 Computation cost analysis

There will be no change in the number of parameters of the Action-GPT variants when compared to the baseline models as we are using the frozen pre-trained text embedding models. Although the computation time of the models both during training and inference is higher for the Action-GPT ($k = 4$) variants when compared with the original baselines as shown in Table. 3.3. For training time, we take the mean of the time taken for hundred epochs. For inference time, we take a batch-size of 16 and average over 10 repetitions. The increase in the computation time of the Action-GPT variants is because of the usage of k GPT-3 text descriptions, each containing around 128 words, whereas, in the baseline models, a single action phrase is used, which contains around $5 - 8$ words.

3.5.5 Current limitations

- *Finger motion:* Current frameworks use SMPL to represent the pose. SMPL does not contain detailed finger joints. Therefore, GPT-generated descriptions for actions requiring detailed finger motion such as rock-paper-scissors, pointing fingers cannot be generated satisfactorily by the current framework.
- *Complex Actions:* Actions containing complex and vigorous body movements such as yoga and dance poses cannot be generated by our current framework.
- *Long duration action sequences:* Due to the limited duration of training data action sequences (< 10 secs), our method cannot generate long sequences.

3.6 Discussion and Conclusion

The key to good quality and generalizable text-conditioned action generation models lies in improving the alignment between the text and motion representations. Through our Action-GPT framework, we show that such alignment can be achieved efficiently by employing Large Language Models whose operation is guided by a judiciously crafted prompt function. Sentences in GPT-generated descriptions contain procedural text which corresponds to sub-actions. During training, the regularity and frequency of such procedural text likely enable better alignment with corresponding motion sequence counterparts. We also hypothesize that the diversity of procedural sentences in the descriptions enables better compositionality for unseen (zero-shot) generation settings.

The plug-and-play nature of our approach is practical for adoption within state-of-the-art text-conditioned action generation models. Our experimental results demonstrate the generalization capabilities and action fidelity improvement for multiple adopted models, qualitatively and quantitatively. In addition, we also highlight the role of various prompt function components and the benefit of utilizing multiple prompts for improved generation quality.

Chapter 4

MoRAG: Multi-Fusion Retrieval Augmented Generation for Human Motion

4.1 Introduction

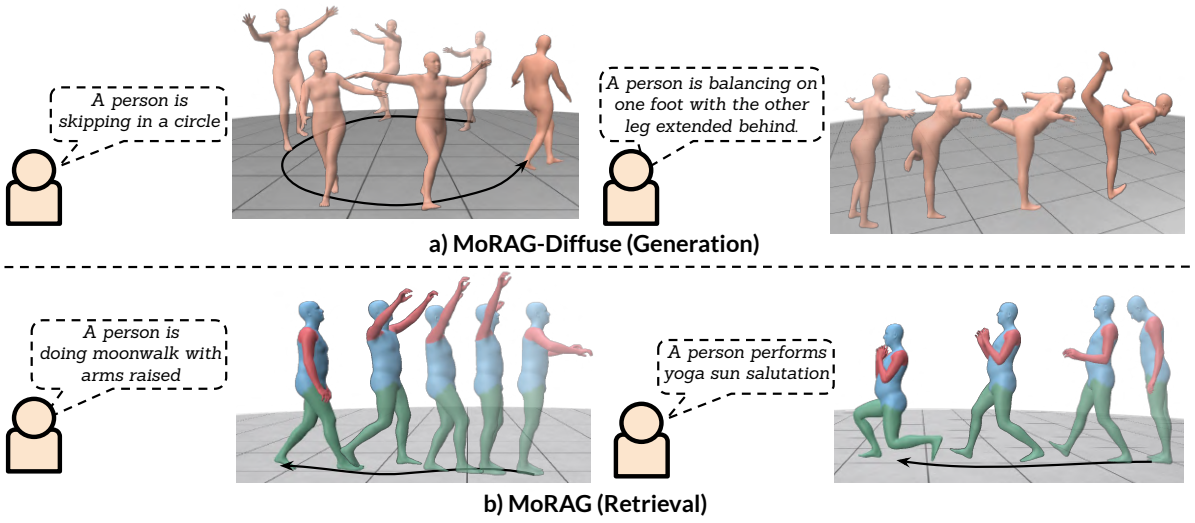


Figure 4.1 MoRAG is a retrieval-augmented framework for generating human motion from text. It integrates part-specific motion retrieval models with large language models to improve the quality of generation and retrieval tasks across various text descriptions. The black arrow illustrates motion translation. In the bottom figures, red, blue, and green represent the retrieved motion for the **hands**, **torso**, and **legs**. The varying transparency in the figure indicates the progression of time steps.

Text-driven human motion generation has seen unprecedented growth in recent years [68, 64, 65, 63, 50, 24]. Numerous works have been proposed for this task, ranging from encoder-decoder style architectures [2, 16, 18] to the recent emerging trend of diffusion-based models [63, 11, 65], which

Motion Retrieval Method	Text Robustness	Generalizability	Diversity	Zero-shot setting
TMR[39]	✗	✗	✗	✗
TMR++[7]	✓	✗	✗	✗
Ours	✓	✓	✓	✓

Table 4.1 Comparision of text-to-motion retrieval approaches - Text Robustness (ability to handle diverse language inputs), Generalizability (adaptation to similar yet altered data), Diversity (capacity to produce varied outputs), and Zero-Shot Setting (performance on previously unseen data).

generate fine-grained, realistic motion sequences. While they can generate high-quality motion sequences for simple or familiar text descriptions similar to those in the training set, they perform poorly with complex or unseen text descriptions.

Retrieval-augmented Generation (RAG) has gained significant attention in recent years for its potential to enhance generative models by incorporating additional information through retrieval methods [14, 66]. By integrating retrieval-based techniques with generative models, RAG produces outputs that are more accurate, contextually relevant, and reliable. Moreover, this additional information helps enhance the model's generalizability across language space and also improves the stochasticity. However, the application of RAG in motion generation is underexplored.

A RAG system typically comprises two key components: the retriever and the generator. The retriever identifies relevant information from a database based on the input query, while the generator uses both the input query and the retrieved information to generate the desired content.

Recently proposed text-to-motion retrieval approaches[39, 58] aim to retrieve full-body motion sequences from the motion database using a contrastive training strategy between text and motion embeddings. However, these retrieval strategies do not perform well when text phrases contain spelling errors, rephrased text sequences, or substitution of synonymous words. (See Fig. 4.7)

Action-GPT[24], TMR++[7] prompt large-language models (LLMs) to provide detailed descriptions as input. However, these approaches are limited in their ability to generate or retrieve motion sequences for text descriptions that are not present in the database, restricting the diversity of output motion sequences and reducing generalization to out-of-domain or unseen text descriptions (Fig. 4.9)

Based on the RAG concept, ReMoDiffuse[64], adopts a hybrid retrieval approach using motion length and CLIP[44]-based text-to-text similarity, which does not incorporate any motion-specific information, which can result in inaccurate retrievals (Fig. 4.2) and motion generation. (Fig. 4.10)

To overcome these shortcomings, we propose a multi-part fusion-based augmented motion retrieval strategy that is capable of constructing diverse and reliable motion sequences. We train part-specific independent motion retrieval models that retrieve motion sequences with movements corresponding to each part aligned with the provided text description. The retrieved part-specific motion sequences are

fused accordingly to construct full body motion sequences, allowing our method to even query unseen text descriptions.

Our experiments show that incorporating the constructed motion sequences as an additional conditioning to the diffusion based motion generation model improves the alignment with the semantic information of the text description and diversity of generated sequences.

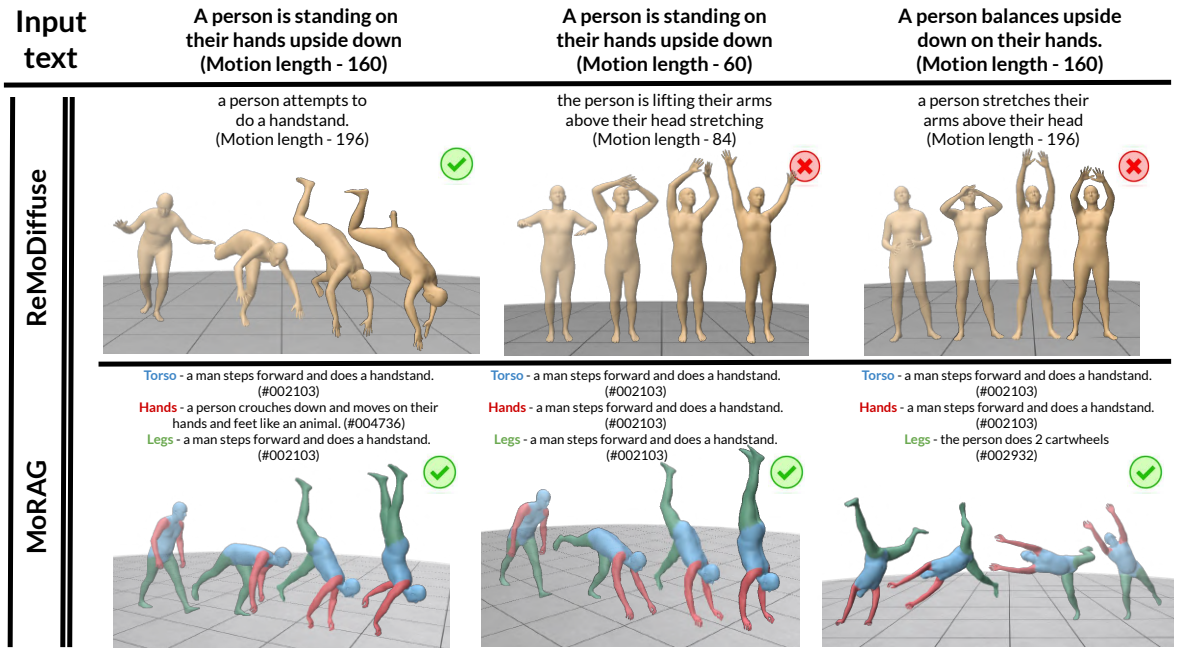


Figure 4.2 MoRAG utilizes part-specific descriptions to effectively retrieve relevant samples, demonstrating robustness to variations in motion length and descriptive text. In contrast, ReMoDiffuse [64], a hybrid approach based on motion length and text similarity, fails to retrieve suitable samples when there are changes in motion length or text. Each figure of ReMoDiffuse displays the retrieved text at the top and the corresponding motion length in brackets. For MoRAG, three part-specific retrieved texts, along with their corresponding HumanML3D [18] ID, are provided using the #. tick and cross to indicate whether the motion corresponds to the input text.

In summary, our contributions are as follows:

- We propose MoRAG, a novel multi-part fusion-based retrieval augmented human motion generation framework to enhance the performance of the diffusion based motion generation model.

- We adapt *part-wise motion retrieval* approach that utilizes *generative prompts* to construct motion sequences that align with the provided text description.

- Motion sequences constructed using our retrieval strategy exhibit superior generalization and diversity, as shown by the qualitative and quantitative analysis.

4.2 Related Works

Text-conditioned human motion generation : The early research efforts concentrated on encoder-decoder models with multimodal joint embedding space spanning both text and motion domains[2, 1]. Text2Action[1] proposed a GAN based generative model constituting RNN based text encoder and action decoder. JL2P[2] focused on learning a joint embedding space for language and pose on the motion reconstruction task using text and motion embedding. Ghosh *et al.*[16] further improved the joint embedding space using a hierarchical two-stream model where two motion representations are learned, one for each lower body and upper body.

To enhance the text-to-motion generalizability, MotionCLIP[49] incorporated image embedding of the poses into the training paradigm alongside text embedding. Both the image and text embedding are generated using CLIP.[44]. TEMOS[38] proposed a transformer based VAE encoding approach for text and motions to improve the diversity of generated motion sequences. TEACH[4] extends TEMOS[38] with an additional past motion encoder to generate long motion sequences using the text description and previous sequence. T2M-GPT[62] transforms the challenge of text-conditioned motion generation into a next-index prediction task by encoding motion sequences as discrete tokens and utilizing a transformer to predict future tokens.

Unlike the above mentioned approaches, we propose a novel multi-part fusion-based augmented motion retrieval strategy to improve the diversity of retrieved motions and to enhance their generalizability for unseen text descriptions.

Motion Diffusion Models : With recent advancements in diffusion models for text and image domain tasks, several works have been proposed in the area of text-to-motion generation. MotionDiffuse[63] incorporated efficient DDPM in motion generation tasks to generate diverse, variable-length, fine-grained motions. MDM[50] is a lightweight diffusion model that utilizes a transformer-encoder backbone. Instead of predicting noise, it predicts motion samples, allowing geometric losses to be used as training constraints. MLD[11] adapted the diffusion process on latent motion space instead of using raw motion sequences, generating better motions at a reduced computational overhead.

However, these diffusion-based models struggle to generalize across the language space, especially when dealing with complex or unusual text descriptions. Recent text-to-image generation works introduced retrieval-augmented pipelines in their frameworks [10] to address such issues. ReMoDiffuse[64] extended MotionDiffuse[63] by integrating a hybrid retrieval mechanism to refine the denoising process. We improve the generalizability and diversity of ReMoDiffuse by the inclusion of large language models (LLMs) and the integration of part-specific retrieval.

Text-to-Motion retrieval : Recently, significant progress has been made by multi-modal retrieval based systems in the field of text-to-image [27, 43, 61, 44] and text-to-video [15, 12]. However, it has been

underexplored in the field of text-to-motion due to the lack of large and diverse annotated motion capture datasets. Recent works such as BABEL[42] and HumanML3D[18] have provided detailed description annotations for the large-scale motion capture collection AMASS[34]. Following, a few works have been proposed in the field of text-to-motion retrieval.

Initially, motion generation works[18] used retrieval as a performance metric for evaluation purposes. TMR[39] is the first work to showcase text-to-motion retrieval as a standalone task. To query motions, TMR[39] adopted the idea of contrastive learning from CLIP[44] and extended text-to-motion generation model TEMOS[38]. Although TMR demonstrated impressive results, there is a large scope for improvement in the generalizability of the model over language space. LAVIMO[58] integrated human-centric videos as an additional modality in the task of text-to-motion retrieval to effectively bridge the gap between text and motion. TMR++[7] extended TMR by leveraging LLMs in the motion retrieval pipeline via label augmentations to increase the robustness and generalizability. However, a significant gap remains in utilizing these existing retrieval strategies for retrieval-based human motion generation approaches due to their lack of diversity and generalizability for complex or unseen text descriptions.

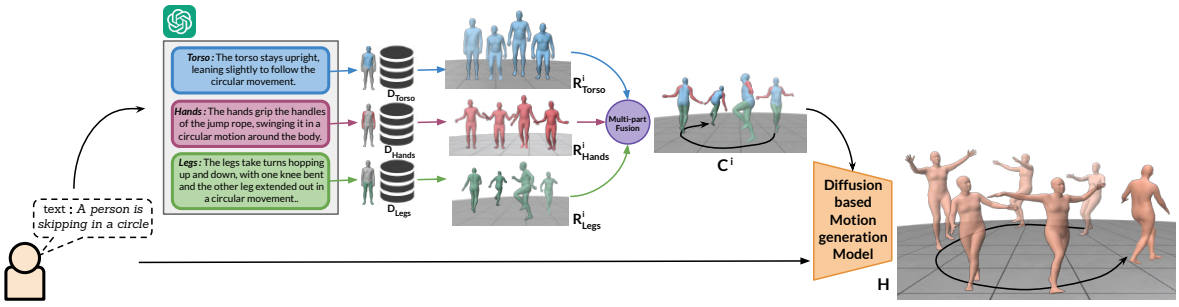


Figure 4.3 MoRAG Overview: Given a text description `text`, we generate part-specific descriptions corresponding to "Torso," "Hands," and "Legs" by prompting an LLM. These generated descriptions are used as queries to retrieve corresponding part-specific motions: R^i_{torso}, R^i_{hands}, and R^i_{legs} from the motion databases D_{torso}, D_{hands}, and D_{legs}, respectively. The retrieved motions are then fused to construct a full-body motion sequence C^i that aligns with the input text. The constructed motion samples are used as additional information in the motion generation pipeline during both training and inference, alongside the input text, to further improve model performance.

4.3 Our approach (MoRAG)

Fig. 4.3 illustrates our multi-fusion retrieval-augmented human motion generation framework, **MoRAG**, which aims to enhance the performance of diffusion based motion generation model by leveraging additional motion information constructed using part-specific motion retrieval. Given an input text descrip-

tion `text`, we generate N diverse, semantically coherent human motion sequences $\{H^1, \ldots, H^n, \ldots, H^N\}$. We prompt the input text description to an LLM to generate motion descriptions specific to the "Torso," "Hands," and "Legs" (Sec.4.3.2). These descriptions are used for the retrieval of part-specific full-body motion sequences from pre-computed part-specific motion databases (Sec.4.3.3). The retrieved motion sequences are then fused to construct full-body motion sequences, which serve as additional knowledge for the diffusion model (Sec.4.3.4). This methodology enhances the model's ability to effectively handle both typical and complex/unseen input conditions (Sec.4.3.5).

4.3.1 Augmented Motion Retrieval Strategy

The key component of the MoRAG framework is the retrieval of diverse and semantically aligned motion samples from the database based on the given input text query. Existing text-to-motion retrieval methods [39, 58, 7] typically retrieve full-body motion samples directly from the database. However, these approaches overlook the fact that actions are frequently characterized by localized dynamics, often involving only small subsets of joint groups, such as the hands (e.g., 'eating') or legs (e.g., 'sitting'). [51, 20] This results in two significant issues: (i) limited generalizability and (ii) lack of diversity in the retrieved samples. (See Table 4.1 and Fig. 4.9 (a)) This is due to the limited availability of text-motion annotated datasets. Although BABEL [42] and HumanML3D [18] provide detailed text annotations for the large-scale motion capture collection AMASS [34], they are still insufficient to generalize across the language space. However, the AMASS dataset contains extensive low-level body parts information that holds the potential to generalize across a significantly broader language space.

Based on this observation, we design independent part-specific motion retrieval models that can retrieve full-body motion sequences with movements corresponding to specific parts aligned with the provided text description. This enables dedicated part descriptions for retrieval of actions involving specific part movement. By composing these motion samples, we can construct full-body motion sequences that are semantically coherent with the given text input. The composition also improves the expressivity of motions, since fine-grained motion details are often expressed in text in terms of body parts. The wide variety of composing combinations provides huge diversity in the constructed motion samples. Integrating these samples into a motion generation pipeline as additional information can enhance the model's performance. We also observe better generations for unseen text descriptions. (See Fig. 4.10 (b))

The objective is to construct a series of motion sequences $\{C^1, \ldots, C^i, \ldots, C^k\}$ from the motion database ranked from 1 to k where each motion sequence C^i is represented as a sequence of human poses $\{C^i_1, \ldots, C^i_t, \ldots, C^i_{f_i}\}$ with f_i representing the number of timesteps for motion C^i. The motion retrieval strategy in MoRAG comprises three steps: (1) generating part-specific body movement descriptions, (2) retrieving part-specific motion sequences, and (3) composing the retrieved motion sequences. Details of each are provided in the following section.

4.3.2 Generation of part-specific descriptions

Given the text description `text`, we generate part specific body movement descriptions using an LLM as a knowledge engine. We construct a suitable prompt $\texttt{text}_{prompt}$, for `text` using a prompt function f_{prompt}, comprising of three components:

(i) **Task instructions**, to specify the details of our task:

"The instructions for this task is to describe the listed body parts' position and movements in a sentence using simple language. ['Torso',' Hands', 'Legs']"

(ii) **Few-shot examples**, provides a set of examples consisting of diverse action descriptions to determine the format of the output we are expecting.

(iii) **Query**, to incorporate the input text `text` to generate part-specific body movement and orientation information.

"*Query: Describe the below body parts position and movements involved in the action [`text`] in a sentence using simple language. 1) Torso 2) Hands 3) Legs*".

Specifically prompting for the position provides the global orientation of the body parts which results in better retrieval.

The constructed prompt is then passed through LLM to generate descriptions of the positions and movements for the specified body parts denoted as $\texttt{text}_{torso}$, $\texttt{text}_{hands}$ and $\texttt{text}_{legs}$. Training a retrieval model on these body part descriptions enables the retrieval of motions for rephrased and spell-error text phrases, thereby having a better generalization over language space. (Fig.4.7)

Table 4.2 presents the LLM outputs for various text descriptions generated using our prompt function, demonstrating its effectiveness in motion retrieval. We observe that the generated descriptions are often correlated with other body parts, which led us to train part-specific retrieval models on full-body motion sequences.

4.3.2.1 Significance of "position"

To train independent part-specific retrieval models, $MoRAG_p$, for $p \in \{torso, hands, legs\}$, it is essential to obtain movement information specific to each part. However, since the framework includes a composition step that combines the retrieved motions into a single motion sequence, relying solely on movement information is insufficient. The composition also requires positional information. For example, as shown in Fig. 4.4 for the text: *"A person is swimming"*, explicitly prompting for the positional information allows the leg description to reflect its relative position to the ground, thereby retrieving the correct motion sample. The global orientation of the leg corresponding retrieved sample is important as it will determine the global orientation of the composed motion sequence.

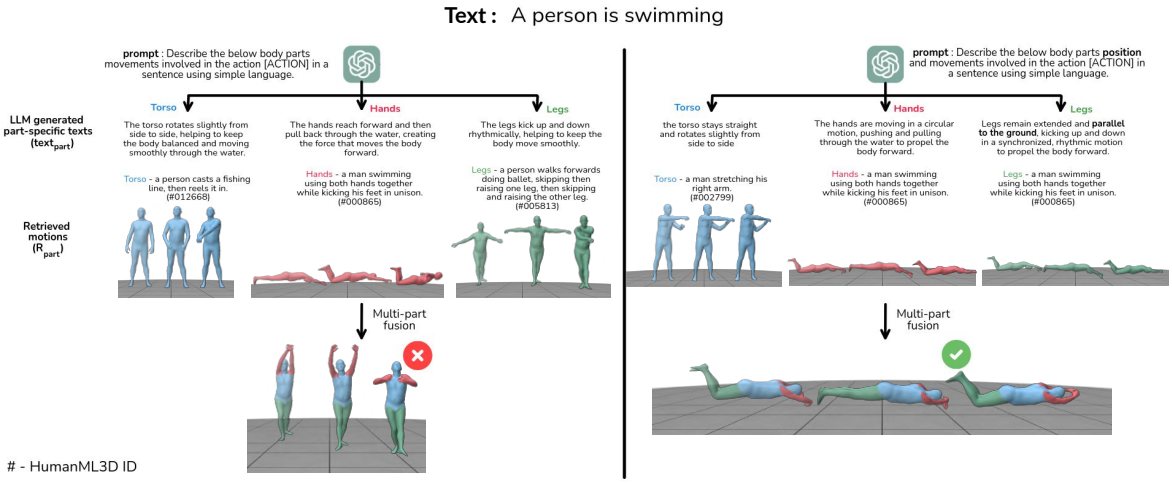


Figure 4.4 Position Significance: Impact of positional information in leg descriptions on motion retrieval accuracy and global orientation consistency in composed sequences for the text: *"A person is swimming"*

4.3.2.2 Issue with left/right parts retrieval strategy

In MoRAG, we avoided retrieving left and right hands or legs separately, as their movements risk becoming asynchronous when composed into a single motion sequence. This challenge is depicted in Fig. 4.5, which illustrates asynchronous motion composition of **hands** for the text: *"A person is clapping his hands."*

4.3.3 Multi-part motion retrieval

As shown in Fig. 4.6, MoRAG uses 3 independently trained TMR[39] models, $MoRAG = \{MoRAG_{torso}, MoRAG_{hands}, MoRAG_{legs}\}$ corresponding to the respective body part. We do not train separate left and right body parts models to avoid asynchronous movements in the composed motion. For a part $p \in \{torso, hands, legs\}$, the retrieval model $MoRAG_p = \{T_p^{Enc}, M_p^{Enc}, M_p^{Dec}, \mathcal{D}_p^{Enc}\}$ consists of a text encoder, motion encoder, motion decoder, and motion database respectively. The model architecture for the encoder and decoder are based on TEMOS[38].

Text Encoders (T_p^{Enc}) : The LLM-generated part-specific motion descriptions of the text sequence `text` are first passed through a pre-trained and frozen DistilBERT[46] to generate features $\mathcal{F}_p^T$ for each part-wise text description $\texttt{text}_p$. Along with the features $\mathcal{F}_p^T$, two learnable distribution tokens are passed as input to the text encoders. The outputs corresponding to the distribution tokens passed are considered as Gaussian distribution parameters (μ_p^T and σ_p^T) from which the latent vector Z_p^T is

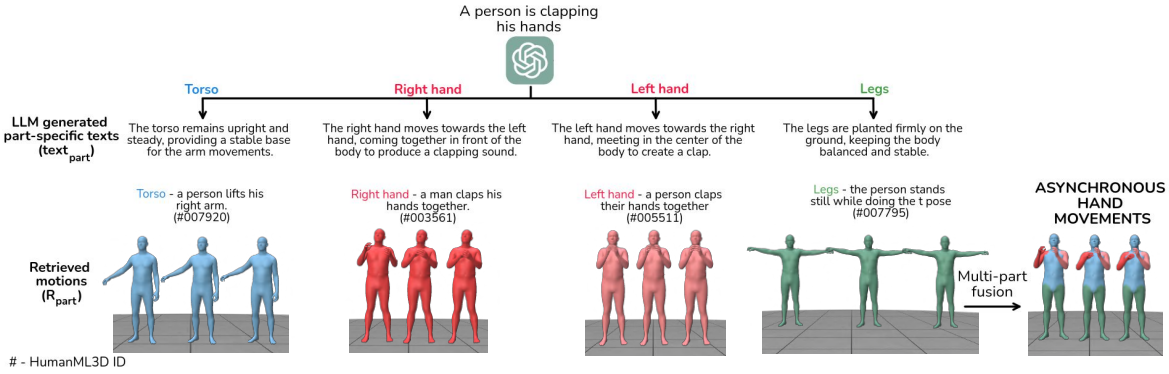


Figure 4.5 Asynchronous motion caused by separate retrieval of left and right parts: Illustration of asynchronous motion composition resulting from separate left and right hand retrievals for the text: *"A person is clapping his hands."*

sampled using reparametrization trick[26].

$$(\mu_p^T, \sigma_p^T) = T_p^{Enc}(\mathcal{F}_p^T) \tag{4.1}$$

$$Z_p^T \sim \mathcal{N}(\mu^T, \sigma^T) \tag{4.2}$$

Motion Encoders (M_p^{Enc}) : Similarly, Z_p^M is obtained from the motion encoders by inputting the corresponding full-body motion sequence $M_{1:f}$ associated with the text description `text` and duration f.

$$(\mu_p^M, \sigma_p^M) = M_p^{Enc}(M_{1:f}) \tag{4.3}$$

$$Z_p^M \sim \mathcal{N}(\mu^M, \sigma^M) \tag{4.4}$$

During retrieval, instead of sampling, we directly use the embedding corresponding to the mean parameter (i.e $Z_p^T = \mu_p^T$ and $Z_p^M = \mu_p^M$)

To enhance the effectiveness of motion retrieval, we train all three motion encoders (M_p^{Enc}) using full-body motion sequences instead of just the respective body parts. This approach is based on the observation that LLM-generated part-specific descriptions contain information about the queried body part about other body parts. Utilizing full-body motion sequences allows us to leverage intra-joint information, resulting in more coherent and semantically accurate motion retrieval.

Motion Decoders (M_p^{Dec}) : Motion decoders input a latent vector Z and sinusoidal positional encoding of the duration f and output a full body motion sequence $\hat{M}_{1:f}$ non-autoregressively. The input latent vector Z is obtained from one of the two encoders during training. However, since our task is motion retrieval, the decoder is not used during inference.

$$\hat{M}_{1:f} = M_p^{Dec}(Z) \tag{4.5}$$

$$Z \in \{Z_p^T, Z_p^M\} \tag{4.6}$$

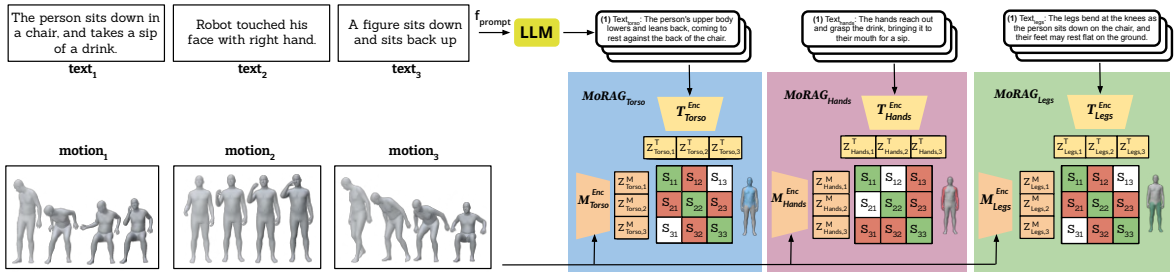


Figure 4.6 MoRAG Training: Our objective is to construct three independent part-specific motion databases. The training paradigm includes three motion retrieval models: $MoRAG_{torso}$, $MoRAG_{hands}$, and $MoRAG_{legs}$, each corresponding to a specific body part. We train these three models independently using part-specific body movement descriptions generated by LLMs for text phrases $\texttt{text}_i$ and their corresponding full-body motion sequences $\texttt{motion}_i$. We adopt a contrastive training objective between part-specific text embeddings ($Z^T_{p,i}$) generated by text encoders (T^{Enc}_p) and motion embeddings ($Z^M_{p,i}$) generated by the corresponding part-specific motion encoder(M^{Enc}_p). The diagonal elements, representing positive pairs (green), are maximized, while the off-diagonal elements, representing negative pairs with text similarity below a threshold (red), are minimized. For simplicity, we do not visualize the motion decoder, but we follow a similar training procedure as described in [39].

Loss : Each retrieval model, $MoRAG_p$ is trained with the loss $\mathcal{L}^p$ [39]:

$$\mathcal{L}^p = \mathcal{L}_R + \lambda_{KL}\mathcal{L}_{KL} + \lambda_E\mathcal{L}_E + \lambda_{NCE}\mathcal{L}_{NCE} \tag{4.7}$$

$\mathcal{L}_R$ is the motion reconstruction loss given motion and text embeddings to the decoder. $\mathcal{L}_{KL}$ is the Kullback-Leibler(KL) divergence loss composed of four losses. The first two are for the text and motion distributions with normal distribution and the other two are between text and motion distributions. $\mathcal{L}_E$ is the cross-modal embedding similarity loss between both text and motion latent embeddings Z^T_p and Z^M_p.

$\mathcal{L}_{NCE}$ is the contrastive loss which is based on InfoNCE[53] formulation, used to better structure the cross-model latent space. The text and its corresponding motion embedding are considered positive pairs ($Z^T_{p,i}$ and $Z^M_{p,i}$), whereas all other combinations are considered to be negative ($Z^T_{p,i}$ and $Z^M_{p,j}$) with $i \neq j$. Similarity matrix S computes the pairwise cosine similarities for all the pairs, S_{ij} =cos($Z^T_{p,i}, Z^M_{p,j}$). However, not all negative pairs are involved in the loss computation. Text-motion pairs with text description similarities above a certain threshold, referred to as 'wrong negatives', are filtered out from the loss computation. The threshold to filter negatives is set to 0.8. These text similarities are computed using MPNet[48].

$$\mathcal{L}_{NCE} = -\frac{1}{2N}\sum_{i}\left(log\frac{e^{S_{ii}/\tau}}{\sum_j e^{S_{ij}/\tau}} + log\frac{e^{S_{ii}/\tau}}{\sum_j e^{S_{ji}/\tau}}\right) \tag{4.8}$$

Motion database ($\mathcal{D}_p$) : Post training, we create a database consisting of three key-value tables for every body part where each key is a unique identifier for a motion sample from the AMASS[34] database. The corresponding value is a vector inferred from the motion encoder. During retrieval, the LLM-generated part description is encoded into a query vector for every text encoder, T_p^{Enc}. We use this query vector to search the corresponding vector indexes, finding the k-nearest neighbors in the embedding space using cosine similarity. The corresponding k full-body motion sequences $\{R_p^1, \ldots, R_p^i, \ldots, R_p^k\}$ are retrieved for each body part p.

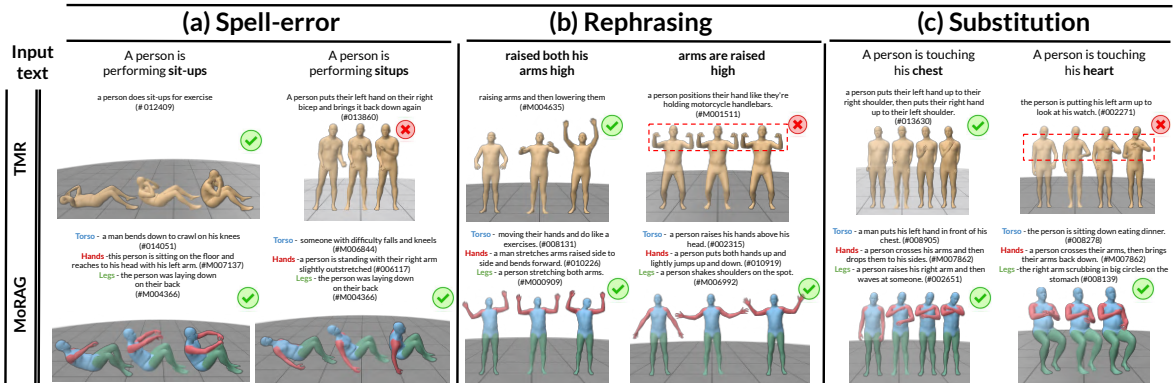


Figure 4.7 LLM Importance: Incorporating part-wise descriptions generated by LLMs into text-to-motion retrieval improves generalization over the language space. (a) **Spell Error** - MoRAG successfully retrieves and constructs the correct motion sequence when 'sit-ups' is replaced with 'situps', unlike TMR[39]. (b) **Rephrasing** - MoRAG effectively retrieves the correct motion sequence even when the voice is changed from active to passive. (c) **Substitution** - MoRAG accurately retrieves the correct motion sequence when 'chest' is replaced with its synonym 'heart'.

4.3.4 Spatial motion composition

The retrieved motion sequences $\{R_p^1, \ldots, R_p^i, \ldots, R_p^k\}$ are composed such that the i_{th} sequence corresponding to each part p is used to construct the i_{th} full-body motion sequence C^i. This results in k full-body motion sequences $\{C^1, \ldots, C^i, \ldots, C^k\}$, which are used as additional guidance for the motion diffusion model. We followed a rank-by-rank combination approach to generate these top-k sequences. However, alternative combination methods could be employed to create a significantly larger number of sequences. Our composition approach is similar to SINC[5] but we do not require the use of an LLM for mapping joints from the retrieved sequences to the composed sequence.

text	LLM Output			MoRAG part-specific retrieval		
	$\texttt{text}_{torso}$	$\texttt{text}_{hands}$	$\texttt{text}_{legs}$	$\texttt{R}_{torso}$	$\texttt{R}_{hands}$	$\texttt{R}_{legs}$
A person is standing and raising both hands	The person's torso is upright and still, while standing tall and balanced.	The person's hands are being lifted up towards the sky, using their arms to extend upwards.	The legs are steady and stable, acting as a strong foundation to support the body as the arms raise up.	a person uses their hands to clap (#002573)	a person raises his hands above his head. (#002315)	a person raises his hands above his head. (#002315)
A person is standing and raising single hand	The person's torso is upright and still, while their arm raises up.	One hand is lifted up from the side of the body, extended upwards and reaching toward the ceiling.	The legs are supporting the person's weight, standing firmly on the ground.	(person stands still and lifts right hand to face and mouth area#000813)	a person raises his right arm and then lowers it. (#001179)	a figure claps around shoulder height (#007523)
A person is standing on one leg and raising both hands	The person's upper body is straight and centered while standing on one leg.	Both hands are lifted above the head, reaching towards the sky.	One leg is holding the person's weight while the other is slightly lifted off the ground, balancing the body.	a person grabs their right foot and places it on their left thigh, and balances on one foot and then does the same with the other foot. (#006123)	raising and lowering arms. (#011583)	a person balances on their left leg and then their right. (#009917)
A person is standing on one leg and raising single hand	The person's body is upright and balanced on one leg, with the other leg lifted off the ground.	One hand is raised up in the air, reaching toward the ceiling or sky.	The standing leg is firmly planted on the ground, while the other leg is lifted up and may be slightly bent or straight.	the person raises their left foot up to their knee and then kicks their foot out, then returns their foot to their knee. (#004012)	a person raises his right arm and then lowers it. (#001179)	person balances on left leg then with arms up high has arms fully extended keeping balance on left leg (#004180)
A person is running with their arms crossed.	The person's upper body is straight and upright, while their chest and shoulders may be slightly forward as they run.	The hands are crossed over the chest, alternating in front of the body as the person runs.	The legs are moving back and forth in a rhythmic motion, propelling the person forward as they run.	a person jogs forward with arms moving to his side. (#002755)	a figure stands in place, crossing its arms (#003973)	a person slowly jogs to the left to right and the jogs back into place (#004273)
A person is running with their arms stretched out to the sides.	The person's torso is upright and slightly leaning forward as they jog, with their chest and shoulders in a relaxed position.	The person's hands are held out to the sides at shoulder level, moving rhythmically with each step.	The legs are moving in a bouncing motion, alternating between the left and right sides as the person jogs forward.	jogging forward in medium pace. (#000972)	the person is flying like a airplane. (#006433)	a person slowly jogs to the left to right and the jogs back into place (#004273)
A person is walking with their arms circling around.	As the person runs, their torso is upright and facing forward, with their chest and stomach comfortably relaxed.	The person is moving their arms back and forth in a swinging motion, with their arms stretched out to the sides.	The legs are taking turns lifting off the ground and propelling the person forward in a steady, rhythmic motion.	running from side to side. (#014305)	this person moves both arms out to his sides in a large circular motion then walks forward. (#005375)	a person doing a casual walk (#004183)
A person is walking with their arms crossed.	The person's upper body remains upright while they walk with their arms crossed, with their spine and shoulders in a stable position.	The person's hands are placed on opposite shoulders, crossing in front of their torso as they walk.	The legs are alternately lifting and stepping forward, propelling the person forward while they walk.	a person walks on uneven ground whilst holding on to handrail. (#001029)	a person crosses their arms in front of their chest, then drops them back at their sides. (#001014)	a person walks forward while holding out their arms for balance (#002087)

Table 4.2 Prompt Examples : We illustrate the LLM-generated part-specific outputs for text descriptions alongside their corresponding top-1 retrieval results to demonstrate the effectiveness of our prompt strategy. The HumanML3D [18] ID for the retrieved motions is indicated with the # symbol.

To construct the composed motion sequence C^i using the corresponding retrieved full-body motion sequences, $\{R^i_{torso}, R^i_{hands}, R^i_{legs}\}$, we follow these steps: (1) Trimming all three retrieved sequences to the length of the shortest one; $f_{min} = \min_p(f_p)$, (2) Selecting the respective body part's joint information from the corresponding retrieved motions. We follow the SMPL[30] skeleton structure with the first 22 joints and partition it into three disjoint sets of joints: $J = \{j_{torso} \cup j_{hands} \cup j_{legs}\}$.

$$C^i_f[j_p] = R^i_{p,f}[j_p] \tag{4.9}$$

$$p \in \{torso, hands, legs\}, f \in [1, f_{min}]$$

(3) Choosing the global orientation and translation from R^i_{legs}, as leg motion is closely associated with changes in global translation and orientation. Fig. 4.8 provides an illustration of the spatial motion composition procedure using retrieved part-specific motion samples for the text: *"A person is standing on one leg and raising both hands."*

4.3.5 MoRAG-Diffuse

For generation, we extend ReMoDiffuse[64], a diffusion-based model, by incorporating our retrieval mechanism within the motion generation pipeline. Unlike ReMoDiffuse[64], we adapt our multi-part composed motion, rather than their motion length and text based similarity retrieval approach.

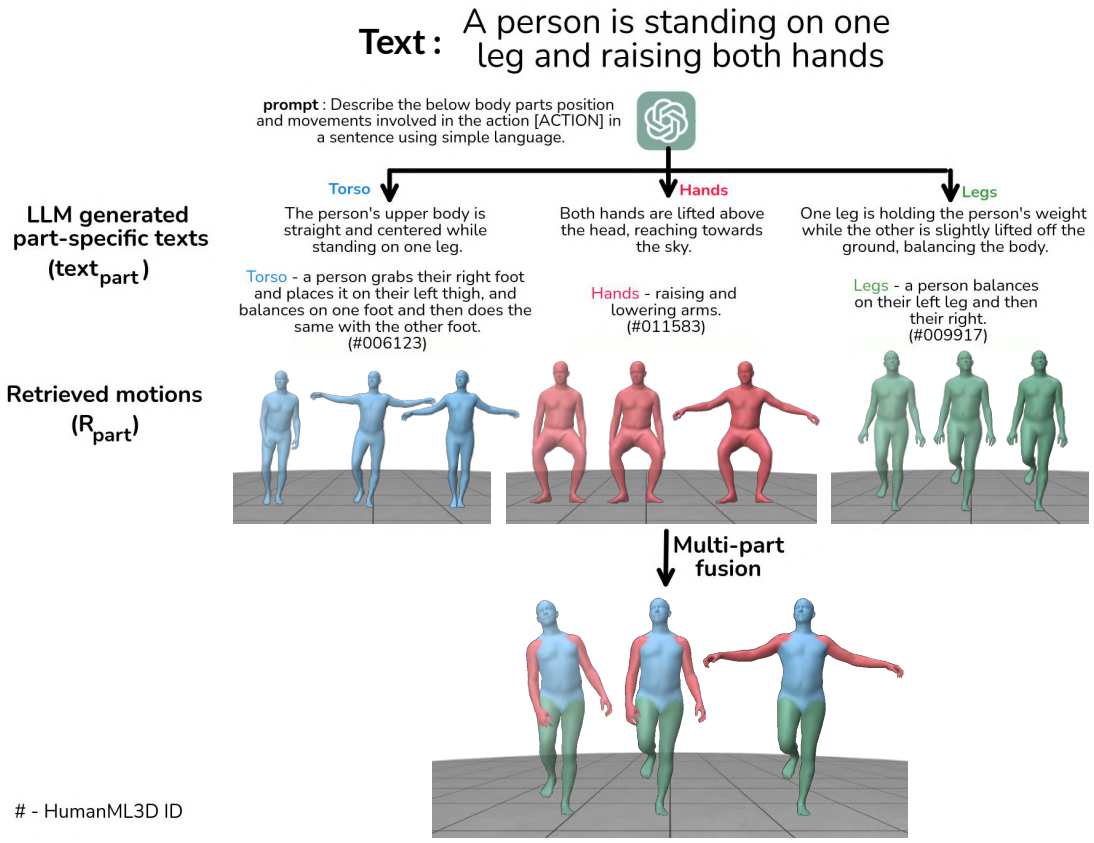


Figure 4.8 Spatial Motion Composition: Illustration of spatial motion composition using retrieved part-specific motion samples for the text: *"A person is standing on one leg and raising both hands"*

For the top-k retrieved motion sequence C^i, we follow the 263 dimension motion representation as in [18], where each human pose C_t^i is represented by $(\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, \mathbf{j}^p, \mathbf{j}^v, \mathbf{j}^r, \mathbf{c}^f)$, where $\dot{r}^a \in \mathbb{R}$ is root angular velocity along Y-axis; $\dot{r}^x \in \mathbb{R}$, $\dot{r}^z \in \mathbb{R}$ are global root velocities in X-Z plane; $\dot{r}^y \in \mathbb{R}$ is root height; $\mathbf{j}^p \in \mathbb{R}^{3\times n(J)}$, $\mathbf{j}^v \in \mathbb{R}^{3\times n(J)}$, $\mathbf{j}^r \in \mathbb{R}^{6\times n(J)}$ are the local pose positions, velocity and rotation respectively. $\mathbf{c}^f \in \mathbb{R}^4$ is the foot contact features calculated by the heel and toe joint velocity. $n(J)$ is the number of joints and T_i represents the number of timesteps for motion C^i.

To effectively utilize information from retrieved motion samples, we use the Semantics-Modulated Transformer (SMT) introduced in ReMoDiffuse[64]. It comprises of N identical decoder layers, each featuring a Semantics-Modulated Attention (SMA) layer and a feed-forward network (FFN) layer. The SMA layer integrates information from the input description and the retrieved samples, refining the noised motion sequence throughout the denoising process. The SMA layer consists of a cross-attention mechanism where the noised motion sequence serves as the query vector Q. The key vector K and value vector V are derived from three sources of data: (1) the noised motion sequence itself; (2) CLIP's text features of the input description `text` which are further processed by two learnable transformer

encoder layers; and (3) the retrieved motion and text features R^m, R^t, extracted using transformer-based encoders from the retrieved samples. As our composed motion samples do not have associated text sequences for generating text features R^t, we use the features of the input text description `text`.

4.4 Experiments

First, we describe the dataset (Sec. 4.4.1) and implementation details (Sec.4.4.2) used in our experiments. Following that, we analyze the effectiveness of the MoRAG framework, comparing it to previous research works by providing an analysis of both retrieval and generation results. (Sec.4.5).

4.4.1 Dataset

We chose HumanML3D[18] to evaluate our framework due to its extensive and diverse collection of motions paired with a wide range of text annotations. It provides annotations for the motions in the AMASS[34] and HumanAct12[19] datasets. On average, each motion is annotated three times with different texts, and each annotation contains approximately 12 words. Overall, HumanML3D consists of 14,616 motions and 44,970 descriptions. The data is augmented by mirroring left and right. We follow the same splits as TMR[39] and ReMoDiffuse[64] to train the retrieval and generation models respectively.

4.4.2 Implementation Details

We use OpenAI's GPT-3.5-turbo-instruct as the large language model (LLM) due to its efficiency in understanding and executing specific instructions and its ability to provide direct answers to questions. The proposed prompting strategy consumes a maximum of 256 tokens together for prompt and generation. We use the completions API endpoint with the default parameters to generate the desired part-specific descriptions.

For part-specific retrieval models ($MoRAG_p$), we use AdamW [31] optimizer with a learning rate of 0.0001 and a batch size of 32. The latent dimensionality of the embeddings is 256. We set the temperature τ to 0.1, and the weight of the contrastive loss term λ_{NCE} to 0.1. Other hyperparameter values are used similarly to those in TMR [39].

For MoRAG-Diffuse, we use similar settings as that of ReMoDiffuse[64] used for HumanML3D[18]. For the diffusion model, the variances β_t are spread linearly from 0.0001 to 0.02, and the total number of diffusion steps is 1000. Adam optimizer with a learning rate of 0.0002 is used to train the model. MoRAG-Diffuse was trained on an NVIDIA GeForce RTX 2080 Ti, with a batch size of 64, using initial weights of ReMoDiffuse[64] for 50k steps.

Methods	R Precision ↑			FID ↓	MM Dist ↓	Diversity →	MultiModality ↑
	Top 1	Top 2	Top 3				
Real motions	$0.511^{\pm 0.003}$	$0.703^{\pm 0.003}$	$0.797^{\pm 0.002}$	$0.002^{\pm 0.000}$	$2.974^{\pm 0.008}$	$9.503^{\pm 0.065}$	-
MDM[50]	$0.320^{\pm 0.005}$	$0.498^{\pm 0.004}$	$0.611^{\pm 0.007}$	$0.544^{\pm 0.044}$	$5.566^{\pm 0.027}$	$9.559^{\pm 0.086}$	$2.799^{\pm 0.72}$
MotionDiffuse[63]	$0.491^{\pm 0.001}$	$0.681^{\pm 0.001}$	$0.782^{\pm 0.001}$	$0.630^{\pm 0.001}$	$3.113^{\pm 0.001}$	$9.410^{\pm 0.049}$	$1.553^{\pm 0.042}$
MLD[11]	$0.481^{\pm 0.003}$	$0.673^{\pm 0.003}$	$0.772^{\pm 0.002}$	$0.473^{\pm 0.013}$	$3.196^{\pm 0.010}$	$9.724^{\pm 0.082}$	$2.413^{\pm 0.079}$
ReMoDiffuse[64]	$0.510^{\pm 0.005}$	$0.698^{\pm 0.006}$	$0.795^{\pm 0.004}$	$0.103^{\pm 0.004}$	$2.974^{\pm 0.016}$	$9.018^{\pm 0.075}$	$1.795^{\pm 0.043}$
FineMoGen[65]	$0.504^{\pm 0.002}$	$0.690^{\pm 0.002}$	$0.784^{\pm 0.002}$	$0.151^{\pm 0.008}$	$2.998^{\pm 0.008}$	$9.263^{\pm 0.094}$	$2.696^{\pm 0.079}$
MoRAG-Diffuse	$0.511^{\pm 0.003}$	$0.699^{\pm 0.003}$	$0.792^{\pm 0.002}$	$0.270^{\pm 0.010}$	$2.950^{\pm 0.012}$	$9.536^{\pm 0.104}$	$2.773^{\pm 0.114}$

Table 4.3 Quantitative Results: We compare the results of text-to-motion generation between ours and the state-of-the-art diffusion based methods on HumanML3D[18] dataset. Our method achieves better semantic relevance, diversity, and multimodality performances. Indicate best results , indicates second best results.

4.5 Results

4.5.1 Quantitative Analysis

For quantitative evaluations, we adopt the performance metrics used in ReMoDiffuse[64], which include R Precision, Frechet Inception Distance (FID), Multi-Modal Distance, Diversity and Multi-modality. For R Precision and MultiModality, higher scores indicate superior performance. Conversely, lower scores are preferred for FID and Multi-Modal Distance. For Diversity, performance improves as the score more closely aligns with the real motions.

(1) R Precision evaluates how well the generated motion sequences semantically match the text descriptions. We calculate Top-1, Top-2, and Top-3 accuracy by measuring the Euclidean distance between each motion sequence and 32 text descriptions (one ground truth and 31 randomly selected descriptions). (2) FID calculates the distance between features extracted from real and generated motion sequences. (3) Multi-modal distance (MM Dist for short) measures the average Euclidean distance between the feature vectors of text and generated motions. (4) Diversity measures the variability and richness of the generated sequences. Average euclidean distance is calculated between the two-equal sized subsets which are randomly sampled from the generated motions from all test texts. (5) Multimodality measures the diversity of generated motions for a given text. We generate 10 pairs of motions for each text and compute average distance between the each pair feature vectors.

Table. 4.3 summarizes the results of using our framework in comparison with the existing diffusion-based motion generation models. Incorporating part-specific motion retrieval models as additional knowledge in the motion generation pipeline shows an improvement over Diversity, Multi-Modal Distance, and MultiModality metrics. As observed in MUGL[33], quality scores such as FID based on

feature representations often fail to capture the key action dynamics of the motion sequences. We empirically observed that these scores correlate poorly with the visual quality of motion generations.

Challenges in Motion Retrieval Metrics: To evaluate text-to-motion retrieval methods such as TMR[39] and TMR++[7], the similarity between the text corresponding to the retrieved motion sample and the input text is computed. However, for our approach, which involves the spatial composition of motion sequences, there is no single corresponding text for the entire composed sequence. As a result, these metrics cannot be computed in our case.

4.5.2 Qualitative Analysis

Fig. 4.9 presents qualitative comparisons of MoRAG for retrieval task and Fig. 4.10 presents qualitative results for generation task. We compare our retrieval results with TMR++[7], a state-of-the-art motion retrieval model, and our generation results with ReMoDiffuse[64], focusing on generalizability, zero-shot performance, and diversity.

4.5.2.1 Generalizability

We observe that MoRAG constructed samples, utilizing part-specific retrieved motions, exhibit superior generalizability across the language space and effectively adapt to low-level changes. The richer and dedicated part-specific descriptions generated by the LLM helped retrieve precise part-specific motion sequences corresponding to the input text. Multi-part fusion has improved the construction of motion samples for unseen text descriptions, achieving better semantic alignment with the input text. Current motion retrieval methods treat the input skeleton in a monolithic manner, processing all joints in the pose tree as a whole. However, these approaches overlook the significance of sub-parts, which could enhance generalizability for unseen text descriptions. By dividing each action into sub-actions corresponding to specific subsets of joints, retrieval from the database can be made more effective. For example, in Fig. 4.9 (a), the text phrase, *"A person is eating while seated on the ground"*, doesn't exist in the database which leads to the retrieval of the closest matched sample by TMR++[7], *"this person is sitting on the floor and reaches to his head with his right arm."*. However, MoRAG searches for the part-specific sub-actions "eating" and "sitting on the ground," which can be easily retrieved from the database. When composed, these sub-actions construct a relevant motion sequence.

4.5.2.2 Diversity

MoRAG constructs diverse set of motion samples for a given input text `text` by utilizing both, (1) LLMs' ability to generate diverse text descriptions for a prompt and (2) various combinations of retrieved part-specific motion samples. The diverse samples produced by MoRAG improve the diversity of MoRAG-Diffuse.

4.5.2.3 Zero-Shot Performance

Our approach facilitates the construction of motion sequences for unseen text phrases (zero-shot). This capability arises from two key aspects of the MoRAG framework: (1) it utilizes LLM-generated descriptions rather than the input text directly, and (2) it employs part-wise spatial composition. While the input text may be novel, the LLM-generated part-specific descriptions are not entirely unknown; relevant samples exist where the desired action is performed by specific body parts. Our method retrieves such samples independently for each body part and integrates them to construct motion sequences for unseen text descriptions.

Conditioning the samples constructed by MoRAG facilitates the generation of motion sequences for previously unseen text phrases in MoRAG-Diffuse.

4.5.3 Assumptions and Limitations

- *GPT Dependency:* Our approach leverages GPT-generated descriptions for motion retrieval, but its effectiveness depends on the precision of these descriptions. Semantic deviations can occasionally impact retrieval accuracy. Incorporating feedback mechanisms or quality checks could improve robustness and minimize retrieval discrepancies.

- *Limited Dataset:* Due to the constrained dataset size, some retrievals may be suboptimal, particularly for unique or complex inputs where the existing data may not fully represent the needed diversity. Expanding the dataset or supplementing it with additional sources could improve both retrieval diversity and precision.

- *Composition Alignment:* During composition, retrieved motions are trimmed to the shortest sequence for alignment, which may cause slight misalignments in semantic or temporal coherence. Adding a preliminary coherence check could improve alignment with the input text, enhancing fidelity and consistency.

4.6 Conclusion

In this paper, we enhance the performance of the diffusion-based motion generation model using a multi-part fusion-based retrieval augmented motion generation framework, MoRAG. Our method incorporates additional guidance into the motion generation pipeline using the diverse motion sequences constructed from the samples retrieved from part-specific motion databases. We propose a simple solution to construct multiple diverse, semantically coherent motion samples from the database, even for unseen descriptions, which is not feasible with existing full-body motion retrieval approaches. This makes MoRAG a more viable alternative for text-to-human motion generation by combining the strengths of both retrieval models — rapid motion sample construction and generative models — the ability to create novel outputs.

Future work could extend our approach to other architecture-based generative models. Incorporating additional body part information, such as fingers, head, and lip movements from respective part-specific databases, would enhance the realism of generated samples and better handle more complex language descriptions. Furthermore, our approach can be applied to create new data samples, useful for both training and guiding motion generation models, thereby expanding its potential to handle unusual inputs.

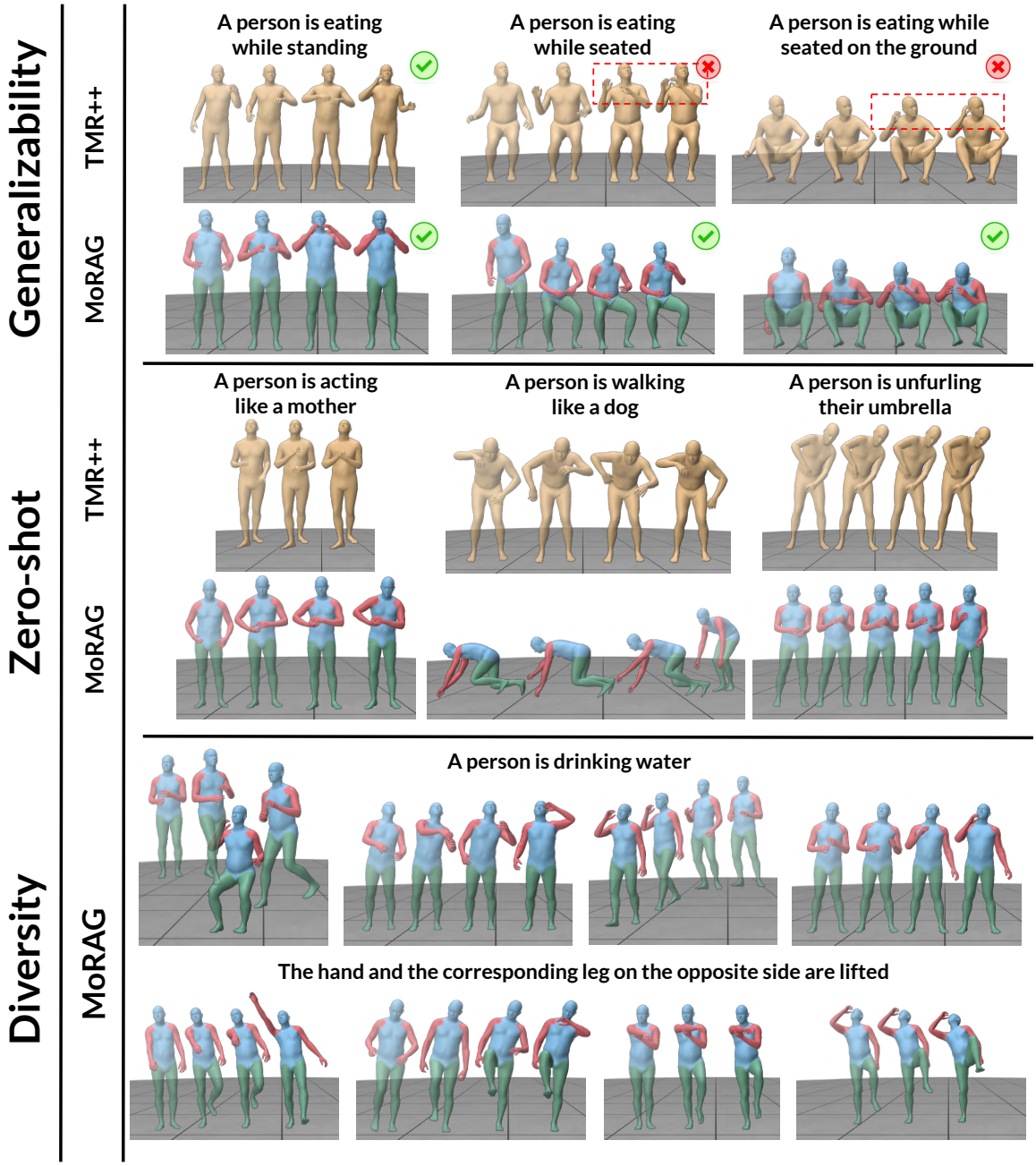


Figure 4.9 Qualitative Results - Retrieval: Comparison of motion retrieval using our multi-part fusion approach with TMR++ [7], a state-of-the-art motion retrieval method. **Top**: Our method demonstrates superior generalization capabilities. **Middle**: Our approach generates accurate motion sequences for unseen text descriptions. **Bottom**: Our setup exhibits increased diversity.

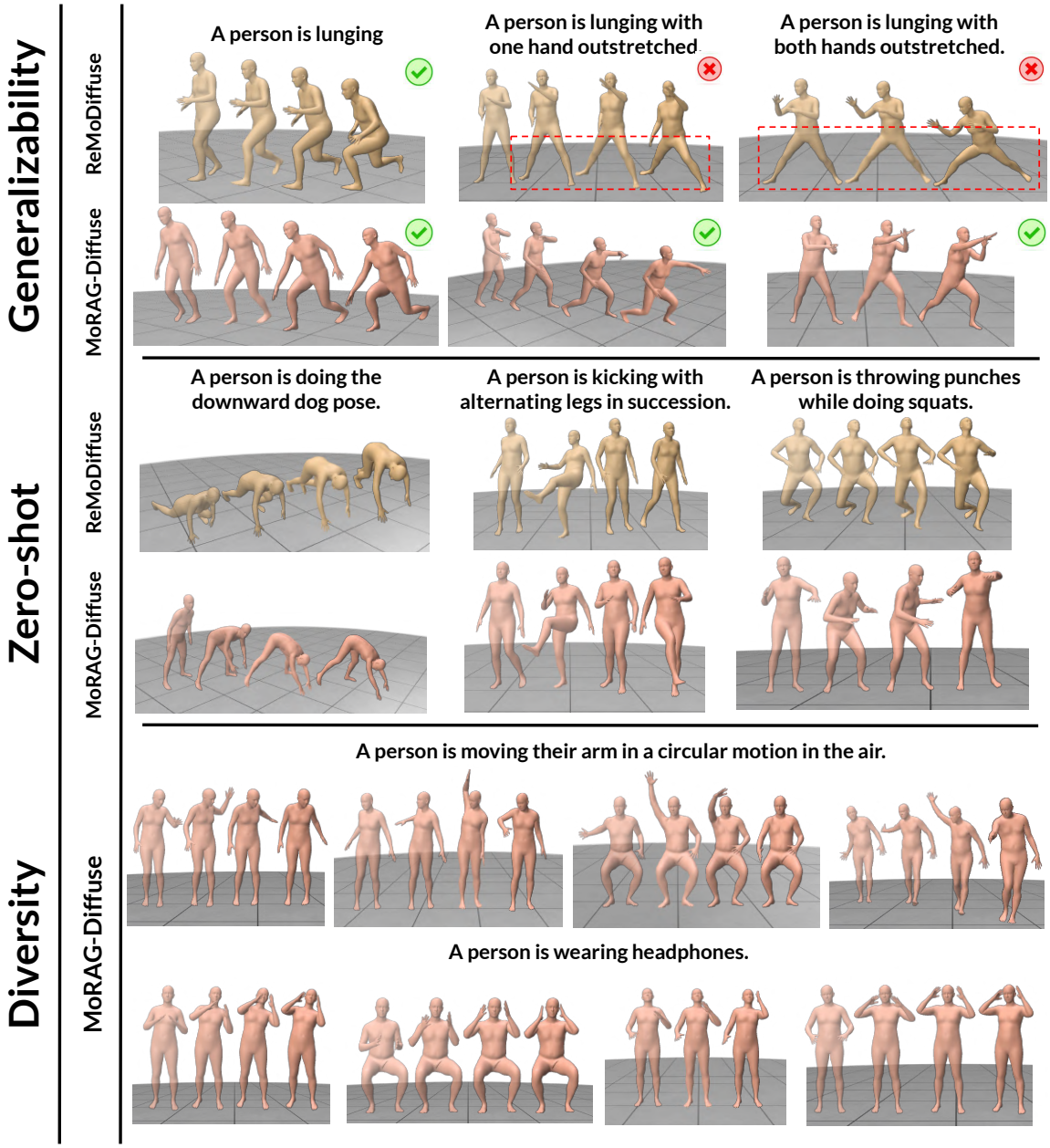


Figure 4.10 Qualitative Results - Generation: Comparison of motion generation using our multi-part fusion approach with ReMoDiffuse [64]. **Top**: Our method demonstrates superior generalization capabilities. **Middle**: Our approach generates accurate motion sequences for unseen text descriptions. **Bottom**: Our setup exhibits increased diversity.

Chapter 5

Conclusions

This thesis presents novel methods for incorporating large language models (LLMs) into human motion generation and retrieval tasks. We began by identifying key limitations in current motion generation models, particularly their poor performance with complex and unseen text inputs. These models struggle to capture fine-grained details, largely due to the minimalist nature of existing text-annotated motion datasets. The text descriptions found in these datasets tend to be short and lacking in information about specific body movements related to the described actions. This scarcity of detail limits the models' ability to generalize, leading to difficulties in handling complex inputs and in producing detailed motion representations.

To address these challenges, we introduced **Action-GPT**, the first method to integrate LLMs into text-conditioned motion generation. Instead of relying on the minimal text annotations from existing datasets, Action-GPT uses LLM-generated prompts to enrich the text input within the generation pipeline. This approach improves the alignment between text and motion latent spaces, resulting in enhanced model generalizability. Through prompt analysis, we demonstrated the suitability of our method for producing more fine-grained motion details. We demonstrated the advantage of using multiple LLM-generated text descriptions rather than relying on a single one, taking into account the inherent variability in LLM outputs. The diversity in LLM-generated text descriptions enabled the generation of a wider variety of motion samples. Additionally, the model exhibited improved performance in zero-shot scenarios, a challenge that previous methods were unable to overcome.

With the growing adoption of diffusion models in generative tasks, motion generation has also transitioned to such frameworks. However, existing diffusion-based models struggle to generate multiple relevant motion samples from a given text input. Inspired by the rise of retrieval-augmented generation (RAG) approaches in the text and image domains, we developed **MoRAG**, a LLM-based Multi-Fusion Retrieval-Augmented Generation method. Our analysis revealed that traditional motion retrieval methods treat the skeleton monolithically, retrieving entire motions at once. This approach struggles with complex or unseen text inputs due to the limited size of current text-annotated motion datasets. However, these datasets contain extensive low-level information about body parts, which holds potential for broader generalization across diverse language inputs. Building on this observation, we proposed a part-

based retrieval method that retrieves specific motion samples for different body parts—torso, hands, and legs—based on LLM-generated part-specific descriptions. These part-specific motions are then fused to construct coherent and diverse motion samples relevant to the input text. Incorporating these motion samples into the diffusion model generation pipeline resulted in the production of more diverse and precise motion sequences, as demonstrated by both qualitative and quantitative analyses.

Our findings can be found explored at https://actiongpt.github.io/ and https://motion-rag.github.io/. The websites contains generated motion sequence examples, code and pre-trained models.

Related Publications

Thesis Publications

- **Sai Shashank**, Shubh Maheswari, Ravi Kiran Sarvadevabhatla. *Action-GPT: Leveraging Large-scale Language Models for Improved and Generalized Action Generation*; **Proceedings of the IEEE International Conference on Multimedia and Expo (ICME) 2023**.

- **Sai Shashank**, Shubh Maheswari, Ravi Kiran Sarvadevabhatla. *MoRAG - Multi-Fusion Retrieval Augmented Generation for Human Motion*; **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2025**.

Other Publications

- Debtanu Gupta, Shubh Maheswari, **Sai Shashank**, Manasvi Vaidyula, Ravi Kiran Sarvadevabhatla; *DSAG: A Scalable Deep Framework for Action-Conditioned Multi-Actor Full Body Motion Synthesis*. **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2023**.

Bibliography

[1] H. Ahn, T. Ha, et al. Text2action: Generative adversarial synthesis from language to action. In *ICRA*, pages 5915–5920, 2018. 1, 5, 16, 33

[2] C. Ahuja and L. Morency. Language2pose: Natural language grounded pose forecasting. In *3DV*, 2019. 5, 16, 30, 33

[3] N. Athanasiou, A. Ceske, M. Diomataris, M. J. Black, and G. Varol. Motionfix: Text-driven 3d human motion editing, 2024. 1

[4] N. Athanasiou, M. Petrovich, M. J. Black, and G. Varol. TEACH: Temporal Action Compositions for 3D Humans. In *International Conference on 3D Vision (3DV)*, 2022. xii, xiii, 1, 5, 15, 16, 17, 20, 21, 23, 24, 33

[5] N. Athanasiou, M. Petrovich, M. J. Black, and G. Varol. SINC: Spatial composition of 3D human motions for simultaneous action generation. *ICCV*, 2023. 40

[6] S. Azadi, A. Shah, T. Hayes, D. Parikh, and S. Gupta. Make-an-animation: Large-scale text-conditional 3d human motion generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15039–15048, October 2023. 1

[7] L. Bensabath, M. Petrovich, and G. Varol. Tmr++: A cross-dataset study for text-based 3d human motion retrieval. 2024. xiv, 6, 31, 34, 35, 45, 48

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 15, 18

[9] Z. Cen, H. Pi, S. Peng, Z. Shen, M. Yang, Z. Shuai, H. Bao, and X. Zhou. Generating human motion in 3d scenes from text descriptions. In *CVPR*, 2024. 1

[10] W. Chen, H. Hu, C. Saharia, and W. W. Cohen. Re-imagen: Retrieval-augmented text-to-image generator, 2022. 33

[11] X. Chen, B. Jiang, W. Liu, Z. Huang, B. Fu, T. Chen, and G. Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023. 5, 30, 33, 44

[12] C. Deng, Q. Chen, P. Qin, D. Chen, and Q. Wu. Prompt switch: Efficient clip adaptation for text-video retrieval, 2023. 33

[13] Y. Feng, J. Lin, S. K. Dwivedi, Y. Sun, P. Patel, and M. J. Black. ChatPose: Chatting about 3d human pose. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2024. 1

[14] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang. Retrieval-augmented generation for large language models: A survey, 2024. 31

[15] Z. Gao, J. Liu, W. Sun, S. Chen, D. Chang, and L. Zhao. Clip2tv: Align, match and distill for video-text retrieval, 2022. 33

[16] A. Ghosh, N. Cheema, C. Oguz, C. Theobalt, and P. Slusallek. Synthesis of compositional animations from textual descriptions. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2021. 5, 16, 30, 33

[17] E. Grossman, M. Donnelly, R. Price, D. Pickens, V. Morgan, G. Neighbor, and R. Blake. Brain areas involved in perception of biological motion. *Journal of Cognitive Neuroscience*, 12(5):711–720, 2000. 4

[18] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, June 2022. xiii, xv, 1, 5, 11, 30, 32, 34, 35, 41, 42, 43, 44

[19] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. 43

[20] D. Gupta, S. Maheshwari, S. S. Kalakonda, Manasvi, and R. K. Sarvadevabhatla. Dsag: A scalable deep framework for action-conditioned multi-actor full body motion synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023. 15, 35

[21] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *ACM Transactions on Graphics (TOG)*, 41(4):1–19, 2022. 5

[22] Y. Ji, F. Xu, Y. Yang, F. Shen, H. T. Shen, and W.-S. Zheng. A large-scale varying-view rgb-d action dataset for arbitrary-view human action recognition. *arXiv preprint arXiv:1904.10681*, 2019. 5

[23] B. Jiang, X. Chen, C. Zhang, F. Yin, Z. Li, G. YU, and J. Fan. Motionchain: Conversational motion controllers via multimodal prompts, 2024. 1

[24] S. S. Kalakonda, S. Maheshwari, and R. K. Sarvadevabhatla. Action-gpt: Leveraging large-scale language models for improved and generalized action generation. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2023. 30, 31

[25] J. Kim, H. Oh, S. Kim, H. Tong, and S. Lee. A brand new dance partner: Music-conditioned pluralistic dancing controlled by multiple dance genres. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3490–3500, June 2022. 1

[26] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *arXiv*, 2019. 38

[27] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 33

[28] J. Liu, A. Shahroudy, et al. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *TPAMI*, 42(10):2684–2701, 2020. 5

[29] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021. 15, 18

[30] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. xi, 10, 17, 41

[31] I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019. 43

[32] T. Lucas*, F. Baradel*, P. Weinzaepfel, and G. Rogez. Posegpt: Quantization-based 3d human motion generation and forecasting. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[33] S. Maheshwari, D. Gupta, and R. K. Sarvadevabhatla. Mugl: Large scale multi person conditional action generation with locomotion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 257–265, January 2022. 9, 15, 44

[34] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019. 5, 34, 35, 40, 43

[35] D. Pavllo, D. Grangier, et al. Quaternet: A quaternion-based recurrent model for human motion. In *BMVC*, 2018. 8

[36] M. Petrovich, M. J. Black, and G. Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *ICCV*, 2021. 9

[37] M. Petrovich, M. J. Black, and G. Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, 2021. 14

[38] M. Petrovich, M. J. Black, and G. Varol. TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision (ECCV)*, 2022. xii, 1, 5, 15, 16, 17, 20, 22, 23, 33, 34, 37

[39] M. Petrovich, M. J. Black, and G. Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *International Conference on Computer Vision (ICCV)*, 2023. xiv, 6, 31, 34, 35, 37, 39, 40, 43, 45

[40] M. Petrovich, O. Litany, U. Iqbal, M. J. Black, G. Varol, X. B. Peng, and D. Rempe. Multi-track timeline control for text-driven 3d human motion generation. In *CVPR Workshop on Human Motion Generation*, Seattle, June 2024. 1

[41] S. Pratt, R. Liu, and A. Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification, 2022. 15

[42] A. R. Punnakkal, A. Chandrasekaran, N. Athanasiou, A. Quiros-Ramirez, and M. J. Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 722–731, June 2021. xi, xiii, xv, 5, 14, 15, 20, 21, 23, 25, 26, 34, 35

[43] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 33

[44] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. 5, 6, 16, 31, 33, 34

[45] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019. 22

[46] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. 37

[47] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2019. cite arxiv:1909.08053. 15

[48] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu. Mpnet: Masked and permuted pre-training for language understanding, 2020. 39

[49] G. Tevet, B. Gordon, A. Hertz, A. H. Bermano, and D. Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 358–374. Springer, 2022. xii, 1, 5, 15, 16, 17, 20, 21, 22, 33

[50] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-or, and A. H. Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. 30, 33, 44

[51] N. Trivedi and R. K. Sarvadevabhatla. Psumnet: Unified modality part streams are all you need for efficient pose-based action recognition. In *European Conference on Computer Vision*, pages 211–227. Springer, 2022. 35

[52] N. F. Troje. A framework for analysis and synthesis of human gait patterns. 2002. 4

[53] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding, 2019. 39

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. 11

[55] Q. Wu, Y. Zhao, Y. Wang, Y.-W. Tai, and C.-K. Tang. Motionllm: Multimodal motion-language learning with large language models, 2024. 1

[56] W. Xiang, C. Li, Y. Zhou, B. Wang, and L. Zhang. Language supervised training for skeleton-based action recognition, 2022. 15

[57] H. Yi, J. Thies, M. J. Black, X. B. Peng, and D. Rempe. Generating human interaction motions in scenes with text control. *arXiv:2404.10685*, 2024. 1

[58] K. Yin, S. Zou, Y. Ge, and Z. Tian. Tri-modal motion retrieval by learning a joint embedding space, 2024. 31, 34, 35

[59] K. M. Yoo, D. Park, J. Kang, S.-W. Lee, and W. Park. Gpt3mix: Leveraging large-scale language models for text augmentation, 2021. 15

[60] T. Yoshida, A. Masumori, and T. Ikegami. From text to motion: Grounding gpt-4 in a humanoid robot "alter3", 2023. 1

[61] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models, 2022. 33

[62] J. Zhang, Y. Zhang, X. Cun, S. Huang, Y. Zhang, H. Zhao, H. Lu, and X. Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5, 33

[63] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 5, 30, 33, 44

[64] M. Zhang, X. Guo, L. Pan, Z. Cai, F. Hong, H. Li, L. Yang, and Z. Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 364–373, October 2023. xiii, xiv, 5, 30, 31, 32, 33, 41, 42, 43, 44, 45, 49

[65] M. Zhang, H. Li, Z. Cai, J. Ren, L. Yang, and Z. Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. *NeurIPS*, 2023. 30, 44

[66] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui. Retrieval-augmented generation for ai-generated content: A survey, 2024. 31

[67] Y. Zhou, C. Barnes, L. Jingwan, Y. Jimei, and L. Hao. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 17

[68] W. Zhu, X. Ma, D. Ro, H. Ci, J. Zhang, J. Shi, F. Gao, Q. Tian, and Y. Wang. Human motion generation: A survey, 2023. 4, 30

[69] S. Zou, X. Zuo, Y. Qian, S. Wang, C. Xu, M. Gong, and L. Cheng. 3d human shape reconstruction from a polarization image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 5