

RL from Physical Feedback: Aligning Large Motion Models with Humanoid Control

Junpeng Yue^{1,2}, Zepeng Wang^{2,3}, Yuxuan Wang^{1,2}, Yu Zhang², Xinrun Xu²,
Bin Cao^{2,4}, Ziluo Ding², Sipeng Zheng², Jiangxing Wang², and Zongqing
Lu^{1,2*}

¹ School of Computer Science, Peking University, Beijing, China
yuejunpeng@stu.pku.edu.cn, zongqing.lu@pku.edu.cn

² BeingBeyond, Beijing, China

³ School of Cyber Science and Engineering, Wuhan University, Wuhan, China

⁴ Institute of Automation, Chinese Academy of Sciences, Beijing, China

Abstract. This paper focuses on a critical challenge in robotics: translating text-driven generated human motions into executable actions for real humanoid robots. While existing text-to-motion (T2M) generation methods achieve semantic alignment between language and motion, they often produce physically infeasible motions unsuitable for real-world deployment. To bridge the gap between T2M and humanoid execution, we propose **Reinforcement Learning from Physical Feedback (RLPF)**, a novel framework that integrates text-conditioned motion generation with motion-conditioned humanoid whole-body control. RLPF employs a low-level motion tracking policy to assess feasibility in physical simulators, providing rewards for fine-tuning the high-level motion generator. Moreover, RLPF introduces an alignment verification module to preserve semantic alignment with text instructions. This joint optimization ensures both physical feasibility and text alignment of T2M generators. Furthermore, to overcome the limitation of frozen tracking policy, we propose **Hierarchical RLPF**, a co-training framework that iteratively optimizes the high-level T2M generator and the low-level tracking policy. Extensive experiments show that RLPF and Hierarchical RLPF greatly outperform baseline methods in generating physically feasible motions while maintaining semantic alignment with text instructions.

Keywords: Reinforcement Learning · Humanoid Robotics

1 Introduction

Humanoid robots have advanced rapidly in recent years, driven by improvements in actuators, torque control, and motion tracking policies. They are now capable of performing expressive whole-body behaviors with greater robustness and fluidity, when provided with high-quality reference trajectories [6, 13, 39]. While such advances endow robots with powerful physical capabilities, this is only the

* Corresponding author.

physical side of the entire humanoid system. To achieve the intelligent side, a humanoid robot also requires another critical component—an intelligent “brain” that can translate open-form instructions into executable motions, serving as the bridge between high-level intention and a low-level controller.

One promising approach towards this goal is text-to-motion (T2M) generation [31, 32, 42]. By learning from human motion datasets with text labels, T2M models translate natural language into temporal sequences of human joint positions and rotations, which can be retargeted to humanoid embodiments and then serve as reference trajectories. Previous works mainly relied on small-scale mocap datasets, such as AMASS [16] and HumanML3d [3]. More recently, with the emergence of large-scale motion datasets derived from Internet videos, such as MotionX [14] and MotionMillion [1], large language models (LLMs) have been incorporated into motion generation, transforming T2M models into **Large Motion Models** [37, 38, 46]. These advances achieve greater motion diversity, improved generalization, and closer alignment with human intent.

However, when we apply T2M models as humanoid brains and integrate them with humanoid whole-body control policies, a crucial gap remains. Existing T2M research has primarily focused on semantic alignment with text descriptions and visual motion fidelity, while largely neglecting physical feasibility. As a result, generated motions often suffer from unrealistic artifacts such as skating, floating, and abrupt transitions, which severely limit their usability for humanoid control. Moreover, the reliance on Internet video-derived motion data further exacerbates this issue due to estimation errors introduced during video-to-motion conversion. Together, these limitations make current T2M models less ideal to serve as reliable humanoid brains capable of producing executable motions in real-world settings. Therefore, this paper attempts to address this core research question:

How can we align text-to-motion models with humanoid motion tracking policies, ensuring generated motions can be deployed on real-world humanoid robots, while maintaining text alignment?

To address this challenge, we consider incorporating physical constraints and feedback into the large motion model training process, which could be essential for improving the physical feasibility of generated motion. Previous attempts have considered physical metrics of human motions such as skating, floating, and penetration [5, 35, 42]. However, these metrics only provide indirect measures of physical feasibility and therefore cannot directly assess the quality of generated motion when deployed on humanoid robots. Instead, we propose to obtain physical feedback from motion tracking policies, in which the generated motions serve as reference for the tracking policy, and the execution results in simulators offer a more direct and comprehensive feedback of physical feasibility.

Building upon the assessment of physical feasibility, we further leverage this physical feedback to refine the motion generation model. Inspired by recent advances in reinforcement learning from human feedback (RLHF) [4, 20], we regard simulator-derived tracking feedback as a reward signal and fine-tune pre-trained large motion models using reinforcement learning. However, our experiments show that optimizing solely for physical feasibility tends to produce overly con-

servative motions and leads to a catastrophic loss of semantic alignment with text instructions. For instance, as illustrated in the appendix, regardless of the input text, the model degenerates to generating only standing motions. To address this problem, we further integrate an Alignment Verification Module into RLPF framework that explicitly preserves semantic correspondence with text.

More specifically, RLPF consists of three key components: (i) *Motion Tracking Policy*, which provides physical feasibility feedback in physical simulators; (ii) *Alignment Verification Module*, which evaluates the semantic alignment between generated motions and input text instructions; and (iii) *reinforcement learning fine-tuning framework*, which optimizes the large motion model using both physical feasibility signals from the motion tracking policy and semantic alignment feedback from the verification module.

Furthermore, the performance of RLPF is ultimately bounded by the capacity of the frozen tracking policy. To mitigate this limitation, we further introduce **Hierarchical RLPF**, which integrates upstream RLPF fine-tuning of the T2M model with downstream tracking policy learning, thereby iteratively optimizing the overall text-motion-action (T2M2A) pipeline and improving robustness and generalization of *both the T2M model and the motion tracking policy*.

In summary, our key contributions are threefold:

- We propose **Reinforcement Learning from Physical Feedback** (RLPF), a novel framework that optimizes large motion models using physical feedback from a motion tracking policy together with an alignment verification module that assesses text-motion correspondence, thereby achieving stable closed-loop improvements in the physical feasibility of generated motions while preserving semantic alignment for real-world deployment.
- We further introduce **Hierarchical RLPF**, which integrates RL fine-tuning of the high-level large motion model with low-level tracking policy learning in an alternating, closed-loop manner, enabling iterative feasibility improvements without sacrificing semantic alignment in the full T2M2A pipeline.
- Through extensive experiments on AMASS and MotionX, we demonstrate that our framework substantially improves physical feasibility, while maintaining strong semantic alignment, thereby strengthening the text-motion-action pipeline and bridging large motion models with real humanoid control.

2 Related Works

Human Motion Generation. Recent progress in text-to-motion generation is driven by high-quality datasets such as AMASS [16] and large-scale video-derived datasets (e.g., MotionX [14], MotionMillion [1]), together with advances in generative modeling. Representative approaches include discrete token-based transformer models (e.g., T2M-GPT [44], MotionGPT [9]) and diffusion-based generators (e.g., MDM [32], MotionDiffuse [45]). However, these models are typically trained in a kinematic space and may introduce unrealistic artifacts such as foot skating, floating, penetration, and abrupt transitions.

Reinforcement Learning Fine-Tuning for Large Models. RL has emerged as a powerful tool for fine-tuning large pre-trained models, optimizing their performance for specific tasks or aligning them with user preferences. In natural language processing, reinforcement learning from human feedback (RLHF) leverages human evaluation data, such as text summarization of tasks [29], to train reward models and refine outputs using RL algorithms like Proximal Policy Optimization (PPO) [24]. This approach overcomes limitations of traditional supervised fine-tuning for complex objectives. In human motion generation, RL is increasingly adopted to enhance motion quality, generalization, and real-world applicability. For instance, InstructMotion [19] frames text-to-motion generation as a Markov decision process, using contrastive pre-trained encoders (text and motion) to design rewards that improve generalization to novel descriptions and reduce overfitting caused by limited training data. Similarly, RL has been applied to motion control, such as training physics-based tracking policies via PPO for simulated environments [15]. Inspired by these advances, we employ RL to fine-tune motion generation models, prioritizing physical feasibility through physical feedback. By leveraging motion tracking policies to generate rewards, we optimize motions to respect robotic kinematic and dynamic constraints, bridging the gap between generative motion models and practical robotic execution.

Physics-based Motion Imitation and Humanoid Whole-body Control. Physics-based motion imitation has been extensively studied in graphics and robotics: DeepMimic [21] trains tracking policies in simulation to imitate reference motions, and subsequent works extend it to multi-skill settings and partial observations [22, 30]. Meanwhile, recent systems track retargeted human motions for real humanoid robots [2, 6, 7, 11, 13, 43], but they typically rely on high-quality human motion inputs and strong low-level policies. Different from the DeepMimic line that primarily optimizes a tracking policy to track fixed reference motions, we use a tracking policy as an evaluator and *optimize the text-to-motion generator* so that its generated motions become inherently more physically feasible and can be deployed on real humanoid robots.

Beyond pure humanoid control, recent work explores *language-conditioned humanoid control* that learns a mapping from language instructions to whole-body actions (e.g., LangWBC [26], UH-1 [18], LeVERB [40] and RoboGhost [12]). In practice, these methods first learn a motion tracking policy in high quality motions, and then extend it with language instructions, which makes them dependent on curated reference motion datasets and limits the textual instruction alignment. In contrast, our method leverages large-scale text-to-motion generators as base models and RL fine-tunes them with execution-based physical feedback, improving physical feasibility while retaining semantic alignment.

3 RL from Physical Feedback (RLPF)

This section outlines the key components of our RLPF framework, as illustrated in Figure 1. First, we pre-train a large motion model as the base model for RLPF (Section 3.1). Second, we introduce two essential reward components: (i)

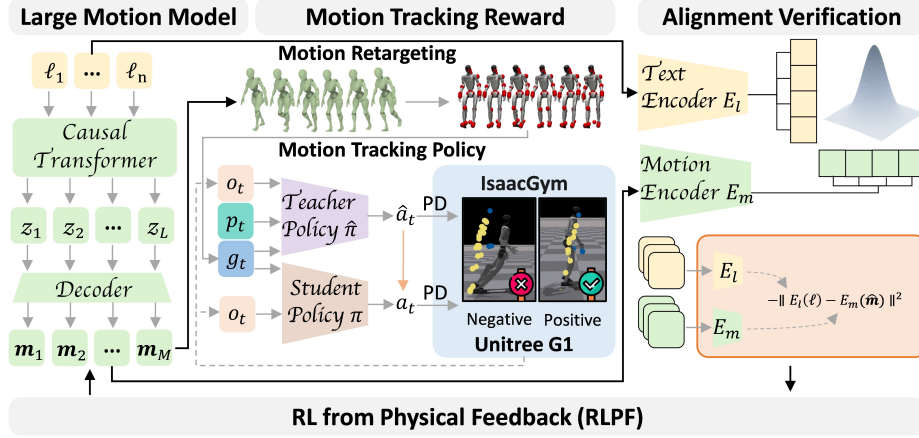


Fig. 1: Overview of RLPF, which consists of three key components: *(i) Large Motion Model*, pre-trained on large-scale motion datasets and utilized as the base model for RL fine-tuning; *(ii) Motion Tracking Reward*, where a humanoid motion tracking policy provides physical feasibility feedback by evaluating generated motions in physics simulators; and *(iii) Alignment Verification Module*, which evaluates the semantic alignment between generated motions and input textual instructions.

a motion tracking reward that evaluates the physical feasibility of generated motions (Section 3.2), and *(ii) an alignment verification module* that preserves text-motion semantic alignment for RLPF framework (Section 3.3). Next, we present the core component of our framework: reinforcement learning (RL) fine-tuning of the pre-trained large motion model (Section 3.4). In addition, we further introduce the Hierarchical RLPF variant, which iteratively refines both the large motion model and the motion tracking policy to iteratively improve physical feasibility while maintaining semantic alignment (Section 3.5).

3.1 Pre-Training of Large Motion Model

Motion Tokenizer. Existing motion models [9,37] in this field typically employ vector quantization (VQ) [36] or its variants [10] to represent human motions, treating them as a “foreign language”. These approaches use a motion tokenizer to discretize motion sequences into tokens. The tokenizer comprises three components: *(i)* an encoder $\mathcal{E}$ that maps motion sequences to latent codes with temporal downsampling with a stride r , *(ii)* a decoder $\mathcal{D}$ that reconstructs motions from tokens, and *(iii)* a codebook $\mathcal{C}$ where each codeword represents a distinct human body movement pattern. Given a motion sequence $\mathbf{m}_{1:M} = \{\mathbf{m}_i\}_{i=1}^M$ with M frames, the tokenizer first encodes $\mathbf{m}_{1:M}$ into motion tokens $z_{1:L} = \{z_i\}_{i=1}^L$ (for brevity, we write $\mathbf{m} = \mathbf{m}_{1:M}$ and $z = z_{1:L}$), then decodes them back to the reconstructed motion $\tilde{\mathbf{m}}_{1:M} = \mathcal{D}(z_{1:L}) = \mathcal{D}(\mathcal{E}(\mathbf{m}_{1:M}))$, where $L = \lceil \frac{M}{r} \rceil$.

Large Motion Model for T2M Generation. Recent works have shown increasing interest in leveraging large language models (LLMs) for text-to-motion

(T2M) generation, enabling temporally coherent motion sequence synthesis, capitalizing on their ability to interpret human intent [9]. While our approach follows this paradigm, we emphasize that our framework is not restricted to any specific T2M methodology. In this work, our model extends the LLM’s vocabulary with K discrete codes from a motion codebook. Our LLM backbone P_Θ , implemented as a decoder-only causal transformer, first encodes the input text tokens $l_{1:n} = \{l_i\}_{i=1}^n$ (for brevity, we denote $l = l_{1:n}$), then auto-regressively generates corresponding motion tokens $z_{1:L} = \{z_i\}_{i=1}^L$. During pre-training, we optimize the model using negative log-likelihood minimization over predicted tokens:

$$\mathcal{L}(\Theta) = - \sum_{j=1}^L \log P_\Theta(z_j | l, z_{1:j-1}) \quad (1)$$

where l represents the text instruction, $z_{1:j-1}$ and z_j denote the first $j-1$ motion tokens and the j -th motion token to predict, respectively. Θ represents the model parameters of P_Θ and L is the length of the target sequence.

3.2 Motion Tracking Reward

Given a language instruction l , the pre-trained large motion model generates a predicted motion sequence $\hat{\mathbf{m}}_{1:M}$ in a kinematic pose representation space (denoted as $\hat{\mathbf{m}}$ for brevity). We then evaluate its physical feasibility using a motion tracking reward, computed via a three-stage pipeline: *(i)* motion retargeting, *(ii)* motion tracking policy training and *(iii)* offline evaluation.

Motion Retargeting. We retarget human motions to the humanoid robot in two stages. First, we estimate a robot-specific SMPL **shape** by placing both SMPL and the robot in a canonical T-pose and optimizing the SMPL shape parameters β via gradient descent, so that the resulting SMPL body proportions (e.g., limb lengths and overall stature) better match the robot morphology. Second, with the fitted shape β^* fixed, we retarget each motion frame-by-frame by solving for the **humanoid motion**: root pose (translation and orientation) and joint angles. Concretely, we minimize a paired-joint position loss that aligns the 3D positions of 14 semantically corresponding joints between the robot and the shaped SMPL motion, optionally with additional regularization terms to ensure physically feasible configurations. We optimize this objective using gradient-based updates over the motion trajectory. The resulting optimized root pose and joint angles constitute the final robot motion, which is subsequently used as the input reference motion for the motion tracking policy.

Motion Tracking Policy. After retargeting, the motion becomes morphologically consistent with the humanoid robot. Nevertheless, verifying its physical feasibility for real-world deployment still requires a well-trained humanoid motion tracking policy to assess the tracking performance of the generated motion in physics simulators. To train such a tracking policy, following BumbleBee [39], we adopt a two-stage teacher–student framework. In the first stage, we learn an oracle teacher using PPO [24] that leverages privileged, simulator-only information. In the second stage, we distill the teacher into a real-world deployable

student policy by removing privileged observations that are inaccessible in the real world. All training is performed in IsaacGym [17], enabling high-throughput parallel simulation. The details of our policy training procedure are as follows:

Stage 1: Teacher Policy Training. The teacher policy $\hat{\pi}$ is trained as an oracle in simulation using PPO [24]. It leverages privileged information p_t , which is unavailable in the real world, together with observations o_t and motion-tracking targets g_t derived from the retargeted reference motion. The policy outputs actions in the form of target joint positions $\hat{a}_t \in \mathbb{R}^{23}$, which are executed by standard Proportional-Derivative (PD) controllers at each timestep. Observation details and reward specifications are provided in the appendix.

Stage 2: Student Policy Training. The student policy π distills the knowledge of teacher policy while receiving only real-world-observable inputs: observations o_t and motion-tracking targets g_t . Following a DAgger-like [23] procedure, we first roll out the student policy in the simulator to collect observations o_t , motion-tracking targets g_t , along with privileged information p_t for logging. We then query the teacher policy $\hat{\pi}$ to compute oracle actions $\hat{a}_t$ at the corresponding input, which serve as supervision labels. The student is refined by minimizing an MSE loss $l = \|a_t - \hat{a}_t\|^2$ over the aggregated dataset. This iterative process continues until convergence, yielding a deployable policy that operates without privileged information at test time.

Offline Evaluation. After obtaining a deployable student policy, we perform offline evaluation in a physics simulator (e.g., IsaacGym). We quantify the physical feasibility of a generated motion $\hat{\mathbf{m}}$ using a tracking success signal, which is then used to construct the motion-tracking reward for reinforcement learning of the large motion model. This process is formalized as:

$$R_{\text{tracking}}^{\hat{\mathbf{m}}} = \mathbb{I}(\text{Succ}(\pi, \hat{\mathbf{m}})). \quad (2)$$

where π denotes the student policy introduced in Section 3.2, $\hat{\mathbf{m}}$ denotes the generated motion, and $\text{Succ}(\pi, \hat{\mathbf{m}})$ is a binary success indicator that specifies whether $\hat{\mathbf{m}}$ is executed successfully by π in the simulator. A rollout is marked as a failure if the positional deviation between the reference motion and the executed state of the robot exceeds a threshold ϵ , or if the robot loses balance.

3.3 Motion Alignment Verification

We perform RL optimization using the motion tracking reward described in Section 3.2, aiming to generate motion sequences that are more physically feasible. However, as shown later in our experiments (Section 4.5), this single-objective optimization may compromise the semantic alignment with textual instructions. To optimize both semantic alignment and physical feasibility, we propose an Alignment Verification Module to regularize the large motion model using contrastive pre-trained encoders (E_m for motion and E_l for text). Both encoders are trained using the following objective:

$$\mathcal{L}_{CL} = (1 - y)(\|E_l(l) - E_m(\mathbf{m})\|)^2 + y(\max(0, \alpha - \|E_l(l) - E_m(\mathbf{m})\|))^2 \quad (3)$$

where y is a binary label indicating whether a text-motion pair is matched, α is a margin threshold, and $E_l(l)$ and $E_m(\mathbf{m})$ denote the text and motion embeddings, respectively. The encoders project both modalities into a shared latent space, in which higher semantic consistency corresponds to smaller Euclidean distances under this metric. We define the alignment verification module as

$$R_{align}^{\hat{\mathbf{m}}} = -(\|E_l(l) - E_m(\hat{\mathbf{m}})\|)^2 \quad (4)$$

where $R_{align}^{\hat{\mathbf{m}}}$ denotes the alignment verification reward.

3.4 Fine-Tuning of Motion Generator via RL

After obtaining the pretrained large motion model and the two reward components, we employ reinforcement learning (RL) to fine-tune the model, thereby improving the physical feasibility while maintaining the textual alignment of the generated motions. We treat the motion generator P_Θ as the policy model and optimize it via Group Relative Policy Optimization (GRPO) [27]. This RL algorithm improves upon PPO by removing the need for a critic model and estimating baselines directly from relative group scores. For each text instruction l sampled from dataset distribution $P(L)$, GRPO first samples a group of motion token sequences $Z = \{z^1, z^2, \dots, z^G\}$ from the previous policy $P_{\Theta_{old}}$. Each sampled token sequence z^i is decoded into a generated motion $\hat{\mathbf{m}}^i$. We compute its scalar reward as the weighted sum of the motion-tracking reward and the alignment-verification reward:

$$\mathcal{R}^i = \lambda_{\text{track}} R_{\text{tracking}}^{\hat{\mathbf{m}}^i} + \lambda_{\text{align}} R_{\text{align}}^{\hat{\mathbf{m}}^i}, \quad (5)$$

where λ_{track} and λ_{align} denote reward weights, with their values provided in the appendix. The current policy P_Θ is then optimized by maximizing the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\mathcal{D}} \left[\frac{1}{G} \sum_{i=1}^G \min \left(r_\theta^i A^i, \text{clip}(r_\theta^i, 1 - \epsilon, 1 + \epsilon) A^i \right) - \beta \mathbb{D}_{\text{KL}}(P_\Theta \| P_{\Theta_{\text{ref}}}) \right] \quad (6)$$

Here $\mathcal{D}$ denotes the joint sampling distribution defined by

$$l \sim P(L), \quad z^i \sim P_{\Theta_{\text{old}}}(Z | l) \text{ for } i = 1, \dots, G.$$

The importance ratio is given by $r_\theta^i = \frac{P_\Theta(z^i | l)}{P_{\Theta_{\text{old}}}(z^i | l)}$.

The KL divergence term is computed as:

$$\mathbb{D}_{\text{KL}}(P_\Theta \| P_{\Theta_{\text{ref}}}) = \frac{P_{\Theta_{\text{ref}}}(z^i | l)}{P_\Theta(z^i | l)} - \log \frac{P_{\Theta_{\text{ref}}}(z^i | l)}{P_\Theta(z^i | l)} - 1. \quad (7)$$

The optimization involves two key hyperparameters: ϵ , which constrains the policy update magnitude, and β , which balances the KL-divergence regularization term for improved stability. Here, $P_{\Theta_{\text{ref}}}$ denotes the reference policy, typically the pre-trained base large motion model, while A^i represents the normalized advantage computed within each group of size G :

$$A^i = \frac{\mathcal{R}^i - \text{mean}(\{\mathcal{R}^j\}_{j=1}^G)}{\text{std}(\{\mathcal{R}^j\}_{j=1}^G)}. \quad (8)$$

Intuitively, the GRPO objective (6) encourages each generated motion to achieve higher group-relative rewards while maintaining a bounded deviation from the previous policy and the reference model, thereby ensuring both optimization stability and physical feasibility throughout the fine-tuning process.

3.5 Hierarchical RLPF

The performance of the plain RLPF described above is ultimately bounded by the capacity of the tracking policy: tracking failures may arise not only from physically infeasible motions, but also from limitations of the policy itself. To mitigate this limitation, we couple upstream RLPF fine-tuning of the large motion model with downstream tracking policy learning. In this framework, RLPF encourages the large motion model to generate motions that are more physically feasible in the simulator while preserving alignment with the input text. Furthermore, during the RLPF fine-tuning process, the large motion model naturally samples a diverse set of on-policy motions; we reuse these motions as training data for the tracking policy, providing an on-policy data source that is both more physically feasible and more diverse than the original offline dataset.

Concretely, we alternate between two steps: (i) updating the large motion via RLPF, where the current tracking policy serves as the evaluator providing rewards, and (ii) updating the tracking policy using motion trajectories generated by the current large motion model. We refer to completing these two steps once as one *iteration*. In this scheme, the large motion model acts as a generator of training data for the tracking policy, while the tracking policy supplies physically grounded feedback to guide large motion model fine-tuning. We refer to this iterative, hierarchical co-training procedure as **Hierarchical RLPF**. This design progressively improves downstream tracking capability while largely preserving text-motion alignment within the overall text-motion-action pipeline, thereby yielding consistent gains in its overall performance.

4 Experiments

In this section, we present a rigorous empirical evaluation of our proposed method, structured to systematically address both high-level generation evaluation and low-level motion tracking evaluation.

4.1 Experimental Setup

Compared Baselines. To the best of our knowledge, RLPF is the first attempt that employs reinforcement learning to optimize the large motion model for enhanced feasibility. We evaluate RLPF against five baseline approaches: (i) **Base Model** which refers to the pre-trained large motion model used as our

base model for fine-tuning. **(ii) SFT-Filter** which refers to supervised fine-tuning on data filtered by a motion tracking policy, where only samples that can be successfully tracked are retained. **(iii) RobotMDM** [25] which learns a critic model for physical feasibility and integrates it with SFT loss. To ensure a fair comparison, we transfer the RobotMDM method onto the same LLM backbone. **(iv) PhysDiff** [42] which injects physical constraints by projecting simulated motion outcomes into the diffusion denoising steps for human avatars. To ensure a fair comparison, we further apply the same retargeting procedure. **(v) LangWBC** [26] which trains a text-based whole-body tracking policy using a CVAE [28] with DAGger [23] to imitate a teacher tracking policy.

Datasets and Benchmarks. To comprehensively evaluate our framework, we apply plain RLPF to the MoCap dataset AMASS, and evaluate Hierarchical RLPF on two subsets of the relatively lower-quality non-MoCap dataset MotionX: the walking subset (MotionX-W) and the jumping subset (MotionX-J). Following BumbleBee [39] and Unitracker [41], we utilize PHC [15] to filter the datasets. Additional details are provided in the appendix.

Simulator Setting. Following previous works on motion tracking policies [6, 8], we perform a three-stage evaluation protocol. First, we conduct large-scale reinforcement learning training in IsaacGym. Second, we perform a zero-shot transfer to the MuJoCo [33] simulator to assess cross-simulator generalization of the policy. Finally, we carry out physical deployment on a Unitree G1 humanoid robot to verify the real-world effectiveness of our framework.

Large Motion Model setting. We use VQ-VAE [36] as our motion tokenizer architecture, and adopt LLaMA2-7B [34] as the LLM backbone for motion generation. Further architectural and training details are provided in the appendix.

Evaluation Metrics. We consider both high-level motion generation metrics and low-level tracking metrics to evaluate our large motion model.

- **High-Level Motion Generation Metrics.** These metrics evaluate text-motion semantic alignment as well as the fidelity of generated motions compared to ground-truth sequences. Following [3], we report standard text-motion retrieval metrics (R@1, R@3), Multimodal Distance (MM-D), Fréchet Inception Distance (FID), and diversity (DIV), which together reflect alignment, realism, and sample variety. Details are described in the appendix.
- **Low-Level Motion Tracking Metrics.** Before real-robot deployment, we assess the trackability and physical executability of generated motions in both IsaacGym and MuJoCo, aligning with OmniH2O [7] and PHC [15]. Key metrics include: **(i) Success Rate (SR)**, indicating whether a rollout completes without falls or severe tracking failure; **(ii) MPJPE** (rad), measuring joint-angle errors; and **(iii) MPKPE** (m), measuring keypoint position errors. Full metric definitions and protocols are provided in the appendix.

4.2 Stable Feasibility Improvement of RLPF

The main results are summarized in Table 1. We ask whether RLPF can make a large motion model generate motions that are **more physically feasible** than

Table 1: Quantitative results for high-level and low-level evaluations on the AMASS.

Method	High-level					Low-level	
	FID↓	R@1↑	R@3↑	MM-D↓	DIV→	SR _{Isaac} ↑	SR _{MuJoCo} ↑
GT	-	0.52	0.80	2.96	10.37	-	-
Base Model	0.29	0.47	0.78	3.06	10.59	0.83	0.85
SFT-Filter	0.25	0.48	0.78	3.02	10.62	0.85	0.87
RobotMDM [25]	0.27	0.48	0.78	3.10	10.61	0.86	0.88
PhysDiff [42]	0.40	0.35	0.64	3.32	9.57	0.84	0.82
LangWBC [26]	-	-	-	-	-	0.81	0.79
RLPF	0.26	0.50	0.79	3.02	10.58	0.95	0.94

baselines **without degrading high-level text-motion alignment**. RLPF yields **substantial gains in low-level feasibility** and, in most cases, **also improves high-level alignment** with input textual instructions.

Supervised fine-tuning baselines (i.e., Base Model, SFT-Filter, and RobotMDM) achieve comparable high-level alignment, but lag far behind in low-level feasibility. PhysDiff improves feasibility on human avatars via projection [42]; however, it is not explicitly aligned with the humanoid tracking policy, leading to suboptimal feasibility after retargeting, and its smaller diffusion backbone also weakens high-level alignment. LangWBC directly predicts actions, bypassing motion generation and retargeting, so high-level generation metrics are not applicable; its lightweight MLP architecture further limits capacity, resulting in weaker feasibility on unseen textual instructions in the test set.

4.3 Stable Feasibility Improvement of Hierarchical RLPF

We evaluate Hierarchical RLPF on two MotionX subsets, **MotionX-W** (walk) and **MotionX-J** (jump) in Table 2. Across iterations, Hierarchical RLPF yields **consistent improvements in feasibility**, boosting the SR by **about 40 percentage points** on both subsets. Meanwhile, **text-motion alignment and generation diversity remain largely stable** (FID, R@1, and MM-D), indicating an iterative refinement of physical feasibility while maintaining high-level generation quality. Complete results are provided in the appendix.

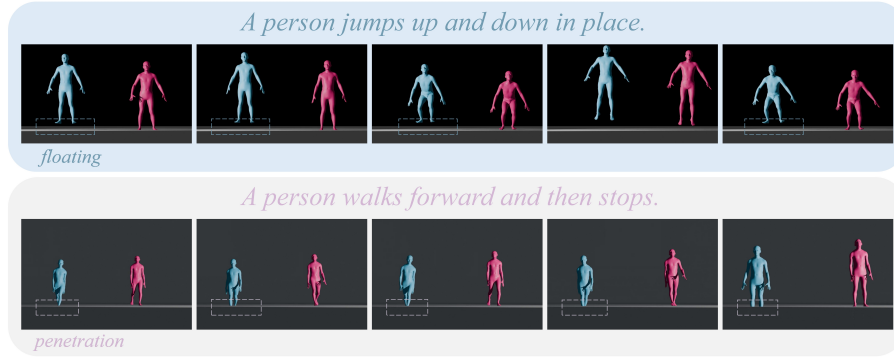
4.4 Qualitative Evaluation

We present a qualitative evaluation of the RLPF framework across three stages: motion generation, cross-simulator transfer, and real-world deployment.

In Figure 2, we compare motions generated by the Base Model and RLPF under the same textual instructions. The results show that, without RL fine-tuning, the Base Model often exhibits physical artifacts such as foot floating and ground penetration, whereas RLPF produces noticeably more physically feasible and semantically aligned motions in a consistent manner.

Table 2: Quantitative results of Hierarchical RLPF on MotionX. **FID**, **R@1**, and **MM-D** are **high-level** generation metrics, while **SR** measures **low-level** physical feasibility.

Iter	MotionX-W				MotionX-J			
	FID ↓	R@1 ↑	MM-D ↓	SR ↑	FID ↓	R@1 ↑	MM-D ↓	SR ↑
GT	-	0.45	2.23	-	-	0.46	2.23	-
Iter0	0.22	0.41	2.59	0.56	0.21	0.39	2.60	0.36
Iter1	0.22	0.40	2.62	0.75	0.21	0.39	2.60	0.73
Iter2	0.22	0.38	2.65	0.85	0.20	0.39	2.60	0.76
Iter3	0.22	0.39	2.61	0.93	0.21	0.39	2.60	0.84

**Fig. 2:** Comparison of generated motions between the **Base Model** (the left) and **RLPF** (the right) under the same textual instructions.

In Figure 3, we evaluate the tracking performance of the generated motions using the same tracking policy in the MuJoCo simulator as a cross-simulator transfer evaluation. Under the same instructions, the Base Model tends to generate “forward-jump” and “spin-jump” motions with overly large amplitudes and excessive forward displacement, causing the tracking policy to fail. In contrast, RLPF generates motions that remain executable, achieving stable and successful tracking performance in this cross-simulator setting.

As shown in Figure 4 and the supplementary videos, under diverse and complex textual instructions, the large motion model generates motions that are both physically feasible and well aligned with the input text, and the tracking policy executes them stably and smoothly on the real humanoid robot, thereby establishing a text-motion-action pipeline for real-world humanoid control.

4.5 Ablation Analysis of RLPF

We conduct ablation studies to analyze the contribution of key components in RLPF. We compare two variants: **RLPF-w/o-T** (ablated by removing the mo-

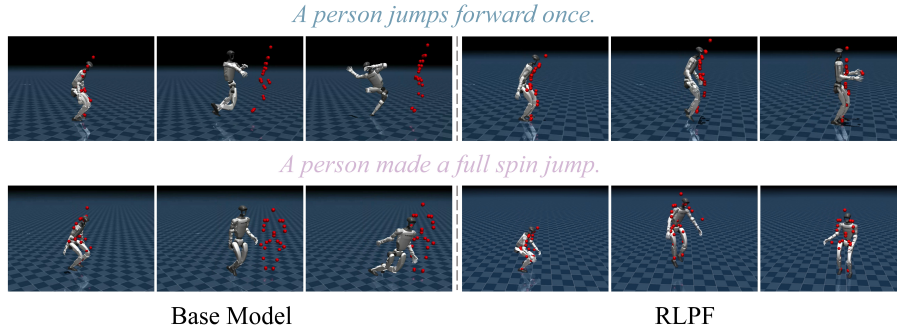


Fig. 3: Comparison between Base Model and RLPF in MuJoCo.

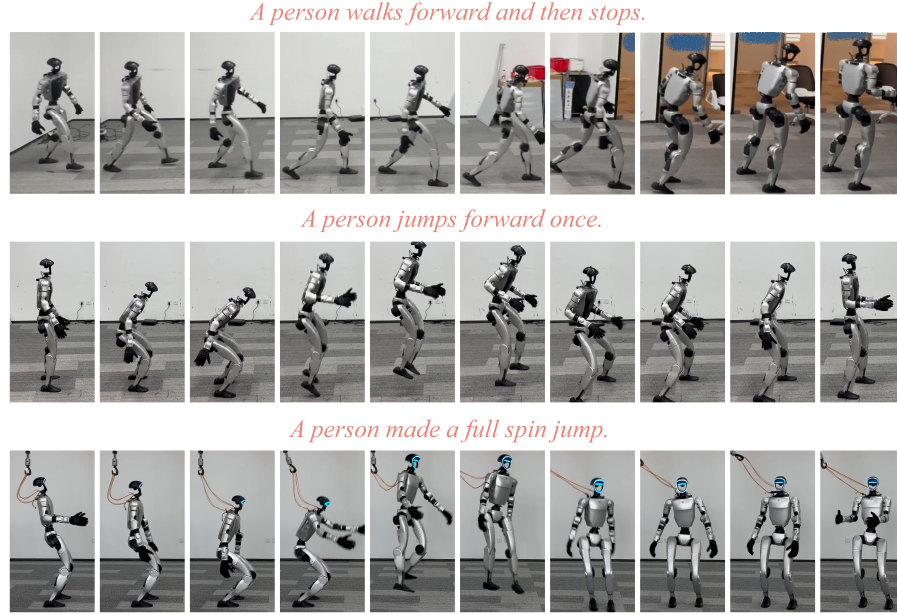


Fig. 4: Real-world deployment of RLPF given natural language instructions.

tion tracking reward), and **RLPF-w/o-A** (ablated by removing the alignment verification module).

Table 3 reveals a clear **trade-off** between physical feasibility and text semantic alignment. **(i)** Removing the tracking reward (RLPF-w/o-T) yields **slight** gains in high-level generation metrics but **substantially** degrades low-level tracking. **(ii)** In contrast, removing the alignment verification module (RLPF-w/o-A) achieves near-perfect tracking yet incurs a **complete** collapse in text-motion alignment: across different instructions, the model degenerates to trivial standing motions, as shown in the appendix visualizations.

Table 3: Ablation evaluations of RLPF.

Method	High-level			Low-level		
	FID ↓	R1 ↑	MM-D ↓	SR ↑	MPJPE ↓	MPKPE ↓
RLPF-Full	0.26	0.50	3.02	0.95	0.20	0.06
RLPF-w/o-T	0.23	0.50	2.98	0.50	0.31	0.21
RLPF-w/o-A	41.97	0.07	7.57	0.99	0.18	0.07

4.6 Sensitivity Analysis of Hierarchical RLPF

Since our text-motion-action pipeline co-trains the large motion model and the tracking policy, a natural concern is that the observed improvements might arise merely from the two components overfitting to each other. In particular, one may ask whether the performance gains persist if we change the source of motions or replace the tracking policy. To address this, we conduct a sensitivity analysis.

Table 4: Sensitivity Analysis of **Hierarchical RLPF**.

Subset	Iter0	Iter1	Iter2	Iter3
# Sensitivity of iterative tracking policies.				
MotionX-W	0.66	0.78	0.81	0.86
MotionX-J	0.21	0.53	0.56	0.69
# Sensitivity of iterative large motion models.				
MotionX-W	0.72	0.76	0.80	0.84
MotionX-J	0.31	0.45	0.51	0.57

First, to analyze the sensitivity of the tracking policy, we evaluate the policies obtained at different RLPF iterations on **unseen ground-truth motions**, rather than on motions generated by the large motion model. Specifically, we use the MotionX-W and MotionX-J test subsets and measure how accurately each iteration’s tracking policy follows these held-out reference motions.

Second, to analyze the sensitivity of the large motion model, we conduct a complementary study by pairing each iteratively refined generator with a *fixed independent* tracking policy. This policy is trained on MotionX directly and is *never* involved in the framework. We then examine the resulting text-motion-action performance under the same frozen tracker.

As shown in Table 4, both sensitivity analyses exhibit **progressive improvements across iterations**, suggesting that Hierarchical RLPF yields genuine gains in *both* the tracking policy and the large motion model, without evidence of mutual overfitting. Complete results are reported in the appendix.

5 Conclusion

In this paper, we present RLPF, a framework for improving the physical feasibility of motion generation models for humanoid robots. RLPF integrates two components, a pre-trained motion tracking policy and an alignment verification module. Building on these components, RLPF performs RL fine-tuning of a large motion model, jointly optimizing for physical feasibility and instruction alignment. To overcome the limitation of a frozen tracking policy, we further propose Hierarchical RLPF, a co-training scheme that iteratively optimizes both the large motion model and the tracking policy. Extensive experiments demonstrate the effectiveness of RLPF and Hierarchical RLPF, highlighting a pathway toward deploying text-conditioned motion generation on real-world humanoid robots.

Limitations and Future Work. Our current RLPF framework is built on an LLM-based backbone with a discrete tokenizer, whose finite codebook size may constrain the exploration space available to RL. In future work, we plan to extend RLPF to diffusion-based T2M models to examine whether a richer continuous motion representation can further improve performance.

Acknowledgements

This work was supported by NSFC under Grant 62450001 and 62476008. The authors thank the anonymous reviewers for their valuable comments.

References

1. Fan, K., Lu, S., Dai, M., Yu, R., Xiao, L., Dou, Z., Dong, J., Ma, L., Wang, J.: Go to zero: Towards zero-shot motion generation with million-scale data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13336–13348 (October 2025)
2. Fu, Z., Zhao, Q., Wu, Q., Wetzstein, G., Finn, C.: Humanplus: Humanoid shadowing and imitation from humans. arXiv preprint arXiv:2406.10454 (2024)
3. 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)
4. Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
5. Han, G., Liang, M., Tang, J., Cheng, Y., Liu, W., Huang, S.: Reindiffuse: Crafting physically plausible motions with reinforced diffusion model. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2218–2227. IEEE (2025)
6. He, T., Gao, J., Xiao, W., Zhang, Y., Wang, Z., Wang, J., Luo, Z., He, G., Sobanbab, N., Pan, C., et al.: Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. arXiv preprint arXiv:2502.01143 (2025)
7. He, T., Luo, Z., He, X., Xiao, W., Zhang, C., Zhang, W., Kitani, K.M., Liu, C., Shi, G.: Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In: Conference on Robot Learning. pp. 1516–1540. PMLR (2025)

8. Ji, M., Peng, X., Liu, F., Li, J., Yang, G., Cheng, X., Wang, X.: Exbody2: Advanced expressive humanoid whole-body control. arXiv preprint arXiv:2412.13196 (2024)
9. Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., Chen, T.: Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems* **36**, 20067–20079 (2023)
10. Lee, D., Kim, C., Kim, S., Cho, M., Han, W.S.: Autoregressive image generation using residual quantization. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 11523–11532 (2022)
11. Li, J., Cheng, X., Huang, T., Yang, S., Qiu, R.Z., Wang, X.: Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control. arXiv preprint arXiv:2505.03738 (2025)
12. Li, Z., Chi, C., Wei, Y., Zhu, B., Peng, Y., Huang, T., Wang, P., Wang, Z., Zhang, S., Xu, C.: From language to locomotion: Retargeting-free humanoid control via motion latent guidance. arXiv preprint arXiv:2510.14952 (2025)
13. Liao, Q., Truong, T.E., Huang, X., Gao, Y., Tevet, G., Sreenath, K., Liu, C.K.: Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. arXiv preprint arXiv:2508.08241 (2025)
14. Lin, J., Zeng, A., Lu, S., Cai, Y., Zhang, R., Wang, H., Zhang, L.: Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems* **36**, 25268–25280 (2023)
15. Luo, Z., Cao, J., Kitani, K., Xu, W., et al.: Perpetual humanoid control for real-time simulated avatars. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 10895–10904 (2023)
16. 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)
17. Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al.: Isaac gym: High performance gpu-based physics simulation for robot learning. arXiv preprint arXiv:2108.10470 (2021)
18. Mao, J., Zhao, S., Song, S., Shi, T., Ye, J., Zhang, M., Geng, H., Malik, J., Guizilini, V., Wang, Y.: Learning from massive human videos for universal humanoid pose control. arXiv preprint arXiv:2412.14172 (2024)
19. Mao, Y., Liu, X., Zhou, W., Lu, Z., Li, H.: Learning generalizable human motion generator with reinforcement learning. arXiv preprint arXiv:2405.15541 (2024)
20. 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)
21. Peng, X.B., Abbeel, P., Levine, S., Van de Panne, M.: Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)* **37**(4), 1–14 (2018)
22. 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)
23. Ross, S., Gordon, G., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. pp. 627–635. *JMLR Workshop and Conference Proceedings* (2011)
24. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)

25. Serifi, A., Grandia, R., Knoop, E., Gross, M., Bächer, M.: Robot motion diffusion model: Motion generation for robotic characters. In: SIGGRAPH asia 2024 conference papers. pp. 1–9 (2024)
26. Shao, Y., Huang, X., Zhang, B., Liao, Q., Gao, Y., Chi, Y., Li, Z., Shao, S., Sreenath, K.: Langwbc: Language-directed humanoid whole-body control via end-to-end learning. arXiv preprint arXiv:2504.21738 (2025)
27. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
28. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems* **28** (2015)
29. Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., Christiano, P.F.: Learning to summarize with human feedback. *Advances in neural information processing systems* **33**, 3008–3021 (2020)
30. Tessler, C., Guo, Y., Nabati, O., Chechik, G., Peng, X.B.: Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions On Graphics (TOG)* **43**(6), 1–21 (2024)
31. 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)
32. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. arXiv preprint arXiv:2209.14916 (2022)
33. Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. pp. 5026–5033. IEEE (2012)
34. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Baid, A., Baevski, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
35. Tseng, J., Castellon, R., Liu, K.: Edge: Editable dance generation from music. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 448–458 (2023)
36. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. *Advances in neural information processing systems* **30** (2017)
37. Wang, Y., Zheng, S., Cao, B., Wei, Q., Zeng, W., Jin, Q., Lu, Z.: Scaling large motion models with million-level human motions. arXiv preprint arXiv:2410.03311 (2024)
38. Wang, Y., Huang, D., Zhang, Y., Ouyang, W., Jiao, J., Feng, X., Xu, D., Tang, S.: Motiongpt-2: A general-purpose motion-language model for motion generation and understanding. *IEEE Transactions on Circuits and Systems for Video Technology* (2026)
39. Wang, Y., Yang, M., Ding, Z., Zhang, Y., Zeng, W., Xu, X., Jiang, H., Lu, Z.: From experts to a generalist: Toward general whole-body control for humanoid robots. arXiv preprint arXiv:2506.12779 (2025)
40. Xue, H., Huang, X., Niu, D., Liao, Q., Kragerud, T., Grasdahl, J.T., Peng, X.B., Shi, G., Darrell, T., Sreenath, K., et al.: Leverb: Humanoid whole-body control with latent vision-language instruction. arXiv preprint arXiv:2506.13751 (2025)
41. Yin, K., Zeng, W., Fan, K., Dai, M., Wang, Z., Zhang, Q., Tian, Z., Wang, J., Pang, J., Zhang, W.: Unitracker: Learning universal whole-body motion tracker for humanoid robots. *IEEE Robotics and Automation Letters* (2026)

42. 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)
43. Ze, Y., Chen, Z., Araújo, J.P., Cao, Z.a., Peng, X.B., Wu, J., Liu, C.K.: Twist: Tele-operated whole-body imitation system. arXiv preprint arXiv:2505.02833 (2025)
44. Zhang, J., Zhang, Y., Cun, X., Zhang, Y., Zhao, H., Lu, H., Shen, X., Shan, Y.: Generating human motion from textual descriptions with discrete representations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14730–14740 (2023)
45. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence* **46**(6), 4115–4128 (2024)
46. Zhang, M., Jin, D., Gu, C., Hong, F., Cai, Z., Huang, J., Zhang, C., Guo, X., Yang, L., He, Y., et al.: Large motion model for unified multi-modal motion generation. In: European Conference on Computer Vision. pp. 397–421. Springer (2024)