FreeMotion: MoCap-Free Human Motion Synthesis with Multimodal Large Language Models

Zhikai Zhang[1,3], Yitang Li[1,3], Haofeng Huang[1], Mingxian Lin[3], and Li Yi[1,2,3]

[1] Tsinghua University [2] Shanghai AI Laboratory [3] Shanghai Qi Zhi Institute

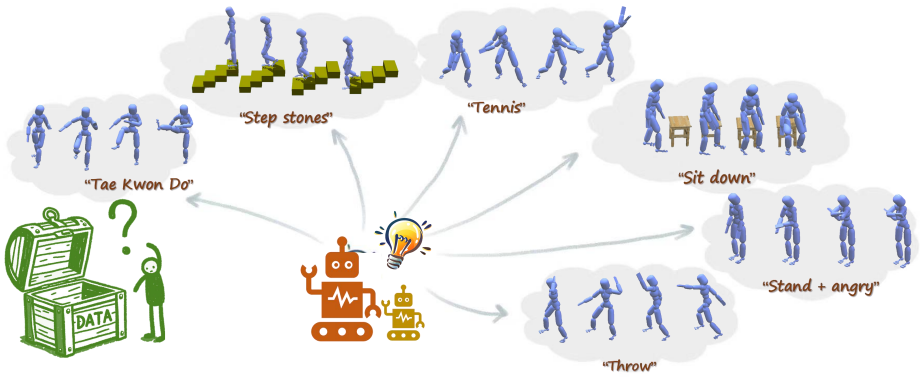

Fig. 1: Our method for the first time, without any motion data, explores open-set human motion synthesis using natural language instructions as user control signals based on MLLMs across any motion task and environment.

Abstract. Human motion synthesis is a fundamental task in computer animation. Despite recent progress in this field utilizing deep learning and motion capture data, existing methods are always limited to specific motion categories, environments, and styles. This poor generalizability can be partially attributed to the difficulty and expense of collecting large-scale and high-quality motion data. At the same time, foundation models trained with internet-scale image and text data have demonstrated surprising world knowledge and reasoning ability for various downstream tasks. Utilizing these foundation models may help with human motion synthesis, which some recent works have superficially explored. However, these methods didn't fully unveil the foundation models' potential for this task and only support several simple actions and environments. In this paper, we for the first time, *without any motion data*, explore open-set human motion synthesis using natural language instructions as user control signals based on MLLMs across any motion task and environment. Our framework can be split into two stages: 1) sequential keyframe generation by utilizing MLLMs as a keyframe designer and animator; 2) motion filling between keyframes through interpolation and

motion tracking. Our method can achieve general human motion synthesis for many downstream tasks. The promising results demonstrate the worth of mocap-free human motion synthesis aided by MLLMs and pave the way for future research.

Keywords: Human motion synthesis · Multimodal large language models · Physics-based character animation

1 Introduction

Synthesizing humanoid movements and interactions is a cornerstone for advancing embodied AI, enhancing the realism of video games, enriching experiences in VR/AR, and empowering robots with the ability to interact with humans. Therefore, researchers have been long seeking automatic ways for humanoid animation synthesis. Existing works [11, 13, 14, 19, 20, 36, 37, 41, 45, 48, 49] have made significant progress facilitated by reference motion trajectories depicting real human movements collected through motion capture (mocap) systems. While these methods have yielded high-fidelity animations, they are intrinsically limited by the scope of the mocap data. Due to the inherent difficulty and expenses of motion capturing, the largest publicly accessible mocap datasets [11, 25] only encompass dozens of hours of motion, which is still far from enough to cover the vast array of daily human motions. As such, data-driven animation synthesis is usually confined to pre-recorded motion datasets and lacks open-set generalizability to novel environments and unseen human behaviors.

In the realm of machine learning, Multimodal Large Language Models (MLLMs) have recently emerged as a transformative force, showcasing remarkable competency in inferring and adapting to open-set scenarios. These models have been successful across a spectrum of tasks that range from perception [1, 8, 44] and high-level planning [17, 18] to low-level manipulative actions [24, 46]. This success prompts us to consider whether we can leverage powerful MLLMs trained on internet-scale image and text data (e.g., GPT-4V [1]) to break free from the dependency on mocap data and instead generate open-set humanoid animations that can dynamically adapt to new and ever-changing environments and tasks.

In this work, we for the first time demonstrate MLLMs' ability for open-set humanoid motion synthesis controlled by natural language user input *without any motion data*. (See Fig. 1.) Directly applying MLLMs trained on image and text data as motion generators is not proper, as they may not capture the subtleties of continuous motion necessary for realistic character animation. Nonetheless, MLLMs excel in understanding high-level action narratives and keyframes, akin to a lead animator's role in traditional studios. We, therefore, propose to first leverage MLLMs to decompose open-set humanoid motion into narrative plots and corresponding keyframes and then in the second stage develop automatic motion filling algorithms to close the gap between the discrete understanding of MLLMs and the continuous nature of humanoid movement.

Specifically, in the first stage, we employ two specialized GPT-4V agents to generate a sequence of keyframes. One agent acts as the keyframe designer, using

text descriptions of the desired motion (e.g., walking) and the current state of the humanoid (e.g., the left leg advancing while the right leg is stationary), along with a rendered image of the humanoid, to predict the text for the subsequent keyframe (e.g., the left leg making contact with the ground as the right leg begins to lift) and the time interval between two adjacent frames. The other agent, the keyframe animator, is then presented with this predicted text. With a set of pre-defined commands to manipulate the humanoid's joints and the current state information, the animator selects appropriate commands to adjust the humanoid's pose to match the designer's description, using the rendered images for visual feedback. This pose adjustment may be refined multiple times. The designer and animator collaborate in this iterative fashion until they complete the motion sequence.

In the second stage, to transform a sequence of keyframes into a fluid motion clip, we engage in motion filling. Initially, we perform interpolation on the keyframe sequence to create a time-continuous motion clip. However, since interpolation may not adhere to physical laws, we utilize a motion tracking policy that corrects for physically implausible poses and transitions. Drawing inspiration from successful model-based tracking methods [9, 40, 45], we develop a CVAE-based policy empowered by an MLP-based world model to track the interpolated motion. Unlike previous approaches confined to flat terrain, we integrate height maps to inform our policy and world model about varying terrain, ensuring our synthesis is adaptable to diverse environments.

We evaluate our method on a wide range of downstream tasks, including motion synthesis, style transfer, human-scene interaction, and stepping stones. Our method achieves surprising results *without any motion data*.

2 Related Work

2.1 Foundation Models for Motion Synthesis

Striking advancements of foundation models [1,3,4,7,8,26,30,35] have been made during the past few years. The success of LLMs stimulated interest in MLLMs (Multimodal Large Language Models) [1, 35], which extends LLMs to accept visual input. They either learn from visual signal and text simultaneously from scratch [35] or employ a cross-modal connector to align the features of visual encoders to the LLM's text embedding space. As foundation models demonstrate impressive capabilities on many downstream applications these years, researchers try to ground their knowledge into motion synthesis tasks. Generating reward functions for particular tasks is adopted by several works [23, 24, 46] as an intermediate interface connecting motion instructions and physics-based motion controllers. Methods based on reward design utilize GPT's ability of logical reasoning and code generation. However, only a small set of motions is suitable to be represented as a reward function. These methods fail to maintain their performance when applied to open-set motion synthesis. Rather than designing task-specific reward functions, [33] utilizes CLIP [30] to compute the similarity between observation and motion text, which serves as the reward value for policy training. It only supports the simplest human motions (e.g. sitting, raising

hands). Compared with existing methods, our method takes the first step toward open-set motion synthesis based on MLLMs.

2.2 Human Motion Synthesis

Human motion synthesis is a fundamental task in computer animation. With the popularity of neural networks and motion capture data, data-driven methods have become mainstream [11, 13, 14, 19, 20, 36–39, 41, 43, 45, 48–51]. Recent researchers use generative models to recover kinematic motions from Gaussian noise, given various conditional signals. VAE-based methods [10, 11, 15, 29] and GAN-based [14, 21, 22] methods have been widely explored during the past few years. [15] employs a Variational Autoencoder (VAE) to acquire a general motion manifold, enabling the synthesis and editing of character motion based on high-level control parameters. Also, employing diffusion models in motion synthesis [31,37,39,43,47,48] has emerged as a new trend due to their state-of-the-art performance. Several recent studies have explored novel approaches in motion control and synthesis. [27] distills a large set of expert policies into a latent space for a high-level controller. [40] employs a conditional VAE for expert demonstration mimicry, while [45] uses a CVAE for flexible skill representation and policy learning. Inspired by GAIL, [28] develops a discriminator to ensure style consistency and task-specific reward but still compliance in motion data.

3 Method

One straightforward way to generate human motion clips from MLLMs is to utilize MLLMs as motion state predictors. However, this method often yields unsatisfactory results due to the underrepresentation of such motion data in the MLLMs' training corpora and the subtleties of continuous motion. MLLMs excel in their world knowledge and logical reasoning ability drawn from internet-scale text and image data. However, these abilities only come into play in high-level semantic space rather than low-level motion space. A significant challenge is bridging this gap and effectively applying the MLLMs' capabilities to motion space. To solve this problem, we propose our framework, FreeMotion, and split the problem into two stages: 1) sequential keyframe generation by utilizing MLLMs as a designer and animator; 2) motion filling between keyframes through interpolation and motion tracking. The underlying insight of our method is the utilization of MLLMs solely within the high-level semantic space. Since keyframes in a motion usually contain richer and more salient semantic information, we utilize MLLMs to decompose a given motion temporally and spatially by generating sequential keyframes. The blank between keyframes is left for motion-filling techniques, including interpolation and environment-aware motion tracking. The overview of our method is shown in Fig. 2.

3.1 Sequential Keyframe Generation from MLLMs

Given a user instruction requiring a specific motion, we hope MLLMs can translate it into a sequence of humanoid poses, each representing a keyframe in the motion. Such a task involves motion understanding, keyframe reasoning, and

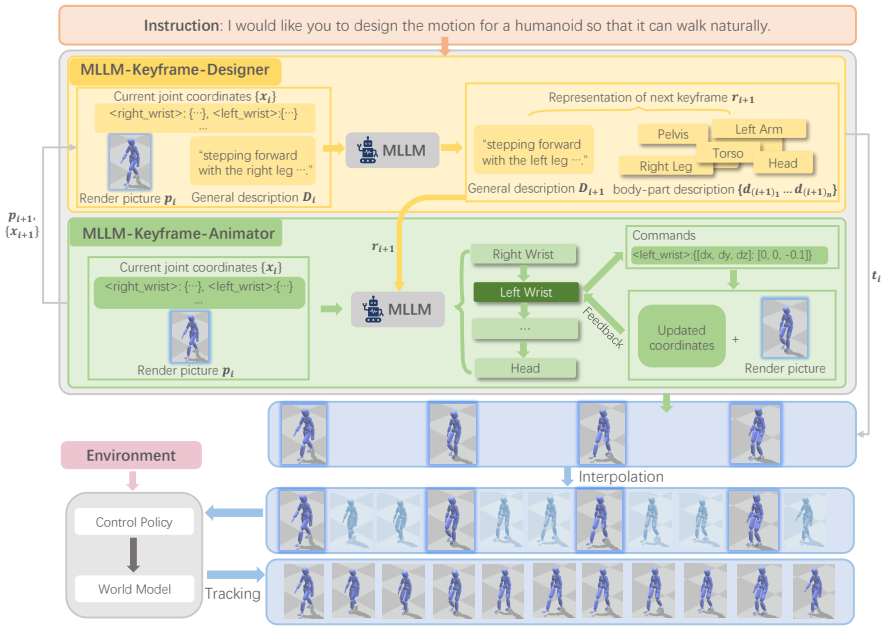

Fig. 2: Overview of FreeMotion. FreeMotion adopts two specialized GPT-4V agents for sequential keyframe generation. Then we utilize interpolation and environment-aware motion tracking to fill the blank between keyframes.

humanoid posture adjustment. As it is challenging for the MLLM to output a correct sequence simultaneously, we employ two specialized GPT-4V agents, each playing a distinct role. One serves as a keyframe designer, aiming to translate the input motion instruction into relatively low-level body part descriptions of sequential keyframes. The other one acts as a keyframe animator, who takes the description of one keyframe generated by the designer and fits a humanoid's pose to the description through visual feedback using a set of pre-defined pose adjustment commands. We will discuss the details of each GPT-4V agent in the following sections.

Keyframe Designer. The keyframe designer's role is to translate high-level motion instruction $\boldsymbol{I}$ into a sequence of more detailed, low-level keyframe representation $\boldsymbol{R} = \{\boldsymbol{r_1}, \ldots, \boldsymbol{r_m}\}$, where m is the number of keyframes to represent the motion. Each $\boldsymbol{r_i}$ is composed of a general full-body description $\boldsymbol{D_i}$ (e.g., *"The humanoid is stepping forward with its left leg, the right leg is stationary and the arms are swinging opposite to the legs."*) and a series of body-part (e.g., left arm, right leg, torso) descriptions $\{\boldsymbol{d_{i_1}}, \ldots, \boldsymbol{d_{i_n}}\}$ (e.g., *"The left arm is moving backwards in a smooth arc, with the shoulder back, the elbow slightly bent, and the hand relaxed."*), where n is the number of body parts. Each time, the keyframe designer depicts the next keyframe given the full-body description $\boldsymbol{D_i}$, a rendered picture $\boldsymbol{p_i}$ of the current humanoid, the humanoid's current joint coordinates $\{\boldsymbol{x_i}\}$, and the motion instruction $\boldsymbol{I}$ as input, outputting: 1) the low-

level keyframe representation of the next keyframe $\boldsymbol{r_{i+1}}$; 2) the time interval $\boldsymbol{t_i}$ (e.g., *"0.5s"*) between the current state and the predicted next keyframe. Starting with $\boldsymbol{D_0}$ (*"The humanoid is standing on the ground."*), the keyframe designer can produce the whole sequence of keyframe representation $\boldsymbol{R}$. This process spatially and temporally decomposes the high-level motion instruction $\boldsymbol{I}$, integrating the MLLM's knowledge into a more tangible motion representation for the subsequent keyframe animator. The MLLM can make a reasonable motion-to-keyframe decomposition without the rendered picture $\boldsymbol{p_i}$ of the humanoid. But the presentation of $\boldsymbol{p_i}$ offers the keyframe designer a chance to better understand the humanoid's current state and make an improved representation $\boldsymbol{r_{i+1}}$ of the next keyframe.

At this stage, the MLLM plays a crucial role in determining the spacing between adjacent keyframes. Excessively distant keyframes can lead to unstable results and result in motion artifacts even with physics correction. Conversely, overly close keyframes can make the generation process excessively tedious, diminishing the keyframes' ability to provide constructive guidance. However, in most of our experiments, GPT-4V successfully generates feasible keyframes. This success could be attributed to the MLLM's inherent understanding of motion dynamics; it comprehends that a motion sequence is comprised of several distinct stages. For instance, in walking, the sequence involves lifting the left foot, stepping forward, setting down the left foot, and lifting the right foot. Given this understanding, the MLLM is able to generate an appropriate number of keyframes, effectively segmenting the entire motion sequence.

We let the keyframe designer to automatically determine the termination point of motion design. It is instructed to signal the completion of the entire motion sequence by outputting "Done" once it perceives the completion of the specified non-periodic motion, or believes that a periodic motion has concluded after a full cycle. Besides leveraging the MLLM's capability to recognize motion termination, we also manually set an upper limit to ensure the motion design won't be endless.

Keyframe Animator. Provided with a detailed next-keyframe representation $\boldsymbol{r_{i+1}}$, joint coordinates $\{\boldsymbol{x_i}\}$, and the rendered picture $\boldsymbol{p_i}$, the keyframe animator is responsible for adjusting a humanoid's pose $\boldsymbol{s_i^k}$ to fit the representation $\boldsymbol{r_{i+1}}$. Adjustments are made in order of body parts listed by the keyframe designer. Rather than directly tuning the joint's position or rotation, we regularize the adjustment as a set of commands, each corresponding to a specific joint movement. These commands are implemented using kinematic methods, such as forward and inverse kinematics. This regularization not only frees MLLM from the tedium of tuning spatial features joint by joint but also gives semantic information to pose adjustments so that the reasoning ability can be utilized. All commands used by GPT-4V are listed in Tab. 1. We also allow the GPT-4V animator to rotate the camera around the humanoid to observe body parts of interest better.

Despite the simplifications that have been made, it remains a complex task for the MLLM to accurately adjust the pose in a single attempt. Luckily, vi-

sual signals can serve as feedback and make multi-iteration adjustment possible. Concretely, given a certain body part and the representation $\boldsymbol{r_{i+1}}$, the MLLM chooses one of the commands and outputs corresponding parameters to adjust the body part. When the command is executed on the humanoid, the updated joint coordinates and the rendered picture are passed to the keyframe animator as feedback. This loop continues until the animator believes the body part's pose aligns with the descriptions or the times of this body part's adjustment meet the upper limit, which is 5 in our method. When the adjustment is finished for one body part, the animator switches to the next body part according to a pre-defined order. The animator finally passes the updated coordinates $\{\boldsymbol{x_{i+1}}\}$ and an updated rendered picture $\boldsymbol{p_{i+1}}$ for the next keyframe back to the designer when the adjustment for every body part is completed.

It is worthing to note that although our method incorporates a visual feedback mechanism, it typically converge within the upper limit for a single body part. This is primarily because most body parts either remain static or experience only minor alterations during transitions. Consequently, the total number of adjustments required by the animator to transition the humanoid from $\boldsymbol{s_i^k}$ to $\boldsymbol{s_{i+1}^k}$ consistently remains under 10.

Table 1: Command Set. We regularize the pose adjustment as a set of commands.

Command	Function
Single joint movement	move a selected joint around its parent joint to a target place
End effector movement	move a selected end effector quickly to a target place through pre-defined IK chains
Pelvis rotation/movement with support points on the ground	rotate/move the pelvis with one or more support points on the ground through IK
Pelvis rotation/movement without support points on the ground	rotate/move the pelvis without support points on the ground through direct rotation/movement
Single joint roll	roll a selected joint
Camera rotation	rotate the camera around the humanoid

3.2 Motion Filling through Interpolation and Motion Tracking

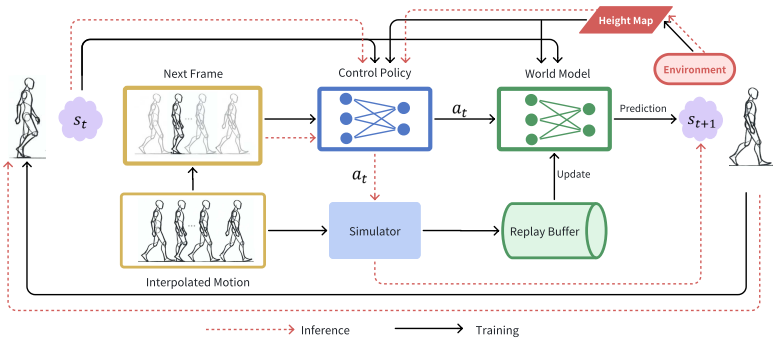

Fig. 3: Policy training and inference. We incorporate height maps as visual signals, enabling our policy and world model to be aware of diverse environmental conditions.

After obtaining a series of keyframes with each keyframe specified by humanoid poses $\{\boldsymbol{s_1^k}, \ldots, \boldsymbol{s_m^k}\}$ and time intervals $\{\boldsymbol{t_1}, \ldots, \boldsymbol{t_{m-1}}\}$ based on our instructions, we perform linear position and rotation interpolation on these keyframes to achieve continuous motion frames, resulting in an interpolated frame rate of 20 frames per second. However, straightforward interpolation may fall short of ensuring the motion's physical validity. To address this, we turn to model-based motion tracking methods, as proven in [32], which successfully navigate the challenges of infeasible state transitions. We implement a refined motion tracking system using a CVAE-based policy combined with an MLP-based world model, drawing inspiration from the methodology of ControlVAE [45]. A notable innovation in our study is the integration of environmental signals, enhancing the model's responsiveness to dynamic contexts. The methodology is depicted in Fig. 3.

Environment Visual Signals Extraction. Our incorporation of height maps as visual signals enables our policy and world model to be cognizant of the environment and thus to be environment-aware. We derive a height map around the humanoid pelvis from the current environment observation and flatten it into a vector $\boldsymbol{o_t}$ at simulation time step t.

CVAE-based Motion Control Policy. Our CVAE-based motion control policy is formulated as a conditional encoder and decoder. The state $\boldsymbol{s}$ at each simulation time step can be fully characterized by $\{\boldsymbol{x_j}, \boldsymbol{q_j}, \boldsymbol{v_j}, \boldsymbol{\omega_j}\}, j \in B$, where B is the set of rigid bodies and $\boldsymbol{x_j}, \boldsymbol{q_j}, \boldsymbol{v_j}, \boldsymbol{\omega_j}$ stand for the position, orientation, linear velocity, and angular velocity of each rigid body, respectively. Given the current state $\boldsymbol{s_t}$ and the interpolated trajectory $\tilde{\tau} = \{\tilde{\boldsymbol{s}}_\mathbf{1}, \ldots, \tilde{\boldsymbol{s}}_\boldsymbol{T}\}$, we first encode the state transition $(\boldsymbol{s_t}, \tilde{\boldsymbol{s}}_{\boldsymbol{t+1}})$ and visual signals $\boldsymbol{o_t}$ into a latent variable $\boldsymbol{z}$. The network for encoding is referred to q_ϕ, parameterized by ϕ, which models the embedding to a Gaussian distribution:

$$q_\phi(\boldsymbol{z_t}|\boldsymbol{s_t}, \tilde{\boldsymbol{s}}_{\boldsymbol{t+1}}, \boldsymbol{o_t}) = \mathcal{N}(\boldsymbol{z_t}; \mu_\phi(\boldsymbol{s_t}, \tilde{\boldsymbol{s}}_{\boldsymbol{t+1}}, \boldsymbol{o_t}), \Sigma_\phi(\boldsymbol{s_t}, \tilde{\boldsymbol{s}}_{\boldsymbol{t+1}}, \boldsymbol{o_t})). \tag{1}$$

Using the latent variable derived from the previous network, we can generate an action by the decoder. Our decoder can be formulated as a conditional distribution $p(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{z})$ that outputs an action $\boldsymbol{a}$ according to the character's current state $\boldsymbol{s}$ and a latent variable $\boldsymbol{z}$. We model the policy p_θ parameterized by θ as a Gaussian distribution as well:

$$p_\theta(\boldsymbol{a_t}|\boldsymbol{s_t}, \boldsymbol{z_t}) = \mathcal{N}(\boldsymbol{a_t}; \mu_\theta(\boldsymbol{s_t}, \boldsymbol{z_t}), \Sigma_\theta(\boldsymbol{s_t}, \boldsymbol{z_t})). \tag{2}$$

MLP-based World Model We approximate true transition probability distribution $p(\boldsymbol{s_{t+1}}|\boldsymbol{s_t}, \boldsymbol{a_t})$ in the simulator using an environment-aware world model $\omega(\boldsymbol{s_{t+1}}|\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{o_t})$, which is another Gaussian distribution

$$\omega(\boldsymbol{s_{t+1}}|\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{o_t}) \sim \mathcal{N}(\boldsymbol{s_{t+1}}; \mu_\omega(\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{o_t}), \Sigma_\omega(\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{o_t})). \tag{3}$$

Inference At each time step t, with $\boldsymbol{s_t}, \tilde{\boldsymbol{s}}_{\boldsymbol{t+1}}, \boldsymbol{o_t}$ as input, our policy outputs $\boldsymbol{a_t}$ to the simulator for the computation of $\boldsymbol{s_{t+1}}$. Starting with $\boldsymbol{s}_0 = \tilde{\boldsymbol{s}}_0$ and continuously repeating this process, we obtain a trajectory $\tau = \{\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s_{T-1}}, \boldsymbol{s_T}\}$ where the character keeps moving under the guidance of the given interpolated motion frames.

Training Process. Detailed training process and loss terms are complicated and not the focus of our work. We adopt almost the same training process and loss terms as ControlVAE [45]. We recommend readers refer to the original paper for more details. It is worth noting that we do not train a motion tracker for every single generated motion since it's very time-consuming. For each downstream task, which will be presented in the next section, we concatenate all interpolated motions together to train a policy and world model. The collection of each trajectory is conducted within each interpolated motion so that it doesn't span different motions. The length of each interpolated motion must meets the minimum rollout length for successful training. Therefore motions that don't meet the requirement will be padded with its last frame.

4 Tasks

We evaluated our methods on various downstream tasks across different motion categories and environments, including motion synthesis, style transfer, human-scene interaction, and stepping stones. We use ODE [34] for physical simulation.

4.1 Motion Synthesis

In this task, we evaluate our method's performance on motion synthesis. We conducted two experiments. In the first, we compared our method with two recent data-driven methods, MDM [37] and MLD [6] on HumanAct12. In the second, we compared our method to zero-shot motion synthesis methods [16,36], using motions that were unseen by both the baselines and our model for testing.

Baseline MDM [37] and MLD [6] are recent data-driven methods trained on HumanAct12 [12], which is an action-to-motion dataset, containing 12 action categories and 1191 motion clips. It is worth noting that some actions in HumanAct12 involve interactions with objects, e.g., Drink, Lift dumbbell, Turn steering wheel. GPT-4V can imagine the existence of these virtual objects and generate corresponding keyframes. The test motions listed in Tab. 2 are seen for baseline methods and unseen for MLLMs during development.

Zero-shot motion synthesis has been explored by some CLIP-based methods [16, 36]. To evaluate the ability of our method to tackle this task utilizing the world knowledge of MLLMs, we compared the performance on Olympic sports following the setting in MotionCLIP [36] to [16, 36], excluding motions not suitable for physical tracking on the ground, for example, Cycling and Diving. In this experiment, corresponding motion data is unseen for both baseline methods and ours.

Metric Given those commonly-used inception models for evaluation, as in [6,37], are overfitting on their training datasets and fail to evaluate our method, we choose to utilize user preference as many prior works [2, 16, 36] where inception models are unavailable. We asked 50 volunteers to perform a user study in terms of two focuses: 1) the consistency with input texts, and 2) motion quality (physical feasibility, naturalness, etc.). We show the volunteers with randomly sampled motions generated by the same prompt (an action category in this experiment) from our method and baseline methods side by side. The volunteers are asked to select the one with the best performance according to the above two focuses.

Table 2: Motion Synthesis on HumanAct12. FreeMotion achieves good results without motion data.

	User Study		
	MDM [37]	MLD [6]	Ours
Warm up	26.00%	**38.00%**	36.00%
Walk	10.00%	22.00%	**68.00%**
Run	30.00%	32.00%	**38.00%**
Jump	16.00%	28.00%	**56.00%**
Drink	14.00%	**46.00%**	40.00%
Lift_dumbbell	26.00%	32.00%	**42.00%**
Sit	30.00%	**44.00%**	26.00%
Eat	22.00%	30.00%	**48.00%**
Turn_steering_wheel	32.00%	28.00%	**40.00%**
Phone	30.00%	32.00%	**38.00%**
Boxing	16.00%	24.00%	**60.00%**
Throw	20.00%	14.00%	**66.00%**
Average	22.67%	30.83%	**46.50%**

Analysis The results of motion synthesis on HumanAct12 are shown in Tab. 2. It is evident that FreeMotion outperforms traditional data-driven methods in most cases. The key factor contributing to this success is FreeMotion's ability to maintain physical plausibility, a challenge for methods like MDM and MLD. Fig. 4 highlights FreeMotion's capability to generate realistic, previously unseen motions on HumanAct12.

For the second experiment, the user preference score is shown in Tab. 3. Our method outperforms two existing zero-shot motion synthesis methods significantly. Fig. 5 also vividly illustrates that FreeMotion is capable of synthesizing realistic Olympic sports motions, whereas MotionCLIP and AvatarCLIP tend to produce unnatural motions. Despite MotionCLIP and AvatarCLIP benefiting from CLIP's zero-shot generalizability, they fall short in adhering to physical constraints and accurately interpreting the composition and sequence of motions. A case in point is a jump shot in basketball: ideally, the player first lifts the ball to the chest before jumping. However, these methods often struggle to replicate this sequential accuracy.

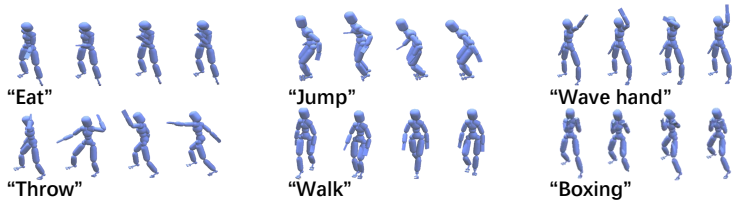

Fig. 4: Motion synthesis visualization results of FreeMotion on HumanAct12. FreeMotion can synthesize realistic motions across different categories.

4.2 Style Transfer

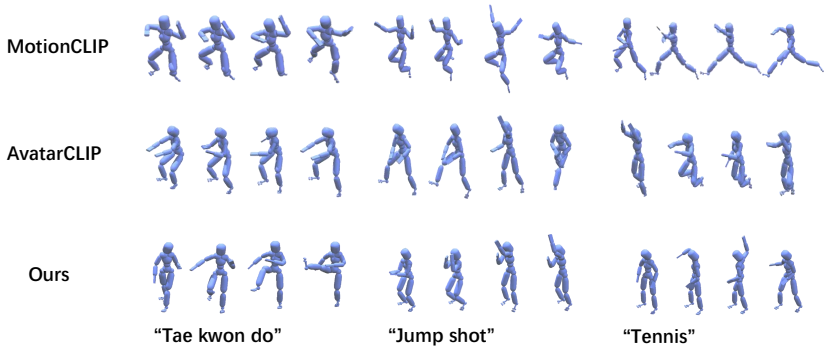

Fig. 5: Motion synthesis visualization results on Olympic sports. FreeMotion can synthesize satisfactory motions even on challenging Olympic sports.

Table 3: Olympic Sports. FreeMotion surpasses existing methods significantly.

Metrics	Methods										
User Study	MotionCLIP [36]	8.00%	12.00%	6.00%	2.00%	10.00%	6.00%	8.00%	4.00%	4.00%	6.00%	16.00%
	AvatarCLIP [16]	10.00%	6.00%	10.00%	8.00%	6.00%	10.00%	12.00%	8.00%	2.00%	12.00%	18.00%
	Ours	**82.00%**	**82.00%**	**84.00%**	**90.00%**	**84.00%**	**84.00%**	**80.00%**	**88.00%**	**94.00%**	**82.00%**	**66.00%**

Metrics	Methods										
User Study	MotionCLIP [36]	6.00%	4.00%	4.00%	12.00%	8.00%	2.00%	14.00%	2.00%	26.00%	20.00%	14.00%
	AvatarCLIP [16]	4.00%	2.00%	8.00%	20.00%	6.00%	12.00%	6.00%	4.00%	34.00%	32.00%	18.00%
	Ours	**90.00%**	**94.00%**	**88.00%**	**68.00%**	**86.00%**	**86.00%**	**80.00%**	**94.00%**	**40.00%**	**48.00%**	**68.00%**

We evaluate our method's ability to represent motion styles without any training data. For this evaluation, we closely adhere to the settings established in MotionCLIP [36], generating actions with specific styles directly from textual descriptions. This evaluation encompasses three action categories: Jump, Walk, and Stand, each expressed in eight distinct styles. We adopt the user study as before.

Table 4: Style Transfer. FreeMotion surpasses existing methods significantly.

	User Study		
	MotionCLIP [36]	AvatarCLIP [16]	Ours
Happy	22.67%	25.33%	**52.00%**
Proud	24.00%	18.00%	**58.00%**
Angry	14.00%	34.67%	**51.33%**
Childlike	28.67%	29.33%	**42.00%**
Depressed	14.67%	17.33%	**68.00%**
Drunk	11.33%	9.33%	**79.33%**
Old	17.33%	28.00%	**54.67%**
Heavy	20.00%	16.00%	**64.00%**
Average	19.08%	22.25%	**58.67%**

Analysis The average user preference score of each style is shown in Tab. 4. FreeMotion won more than half of the votes. One impressive capability of MLLM is imagining what one will do in a specific style. For example, one will walk with a stoop when he is old. MotionCLIP and AvatarCLIP can hardly make it since they don't have explicit world knowledge and reasoning ability using natural language. The visualization results are shown in Fig. 6. Though sometimes CLIP-based baseline method can generate visually realistic frames such as the "Jump+happy" of AvatarCLIP in Fig. 6, the whole motion clip is not physically plausible and has low quality.

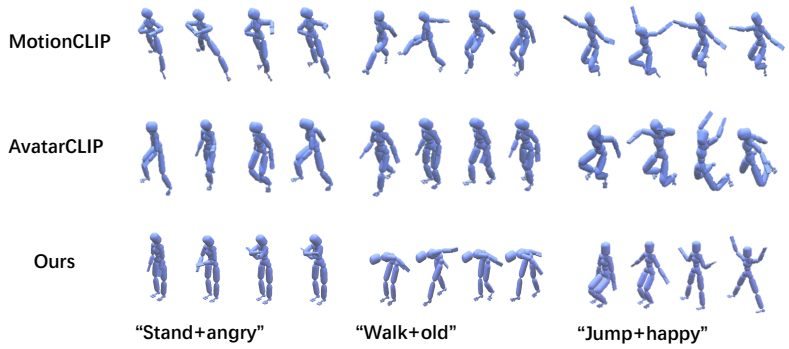

Fig. 6: Visualization results of style transfer. FreeMotion can add style to human motion using its world knowledge.

4.3 Human-Scene Interaction

Human-Scene Interaction presents a significant challenge in computer animation, necessitating not only the recognition of objects for interaction but also the generation of contextually appropriate motions. Specifically, for sitting and lying down, we utilize approximately 50 diverse items, including chairs, sofas, and beds, sourced from ShapeNet [5], and direct the humanoid to appropriately sit or lie on them. Similarly, for reaching tasks, we select around 50 different objects from ShapeNet and instruct the humanoid to reach them with a hand. The humanoid's initial position is set at a random distance from the object with a random orientation. We ran 40 times for each task.

Baseline Data-driven methods [14,41] solve Human-Scene Interaction synthesis by combining style reward from unlabeled motion data and task reward from human-designed reward function as in AMP [28]. UniHSI [41] formulates the task reward in human-scene interaction tasks as Chain of Contacts. It obtains knowledge from LLMs to reason the contact pairs. For the different settings and the unavailability of code, we report the results from [14, 41] for a rough comparison. We also implement an AMP-based baseline trained on SAMP [13] for a fairer comparison.

Metric In this task, we follow previous works [14, 41] that use *Success Rate* and *Contact Error* as the main metrics. However, these metrics should be computed with ground truth contact pairs, which are not available in our method. We manually set the contact pairs in the form of joints and target positions and inform GPT-4V in the prompt.

Table 5: Human-Scene Interaction. FreeMotion achieves good results on three interaction tasks.

Methods	Success Rate (%) ↑			Contact Error ↓		
	Sit	Lie Down	Reach	Sit	Lie Down	Reach
InterPhys - Sit [14]	93.7	-	-	0.09	-	-
InterPhys - Lie Down [14]	-	80.0	-	-	0.30	-
UniHSI [41]	94.3	**81.5**	**97.5**	**0.032**	**0.061**	0.016
AMP-Sit [28]	83.6	-	-	0.074	-	-
AMP-Lie Down [28]	-	28.3	-	-	0.334	-
AMP-Reach [28]	-	-	96.6	-	-	0.041
Ours	**95**	60	95	0.066	0.224	**0.012**

Analysis The results, as shown in Tab. 5, indicate that our method attains results comparable to previous methods, even in the absence of any motion data, which further implies that the MLLMs possess an innate understanding of scene interaction and object contact. However, the success rate decreases when lying down on a bed. This phenomenon can be mainly attributed to the rich contact in this process. Fig. 7 illustrates the visualization results, where FreeMotion effectively guides the humanoid in navigating towards and interacting with the target object.

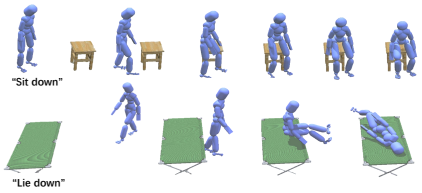

Fig. 7: Visualization of Human-scene interaction. FreeMotion can navigate to and interact with the target object.

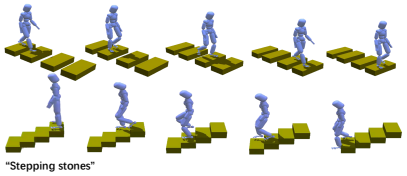

Fig. 8: Visualization results of stepping stones. FreeMotion can navigate over irregular terrain.

4.4 Stepping Stones

Navigating challenging, irregular terrain is crucial for locomotion, with each footstep subject to strict constraints in this task. In this experiment, we adopt ALLSTEPS [42] and select the best-performing policy (Adaptive) in the work for comparison.

Baseline ALLSTEPS [42] learns stepping-stone skills by utilizing deep reinforcement learning and curriculum learning. Though requiring no motion data, it needs carefully designed task-specific reward functions and training strategies to achieve good results.

Table 6: Stepping Stones. Please see the text for a detailed explanation of the numbers.

Task Parameter	ALLSTEPS [42]	Ours
Flat ($\Theta = 0$)		
$\Phi = 0$	**1.45, 1.50**	1.40, 1.45
$\Phi = 20$	1.35, 1.40	**1.40, 1.40**
Single-step ($\Phi = 0$)		
$\Theta = 50$	**0.80, 0.80**	0.60, 0.75
$\Theta = -50$	0.90, 0.95	**1.00, 1.10**
Continuous-step ($\Phi = 0$)		
$\Theta = 50$	—, 0.65	**0.50, 0.65**
$\Theta = -50$	0.65, 0.70	**0.75, 0.85**
Spiral ($\Phi = 20$)		
$\Theta = 30$	**0.80, 0.85**	0.40, 0.80
$\Theta = -30$	1.00, 1.10	**1.10, 1.30**

Metric We denote pitch Θ, yaw Φ, and distance d as the parameters of each step relative to the previous step. We repeat each scenario five times and record two numbers following [42]. The first number represents the maximum value of d for which the policy succeeds for all five runs. The second number represents the maximum value of d for which the policy succeeds in at least one of the runs. A larger number generally means a better capability to walk on difficult terrain.

Analysis The results are shown in Tab. 6. FreeMotion not only achieves comparable or superior results without motion data or complex reward design expertise but also excels in generating actions that are both naturally harmonious and physically feasible. As showcased in Fig. 8, it adeptly navigates irregular terrain, producing movements that are in line with the physical constraints of the stepping stones, further emphasizing its realistic motion synthesis capabilities.

5 Ablation Study

In this section, we conduct ablation experiments on the 12 action categories from HumanAct12 [12] to evaluate the effectiveness of our main designs. The dataset and the metric were introduced in the Motion Synthesis Task.

5.1 Keyframe Designer

The generation of keyframe descriptions is the core of our keyframe designer. In FreeMotion, we ask the MLLM to output a general full-body description with detailed body-part descriptions to decompose the keyframe spatially. In this experiment, we remove the detailed body-part descriptions, only outputting a general sentence of the full body, to evaluate the effectiveness of explicit spatial decomposition. The result is shown in Tab. 7. Detailed body-part descriptions help the keyframe generation process in motion synthesis.

5.2 Keyframe Animator

In this part, we remove the visual feedback mechanism in our keyframe animator, without which the MLLM can only call the command once for each body part. The result is shown in Tab. 8. Visual feedback allows the MLLM to further adjust the humanoid's pose and improve the motion quality.

Table 7: Ablation on body-part desc.

	User Study
w/o body-part desc.	26.00%
Ours	74.00%

Table 8: Ablation on visual feedback.

	User Study
w/o visual feedback	32.00%
Ours	68.00%

6 Conclusion

In this work, we for the first time, without any motion data, explore open-set human motion synthesis using natural language instructions as user control signals based on MLLMs across any motion task and environment. Our method can potentially serve as an alternative to motion capture for collecting human motion data, especially when the cost of motion capture is huge (e.g., collecting human interaction with different scenes).

Though we have evaluated the effectiveness of our method on many downstream tasks, its application can be expanded to more scenarios (e.g., human-human interactions, contact-rich human-object interaction).

There is much progress to be made in investigating technologies to improve the performance of our framework. Currently, our method can not handle complex human motions (e.g., dancing) or long text instructions. Its performance will also downgrade when the contact is rich. Future researchers may consider finetuning an MLLM with expert human motion knowledge. More powerful pose adjustment technologies, sometimes even a neural network, can be utilized for the mapping between natural language description and human pose. We hope our work can pave the way for future work in this area.

References

1. Gpt-4v(ision) system card (2023), https://api.semanticscholar.org/CorpusID:263218031
2. Aberman, K., Weng, Y., Lischinski, D., Cohen-Or, D., Chen, B.: Unpaired motion style transfer from video to animation. ACM Transactions on Graphics (TOG) **39**(4), 64–1 (2020)
3. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
5. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
6. Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., Yu, G.: Executing your commands via motion diffusion in latent space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18000–18010 (2023)
7. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. Journal of Machine Learning Research **24**(240), 1–113 (2023)
8. Dong, R., Han, C., Peng, Y., Qi, Z., Ge, Z., Yang, J., Zhao, L., Sun, J., Zhou, H., Wei, H., et al.: Dreamllm: Synergistic multimodal comprehension and creation. arXiv preprint arXiv:2309.11499 (2023)
9. Fussell, L., Bergamin, K., Holden, D.: Supertrack: Motion tracking for physically simulated characters using supervised learning. ACM Transactions on Graphics (TOG) **40**(6), 1–13 (2021)
10. Ghosh, A., Dabral, R., Golyanik, V., Theobalt, C., Slusallek, P.: Imos: Intent-driven full-body motion synthesis for human-object interactions. In: Computer Graphics Forum. vol. 42, pp. 1–12. Wiley Online Library (2023)
11. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5152–5161 (2022)
12. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: Conditioned generation of 3d human motions. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2021–2029 (2020)
13. Hassan, M., Ceylan, D., Villegas, R., Saito, J., Yang, J., Zhou, Y., Black, M.J.: Stochastic scene-aware motion prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11374–11384 (2021)
14. Hassan, M., Guo, Y., Wang, T., Black, M., Fidler, S., Peng, X.B.: Synthesizing physical character-scene interactions. arXiv preprint arXiv:2302.00883 (2023)
15. Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG) **35**(4), 1–11 (2016)
16. Hong, F., Zhang, M., Pan, L., Cai, Z., Yang, L., Liu, Z.: Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. arXiv preprint arXiv:2205.08535 (2022)
17. Hu, Y., Lin, F., Zhang, T., Yi, L., Gao, Y.: Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. arXiv preprint arXiv:2311.17842 (2023)

18. Huang, W., Wang, C., Zhang, R., Li, Y., Wu, J., Fei-Fei, L.: Voxposer: Composable 3d value maps for robotic manipulation with language models. arXiv preprint arXiv:2307.05973 (2023)
19. Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., Chen, T.: Motiongpt: Human motion as a foreign language. arXiv preprint arXiv:2306.14795 (2023)
20. Li, J., Wu, J., Liu, C.K.: Object motion guided human motion synthesis. ACM Transactions on Graphics (TOG) **42**(6), 1–11 (2023)
21. Li, P., Aberman, K., Zhang, Z., Hanocka, R., Sorkine-Hornung, O.: Ganimator: Neural motion synthesis from a single sequence. ACM Transactions on Graphics (TOG) **41**(4), 1–12 (2022)
22. Liu, Z., Lyu, K., Wu, S., Chen, H., Hao, Y., Ji, S.: Aggregated multi-gans for controlled 3d human motion prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 2225–2232 (2021)
23. Ma, Y.J., Liang, W., Som, V., Kumar, V., Zhang, A., Bastani, O., Jayaraman, D.: Liv: Language-image representations and rewards for robotic control (2023)
24. Ma, Y.J., Liang, W., Wang, G., Huang, D.A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., Anandkumar, A.: Eureka: Human-level reward design via coding large language models. arXiv preprint arXiv:2310.12931 (2023)
25. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: Amass: Archive of motion capture as surface shapes. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5442–5451 (2019)
26. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems **35**, 27730–27744 (2022)
27. Peng, X.B., Guo, Y., Halper, L., Levine, S., Fidler, S.: Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. ACM Transactions On Graphics (TOG) **41**(4), 1–17 (2022)
28. Peng, X.B., Ma, Z., Abbeel, P., Levine, S., Kanazawa, A.: Amp: Adversarial motion priors for stylized physics-based character control. ACM Transactions on Graphics (ToG) **40**(4), 1–20 (2021)
29. Petrovich, M., Black, M.J., Varol, G.: Temos: Generating diverse human motions from textual descriptions. In: European Conference on Computer Vision. pp. 480–497. Springer (2022)
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
31. Rempe, D., Luo, Z., Bin Peng, X., Yuan, Y., Kitani, K., Kreis, K., Fidler, S., Litany, O.: Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13756–13766 (2023)
32. Ren, J., Zhang, M., Yu, C., Ma, X., Pan, L., Liu, Z.: Insactor: Instruction-driven physics-based characters (2023)
33. Rocamonde, J., Montesinos, V., Nava, E., Perez, E., Lindner, D.: Vision-language models are zero-shot reward models for reinforcement learning. arXiv preprint arXiv:2310.12921 (2023)
34. Smith, R., et al.: Open dynamics engine (2005)
35. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)

36. Tevet, G., Gordon, B., Hertz, A., Bermano, A.H., Cohen-Or, D.: Motionclip: Exposing human motion generation to clip space. In: European Conference on Computer Vision. pp. 358–374. Springer (2022)
37. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. arXiv preprint arXiv:2209.14916 (2022)
38. Wang, Z., Chen, Y., Liu, T., Zhu, Y., Liang, W., Huang, S.: Humanise: Language-conditioned human motion generation in 3d scenes (2022)
39. Wei, D., Sun, X., Sun, H., Li, B., Hu, S., Li, W., Lu, J.: Enhanced fine-grained motion diffusion for text-driven human motion synthesis (2023)
40. Won, J., Gopinath, D., Hodgins, J.: Physics-based character controllers using conditional vaes. ACM Transactions on Graphics (TOG) **41**(4), 1–12 (2022)
41. Xiao, Z., Wang, T., Wang, J., Cao, J., Zhang, W., Dai, B., Lin, D., Pang, J.: Unified human-scene interaction via prompted chain-of-contacts. arXiv preprint arXiv:2309.07918 (2023)
42. Xie, Z., Ling, H.Y., Kim, N.H., van de Panne, M.: Allsteps: curriculum-driven learning of stepping stone skills. In: Computer Graphics Forum. vol. 39, pp. 213–224. Wiley Online Library (2020)
43. Xu, S., Li, Z., Wang, Y.X., Gui, L.Y.: Interdiff: Generating 3d human-object interactions with physics-informed diffusion (2023)
44. Yang, Z., Li, L., Lin, K., Wang, J., Lin, C.C., Liu, Z., Wang, L.: The dawn of lmms: Preliminary explorations with gpt-4v (ision). arXiv preprint arXiv:2309.17421 **9**(1), 1 (2023)
45. Yao, H., Song, Z., Chen, B., Liu, L.: Controlvae: Model-based learning of generative controllers for physics-based characters. ACM Transactions on Graphics (TOG) **41**(6), 1–16 (2022)
46. Yu, W., Gileadi, N., Fu, C., Kirmani, S., Lee, K.H., Arenas, M.G., Chiang, H.T.L., Erez, T., Hasenclever, L., Humplik, J., et al.: Language to rewards for robotic skill synthesis. arXiv preprint arXiv:2306.08647 (2023)
47. Yuan, Y., Song, J., Iqbal, U., Vahdat, A., Kautz, J.: Physdiff: Physics-guided human motion diffusion model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16010–16021 (2023)
48. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. arXiv preprint arXiv:2208.15001 (2022)
49. Zhang, Y., Huang, D., Liu, B., Tang, S., Lu, Y., Chen, L., Bai, L., Chu, Q., Yu, N., Ouyang, W.: Motiongpt: Finetuned llms are general-purpose motion generators. arXiv preprint arXiv:2306.10900 (2023)
50. Zhao, K., Wang, S., Zhang, Y., Beeler, T., Tang, S.: Compositional human-scene interaction synthesis with semantic control (2022)
51. Zhao, K., Zhang, Y., Wang, S., Beeler, T., Tang, S.: Synthesizing diverse human motions in 3d indoor scenes (2023)