

# PhyGile: Physics-Prefix Guided Motion Generation for Agile General Humanoid Motion Tracking

Jiacheng Bao<sup>1\*</sup>, Haoran Yang<sup>1,2\*</sup>, Yucheng Xin<sup>1,3\*</sup>, Junhong Liu, Yuecheng Xu<sup>4</sup>, Han Liang<sup>5</sup>  
Pengfei Han<sup>1</sup>, Xiaoguang Ma<sup>6</sup>, Dong Wang<sup>1</sup>, Bin Zhao<sup>1†</sup>

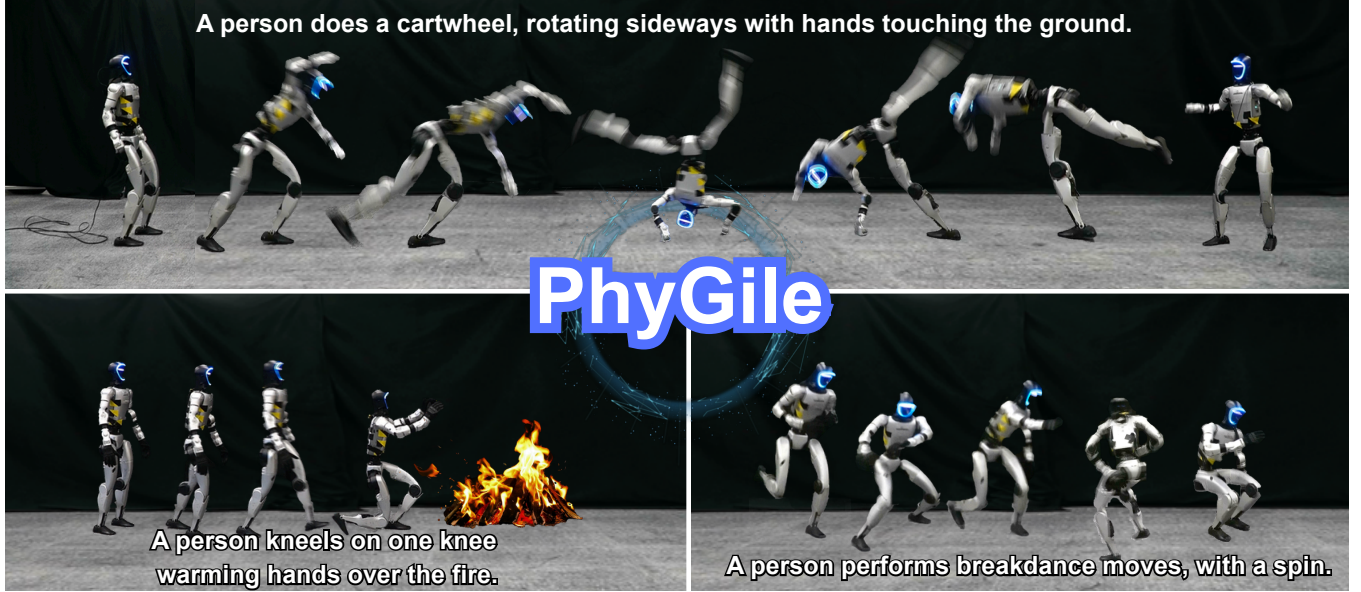


Fig. 1: **PhyGile** translates natural language commands into agile and expressive whole-body motions on humanoid robots, thereby enabling stable real-world execution of highly-difficult motions. **Project Page:** [baojch.github.io/phygile-page/](https://baojch.github.io/phygile-page/)

**Abstract**—Humanoid robots are expected to execute agile and expressive whole-body motions in real-world settings. Existing text-to-motion generation models are predominantly trained on captured human motion datasets, whose priors assume human biomechanics, actuation, mass distribution, and contact strategies. When such motions are directly retargeted to humanoid robots, the resulting trajectories may satisfy geometric constraints (e.g., joint limits, pose continuity) and appear kinematically reasonable. However, they frequently violate the physical feasibility required for real-world execution. To address these issues, we present PhyGile, a unified framework that closes the loop between robot-native motion generation and General Motion Tracking (GMT). PhyGile performs Physics-prefix-Guided robot-native motion generation at inference time, directly generating robot-native motions in a 262-dimensional skeletal space with physics-guided prefix, thereby eliminating inference-time retargeting artifacts and reducing generation–execution discrepancies. Before physics-prefix adaptation, we train the GMT controller with a curriculum-based mixture-of-experts scheme, followed by post-training on unlabeled motion data, to improve robustness over large-scale robot motions. During physics-prefix adaptation, the GMT controller is further fine-tuned with generated objectives under physics-derived prefixes, enabling agile and stable execution of complex motions on real

robots. Extensive offline and real-robot experiments demonstrate that our PhyGile expands the frontier of text-driven humanoid control, enabling stable tracking of agile, highly-difficult whole-body motions that go well beyond walking and low-dynamic motions typically achieved by prior methods.

## I. INTRODUCTION

Humanoid control is undergoing a paradigm shift from task-specific locomotion toward scalable and general-purpose motion generation. Recent advances in general motion tracking (GMT) show that a single policy can imitate large collections of reference motions and reliably transfer them to real hardware [1], [2], [3]. In parallel, text-driven motion generation has progressed rapidly, with diffusion-based models producing semantically rich and diverse human motions conditioned on natural language [4], [5], [6]. Recently, these two paradigms have begun to converge, where text-to-motion models provide high-level motion priors and GMT policies serve as low-level executors, forming a hierarchical pipeline for autonomous humanoid control [7], [8], [9].

Despite the above-mentioned successes, motion generation and physical execution remain fundamentally misaligned in practice. Most text-to-motion models are developed in the human domain and synthesize motions in standardized representations such as SMPL [10]. Running these motions on

<sup>1</sup>Shanghai AI Laboratory, <sup>2</sup>University of Science and Technology of China, <sup>3</sup>Tsinghua University, <sup>4</sup>Fudan University, <sup>5</sup>ByteDance, <sup>6</sup>Northeastern University

\*indicates equal contribution, <sup>†</sup>Corresponding author

humanoid robots requires a retargeting stage [11], [12] that maps human joint trajectories to robot morphologies. While forward-kinematics-based retargeting can approximately preserve pose structure, it does not enforce coupled physical constraints, including torque limits, contact consistency, and dynamic balance. Consequently, retargeted motions often contain physically inconsistent segments that appear kinematically plausible yet are dynamically unstable [13]. In large-scale training pipelines, such references degrade policy learning and often necessitate filtering mechanisms to discard invalid samples, reducing data efficiency and robustness.

Beyond retargeting artifacts, GMT is further constrained by severe data imbalance. Large motion datasets exhibit pronounced long-tail distributions. For example, HumanML3D [14], one of the most widely used benchmarks for text-driven motion modeling, contains abundant simple motions such as walking and running, whereas complex and agile motions are comparatively rare. When training a general tracking policy on such skewed distributions, optimization naturally prioritizes frequent and low-difficulty motions. Rare and high-difficulty skills that require tight coordination and dynamic stability remain undertrained, leading to fragile performance when robots attempt agile motions.

Recent humanoid motion generation approaches attempt to reduce reliance on human-first retargeting by aligning language with robot-controllable embeddings [8], [15] or directly modeling robot-native motion data [7]. While these methods strengthen the connection between semantics and embodiment, generation and tracking are typically optimized as loosely coupled modules: a generator proposes trajectories, and a tracker attempts to follow them. When generated motions exceed the feasible region of the current controller, execution usually fails unless a mechanism is available to reconcile generation with control.

In this work, we propose **PhyGile**, a physics-prefix-guided framework that couples robot-native motion generation with agile GMT, enabling high-performance tracking and generation within a single, physically grounded pipeline. With the motion generator frozen, **PhyGile** leverages physics-validated prefixes as a shared interface to improve motion quality and GMT agility, leading to more physically consistent and task-effective motions at execution time.

On the generation side, we introduce a diffusion-based motion model operating in a 262D robot-skeleton space, directly synthesizing robot-native joint trajectories. To enhance compositional language grounding, we incorporate a Token-level Parameter-mixing Mixture-of-Experts (TP-MoE), enabling fine-grained alignment between individual text tokens and the motion timeline, so that temporally localized semantic cues correspond to appropriate motion segments.

On the control side, we address the long-tail challenge of agile motions through a curriculum mixture-of-experts training scheme. In the first stage, we perform explicit curriculum learning on difficulty-annotated motion data, where motions are stratified by complexity and experts specialize across skill levels. Error-aware adaptive sampling progressively increases exposure to high-difficulty motions, improving robustness

on rare and highly dynamic skills. In the second stage, we conduct global soft-moe post-training on large-scale unlabeled motions to enhance overall generalization while preserving the agility acquired during curriculum learning.

To bridge generation and execution, we further introduce a physics-prefix-guided fine-tuning stage. Dynamically feasible motion segments extracted from the tracking policy are injected as conditioning prefixes into the diffusion process, anchoring the initial denoising states to the robot’s motion manifold. During this stage, the diffusion model remains frozen, while the GMT controller is fine-tuned under prefix-conditioned motion generation to improve tracking performance on generated motion distributions, thereby tightening the coupling between generation and execution.

By integrating difficulty-aware tracking, robot-native diffusion generation, and physics-prefix-guided alignment into a unified pipeline, **PhyGile** enables the generation and execution of agile humanoid motions that are both semantically aligned and physically realizable. Extensive offline and real-robot experiments demonstrate that **PhyGile** effectively resolves the semantics–physics trade-off in generation, establishing a highly robust and unified tracking policy. Notably, for highly dynamic agile motions where prior pipelines fail, our framework rapidly unlocks stable hardware execution through lightweight fine-tuning, demonstrating strong scalability toward complex whole-body skills.

In summary, **PhyGile**’s main contributions are as follows:

- We propose **PhyGile**, a physics-prefix-guided framework for general humanoid motion tracking that couples robot-native diffusion-based motion generation with GMT. By closing the loop via physics-consistent prefixes, **PhyGile** mitigates the mismatch between text-driven motion synthesis and real-world control, enabling agile and dynamically feasible humanoid motions.
- We develop three key components: (i) a curriculum-based MOE training strategy for agile GMT with motion-difficulty stratification; (ii) a 262D robot-skeleton representation with TP-MoE-enhanced motion generation for fine-grained text conditioning; and (iii) a physics-prefix-guided fine-tuning stage that aligns generated motions with executable tracking policies.
- Through extensive offline benchmarks and real-robot experiments, we demonstrate that **PhyGile** effectively resolves the semantics–physics trade-off in generation. Furthermore, our robust, unified tracking policy achieves stable hardware execution of highly dynamic humanoid motions, successfully unlocking complex agile behaviors that prior pipelines fail to realize even with additional tuning.

## II. RELATED WORKS

### A. Text-driven Human Motion Generation

Text-driven motion generation aims to learn the conditional distribution of human motion given natural-language descriptions. Early approaches adopt VAE-style formulations or autoregressive modeling with discrete motion representations [16], [17], [18], [19]. Recent diffusion-based

methods [4], [5], [6], [20] reframe text-to-motion as conditional denoising, improving generative quality and controllability. Follow-up works further enhance efficiency and scalability via masked modeling and latent-consistency acceleration [21], as well as improve diversity and long-horizon synthesis through retrieval or compositional generation [22], [23]. Complementary directions incorporate cross-modal alignment or physical constraints to strengthen semantic faithfulness and realism [24], [25], [26].

### B. Humanoid Motion Generation and Control

Humanoid motion generation and control can be broadly categorized into motion synthesis, robot-native imitation learning, and vision-language-action (VLA) policies. A common paradigm generates human motions conditioned on language and retargets them to humanoid morphologies, decoupling semantics from embodiment [27], [28]. To improve scalability, recent methods align language with structured latent or discrete action spaces, enabling retargeting-free or tokenized generation [29], [30], [31]. In parallel, robot-native imitation learning and VLA approaches directly learn executable policies from motion, demonstration, or egocentric visual inputs, reducing reliance on explicit motion synthesis and bridging perception, reasoning, and control [7], [8], [15], [32], [33]. However, such approaches still face challenges in motor control and long-horizon consistency.

General motion tracking for humanoid control casts reference following as scalable motion imitation, producing unified policies that cover diverse behaviors and transfer to hardware [1], [2], [3], [31], [34], [35]. To mitigate interference across heterogeneous motions, prior work leverages mixture-of-experts, expert partitioning with consolidation, or multi-behavior distillation [1], [2], [36], [37], [38]. Shared latent representations over motion, goals, and rewards further support promptable reuse and planning [39], [40], and these trackers can serve as low-level backbones for language-conditioned control [7], [31]. However, highly dynamic and agile behaviors remain challenging due to rapid coordination and contact transitions that amplify tracking instability.

## III. METHODS

**PhyGile** couples robot-native diffusion generation with agile GMT through physics-prefix guidance. It comprises (i) a two-stage MoE tracker trained with curriculum-constrained routing to cover long-tail agile motions, (ii) a TP-MoE-conditioned robot-native diffusion generator, and (iii) physics-prefix guided adaptation that validates and refines generated segments for dynamic feasibility, tightening consistency between generation and executable tracking.

### A. General Motion Tracking

Our training datasets contain approximately 45 hours of motion, including text-annotated sequences from HumanML3D [14] and unlabeled MoCap data from AMASS [41], LaFAN1 [42], plus a private 3-hour MoCap set retargeted to our robot embodiment using GMR [11].

**Data Curriculum.** We interpret motion difficulty from a neuro-control view as the *coordination load* required for planning, prediction, and stabilization (i.e., increasing coupling and dynamical instability). Using LLM-based semantic analysis of HumanML3D text descriptions, we assign each clip to one of 12 ordered levels. Levels 1–10 are feasible in our embodiment and are grouped only for exposition as *Easy* (1–4), *Medium* (5–7), and *Hard* (8–10), while Levels 11–12 are excluded due to environment/physics mismatch.

**Level 1: [Easy]** Near-static poses with minimal planning and balance demands (e.g., standing, clapping).

**Levels 2–4: [Easy]** Automatic locomotion and basic motor patterns with stable CoM dynamics (e.g., walking, bending).

**Levels 5–7: [Medium]** Coordinated transitions and directional or speed changes requiring predictive stabilization (e.g., jogging, walking backward, kicking).

**Levels 8–10: [Hard]** Highly dynamic or composite skills with tightly coupled planning and balance (e.g., rolling, jumping, spinning, crawling).

**Level 11: [Incompatible]** Motions dependent on external terrain or elevation changes (e.g., stair climbing).

**Level 12: [Infeasible]** Motions physically unrealizable in the current simulation setup (e.g., swimming, flying).

**Freeze-and-drop self-purification.** During training, for each motion file  $i$ , we maintain an EMA tracking error  $E_i$  and success-rate estimate  $\hat{p}_i^{\text{succ}}$  (fraction of rollouts whose tracking succeeds). A motion is temporarily frozen once it has been sufficiently exposed, but remains untrackable:

$$(E_i \geq \tau_{\text{err}} \vee \hat{p}_i^{\text{succ}} \leq \tau_{\text{succ}}) \wedge n_i \geq n_{\min}, \quad (1)$$

where  $n_i$  is the number of rollouts sampled from file  $i$ ,  $\tau_{\text{err}}$  and  $\tau_{\text{succ}}$  are error/success thresholds, and  $n_{\min}$  is the minimum exposure before freezing is allowed. A file is dropped if it repeatedly triggers freezing.

**Look-ahead motion encoding.** The policy conditions on multi-scale future motion: a short look-ahead window provides per-frame future targets for accurate near-term tracking, while a longer downsampled horizon is encoded into a compact latent  $\mathbf{z}_t$  via temporal convolution and pooling.  $\mathbf{z}_t$  captures upcoming speed changes, turns, and contact events, and is also used as the *only* input to the MoE gate.

**Two-stage training with MoE routing.** Our actor is a MoE policy with  $K$  expert MLPs  $\{E_1, \dots, E_K\}$  and a lightweight gate  $G$  that maps  $\mathbf{z}_t$  to routing logits  $\ell = G(\mathbf{z}_t) \in \mathbb{R}^K$ . To balance routing stability and compute, we use low-frequency soft top- $k$  routing: every  $M$  steps we refresh a candidate expert set  $\mathcal{K} = \text{top-}k(\ell)$  (thus  $|\mathcal{K}| = k$ ), and between refreshes compute temperature- $\tau$  softmax weights only over candidates. The action is a convex mixture of candidate expert outputs,

$$\mathbf{a}_t = \sum_{j \in \mathcal{K}} p_j E_j(\tilde{\mathbf{o}}_t), \quad (2)$$

where  $\tilde{\mathbf{o}}_t$  denotes the observation vector and  $p_j = \text{softmax}(\ell_{\mathcal{K}}/\tau)_j$  are normalized mixture weights computed from candidate logits  $\ell_{\mathcal{K}}$ ; this requires only  $k$  expert forward

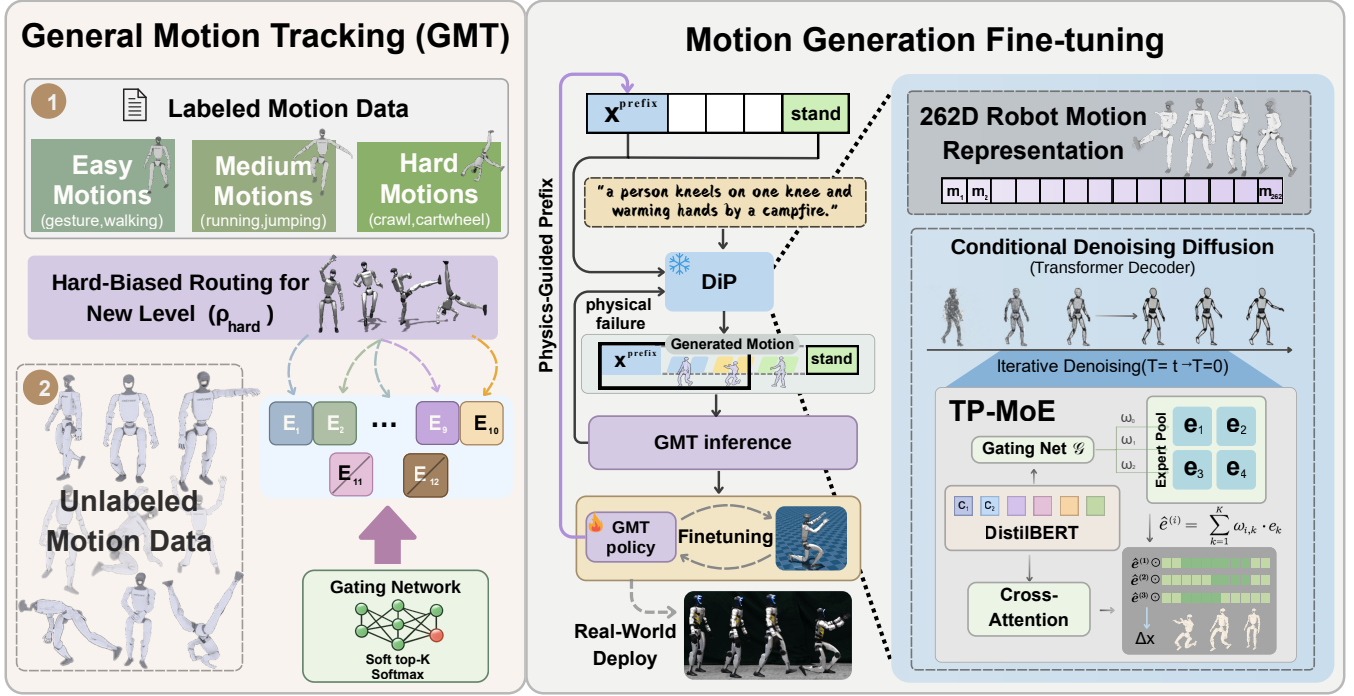


Fig. 2: **Overview of PhyFile.** (Left) **GMT:** A two-stage MoE tracker is first trained with curriculum-constrained routing to induce expert specialization, followed by global soft post-training with dynamic expert expansion to absorb persistently difficult motions. (Right) **Generation of Diffusion Policy:** A TP-MoE-conditioned robot-native diffusion model generating 262D robot motion sequences from text. (Center) **Motion Generation Fine-tuning:** Executable motion prefixes are concatenated with newly generated 1-second continuations and validated by pretrained GMT. Closed-loop simulation refinement further enforces dynamic feasibility and improves consistency between generated and trackable motions, and the fine-tuned GMT policy is deployed on real robots.

passes per step. We optionally apply EMA smoothing to  $\ell$  to reduce routing jitter.

*a) Stage I: level-wise curriculum with hard-biased routing:* We train level-by-level from  $l_{\max} = 1$  to 10, progressively unlocking higher-difficulty motion levels to induce expert specialization. At the file level, we use difficulty-aware in-batch sampling with uniform exploration, while gradually introducing newly unlocked levels and maintaining a minimum replay quota for earlier levels.

We enforce *hard-biased routing* so the newly unlocked expert receives strong gradients. When the curriculum is at level  $l_{\max}$ , routing is restricted to  $\{E_1, \dots, E_{l_{\max}}\}$  by masking locked experts. Moreover, for samples drawn from the current hardest level ( $l_i = l_{\max}$ , where  $l_i$  is the assigned level of file  $i$ ), we bypass the gate with probability  $\rho_{\text{hard}} = 0.8$  and hard-route to the expert  $E_{l_{\max}}$ :

$$\mathbf{a}_t = \begin{cases} E_{l_{\max}}(\tilde{\mathbf{o}}_t), & l_i = l_{\max} \wedge u < \rho_{\text{hard}}, \\ \sum_{j \in \mathcal{K}} p_j E_j(\tilde{\mathbf{o}}_t), & \text{otherwise,} \end{cases} \quad (3)$$

where  $u \sim \text{Uniform}(0, 1)$  and  $\rho_{\text{hard}}$  is the hard-routing probability. Upon level promotion, the new expert is initialized by copying the preceding expert ( $\theta_{E_l} \leftarrow \theta_{E_{l-1}}$ ) to stabilize optimization. To align motion semantics with routing, we jointly optimize the gate with an auxiliary cross-entropy loss:

$$\mathcal{L}_{\text{route}} = \lambda_{\text{CE}} \cdot \text{CE}(G(\mathbf{z}_t), l_i - 1), \quad (4)$$

where  $\text{CE}(\cdot, \cdot)$  is the cross-entropy loss,  $(l_i - 1)$  is the level-to-expert label, and  $\lambda_{\text{CE}}$  is its weight.

*b) Stage II: global soft post-training:* We remove curriculum masks and hard-routing constraints, allowing all  $K$  experts to access the full dataset and enabling end-to-end optimization with differentiable soft top- $k$  routing. To avoid routing collapse, we replace strong level supervision with a load-balancing objective:

$$\mathcal{L}_{\text{bal}} = K \sum_{j=1}^K f_j \bar{p}_j, \quad (5)$$

where  $f_j = \mathbb{E}[\mathbb{I}(\arg \max_k p_k = j)]$  is the fraction of samples whose top-1 route selects expert  $j$  and  $\bar{p}_j = \mathbb{E}[p_j]$  is the mean probability mass assigned to expert  $j$  (both estimated over minibatches), encouraging uniform utilization; a low-weight cross-entropy regularizer can be retained to preserve the semantic prior. To handle distribution shifts in unlabeled data, we additionally support *dynamic expert addition*: we track per-file EMA statistics including routing entropy  $\mathcal{H}(p) = -\sum_j p_j \log p_j$  and the top-1/top-2 gap  $\Delta = p_{(1)} - p_{(2)}$  with  $p_{(1)} \geq p_{(2)}$ , and spawn a new expert when a sufficient fraction of files remains persistently difficult. We cap its initial routing mass (cold-start) and use a higher learning rate for the new expert and a lower learning rate for old experts to accelerate adaptation without forgetting.

## B. Motion Generation

For motion generation, we use filtered text-annotated motion sequences from HumanML3D [14] that are retargeted to our robot embodiment for training. The paired motion–text data provide fine-grained language supervision for training the robot-native diffusion model. A 262-dimensional per-frame motion descriptor for the robot body is defined as  $m_t \in \mathbb{R}^{262}$ :

$$m_t = [\dot{\omega}_t^{\text{root}}, \dot{v}_t^{\text{root}}, z_t, p_t^{\text{ric}}, R_t^{6d}, \dot{p}_t^{\text{local}}, c_t^{\text{foot}}, c_t^{\text{hand}}]. \quad (6)$$

Here,  $\dot{\omega}_t^{\text{root}} \in \mathbb{R}^3$  denotes the root angular velocity,  $\dot{v}_t^{\text{root}} \in \mathbb{R}^3$  the root linear velocity, and  $z_t \in \mathbb{R}$  the root height. The term  $p_t^{\text{ric}} \in \mathbb{R}^{36}$  concatenates the local positions of 12 rigid bodies (end-effectors, knees),  $R_t^{6d} \in \mathbb{R}^{174}$  concatenates the 6D rotation representations of 29 rigid bodies (corresponding to the robot’s 29 DOF), and  $\dot{p}_t^{\text{local}} \in \mathbb{R}^{39}$  concatenates the local velocities of 13 rigid bodies (end-effectors, knees, root). Finally,  $c_t^{\text{foot}} \in \{0, 1\}^4$  and  $c_t^{\text{hand}} \in \{0, 1\}^2$  are binary contact indicators for feet and hands, respectively. All features are extracted under a canonical heading (the first frame faces  $+X$ ). Since the elements of  $R_t^{6d}$  naturally lie in  $[-1, 1]$ , no additional normalization is applied in order to preserve the geometric semantics of rotations.

A conditional denoising diffusion framework is adopted. DistilBERT [43] encodes the text into token embeddings  $\{c_i\}_{i=1}^N$ , and the denoising network is implemented as an AdaLN Transformer Decoder, where the diffusion timestep is injected via AdaLN. The training objective is

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{m_0, t, \epsilon} [\|m_0 - \hat{m}_\theta(m_t, t, l)\|^2], \quad (7)$$

where  $t$  denotes the diffusion timestep (distinct from the frame index in  $m_t$ ),  $\epsilon$  is the injected noise, and  $l$  is the conditioning signal derived from the text.

**Token-level Parameter-mixing Mixture of Experts (TP-MoE).** TP-MoE is inserted after the FFN of each decoder layer to enable fine-grained alignment between individual text tokens and the motion timeline. An expert pool  $\{e_1, \dots, e_K\}$  is maintained. For each text token embedding  $c_i$ , a gating network produces expert weights, and experts are mixed in parameter space:

$$\omega_i = \text{softmax}(\mathcal{G}(c_i)), \quad \hat{e}^{(i)} = \sum_{k=1}^K \omega_{i,k} \cdot e_k, \quad (8)$$

where  $\omega_i \in \mathbb{R}^K$  denotes the expert weights for token  $c_i$ ,  $e_k$  is the  $k$ -th expert (a two-layer FFN), and  $\hat{e}^{(i)}$  is the resulting token-specific mixed expert.

After transforming the motion features, a spatial mask is applied based on the cross-attention weights  $A$ :

$$M_{t,i} = \sigma\left(\gamma(A_{t,i} - \beta \cdot \max_{t'} A_{t',i})\right), \quad \Delta x = \sum_i M_i \odot \hat{e}^{(i)}(x), \quad (9)$$

where  $A_{t,i}$  is the attention weight from token  $i$  to frame  $t$ ,  $\gamma$  and  $\beta$  control mask sharpness and thresholding,  $\sigma(\cdot)$  is the sigmoid function, and  $\odot$  denotes element-wise multiplication. For notational convenience,  $M_i$  stacks  $\{M_{t,i}\}_t$

along the temporal dimension. The resulting update  $\Delta x$  is injected into the backbone through a residual connection. During training, a load-balancing loss  $\mathcal{L}_{\text{bal}}$  is introduced to prevent expert collapse.

**Action-Semantic Frequency-aware Oversampling (ASFO).** To address the long-tailed distribution of motion data, ASFO leverages an LLM to extract a set of action-semantic tags  $\mathcal{K}$  from the text annotations. For each tag, its empirical frequency is computed as  $f_m$ , and the median frequency  $\tau = \text{median}(\{f_m\})$  is used as the target to derive an oversampling multiplier:

$$\rho_m = \min(\lfloor \tau / f_m \rfloor, \rho_{\text{max}}), \quad (10)$$

where  $\rho_{\text{max}}$  caps the multiplier to limit overfitting. For a multi-label sample  $x_j$ , the effective multiplier is set to

$$r_j = \max_{k_m \in \phi(x_j)} \rho_m, \quad (11)$$

where  $\phi(x_j)$  returns the tag set associated with  $x_j$ . To further enhance diversity for scarce semantics without inflating already frequent actions, we augment data via left–right mirroring *only* for rare-tag samples: for each draw of  $x_j$ , we mirror joint channels by swapping left/right counterparts (and contact signals when applicable) and use the mirrored sample with a rarity-dependent probability increasing with  $r_j$  (e.g.,  $p_{\text{mir}}(x_j) = \min(\alpha(r_j - 1), 1)$ ), while keeping the original otherwise; if side-specific tags are present, we swap left/right tags accordingly. Overall, this strategy ensures that rare actions receive strong and diverse training signals.

## C. Physics-Prefix-Guided Motion Generation Fine-tuning

**Physics-Guided Prefix Conditioning.** To improve dynamic feasibility during diffusion sampling, we condition the motion generator on a physically executable prefix. At inference time, a motion segment  $x^{\text{prefix}}$  fine-tuned by the GMT simulator is concatenated with the desired terminal constraint  $x^{\text{target}}$  (e.g., a standing pose) and provided as conditioning context to the diffusion model:

$$x_{1:T} \sim p_\theta(x_{1:T} \mid x^{\text{prefix}}, x^{\text{target}}). \quad (12)$$

Anchoring the denoising process around a dynamically consistent prefix steers sampling toward locally feasible regions of the state space and mitigates unstable rollouts.

**Closed-Loop Refinement and Diffusion Fine-tuning.** We validate each generated motion in simulation using our trained GMT tracker and measure the mean per-joint position error (MPJPE), i.e., the average Euclidean distance between tracked and target joint positions over joints and time. Trajectories whose MPJPE exceeds a tolerance are rejected and resampled, while feasible rollouts are retained. This generate–simulate–select loop suppresses common failure modes such as penetration, loss of balance, and long-horizon drift by explicitly filtering dynamically inconsistent samples.

To further align the generative distribution with physically executable motion, we perform progressive fine-tuning after our pretrained GMT by iteratively applying PPO updates to the Stage-II GMT policy on prefix-conditioned closed-loop



TABLE I: Comparison of motion generation methods under the retarget setting.  $^\dagger$  denotes evaluation under the retarget setting.  $\uparrow$  /  $\downarrow$  indicate higher/lower is better;  $\rightarrow$  denotes a reference metric **Bold** and underlined values denote the best and second-best results, respectively. Results are reported as mean  $\pm$  standard deviation over five generator rollout seeds.

| Motion Generation            | FID $\downarrow$           | R@3 $\uparrow$             | MM-Dist $\downarrow$       | Diversity $\uparrow$       | Penetration (mm) $\downarrow$ | Floating (mm) $\rightarrow$ | Skating $\downarrow$ |
|------------------------------|----------------------------|----------------------------|----------------------------|----------------------------|-------------------------------|-----------------------------|----------------------|
| GT $^\dagger$ (HumanML) [14] | 0.064 $\pm 0.0058$         | 0.7812 $\pm 0.0207$        | 1.144 $\pm 0.0105$         | 1.406 $\pm 0.0141$         | 0.03                          | 23.64                       | 5.15%                |
| T2M-GPT $^\dagger$ [19]      | 0.3782 $\pm 0.0094$        | 0.5234 $\pm 0.0070$        | 1.3859 $\pm 0.0024$        | 1.2501 $\pm 0.0086$        | 1.14                          | 163.80                      | 2.77%                |
| MLD $^\dagger$ [6]           | 0.4060 $\pm 0.0092$        | 0.4962 $\pm 0.0055$        | 1.4067 $\pm 0.0082$        | 1.2054 $\pm 0.0076$        | 25.11                         | 0.00                        | 34.87%               |
| MDM $^\dagger$ [4]           | <u>0.2550</u> $\pm 0.0065$ | 0.6156 $\pm 0.0172$        | <b>1.3143</b> $\pm 0.0037$ | <b>1.3355</b> $\pm 0.0049$ | 5.12                          | 64.49                       | 19.16%               |
| MotionGPT $^\dagger$ [18]    | 0.2963 $\pm 0.0089$        | 0.6023 $\pm 0.0023$        | 1.3651 $\pm 0.0050$        | 0.8538 $\pm 0.0023$        | 1.49                          | 16.28                       | 9.49%                |
| CloSD $^\dagger$ [25]        | 0.3165 $\pm 0.0066$        | <b>0.6208</b> $\pm 0.0124$ | 1.3329 $\pm 0.0029$        | <u>1.2580</u> $\pm 0.0069$ | 3.41                          | 45.32                       | 20.01%               |
| CloSD-Physics $^\dagger$     | 0.3740 $\pm 0.0104$        | 0.5356 $\pm 0.0065$        | 1.4066 $\pm 0.0017$        | 1.1503 $\pm 0.0056$        | <u>0.82</u>                   | 15.28                       | <b>1.21%</b>         |
| TextOp [7]                   | 0.3074 $\pm 0.0011$        | 0.4975 $\pm 0.0039$        | 1.4009 $\pm 0.0013$        | 0.7467 $\pm 0.0022$        | <b>0.00</b>                   | 26.28                       | 7.5%                 |
| Ours w/o TP-MOE              | 0.2297 $\pm 0.0069$        | 0.5276 $\pm 0.0067$        | 1.3857 $\pm 0.0087$        | 1.0522 $\pm 0.0124$        | 2.42                          | 24.36                       | 8.7%                 |
| Ours                         | <b>0.1823</b> $\pm 0.0082$ | <u>0.6176</u> $\pm 0.0063$ | <u>1.3302</u> $\pm 0.0033$ | 1.1147 $\pm 0.0082$        | 3.24                          | 32.43                       | 8.2%                 |
| Ours [Fine-tuned]            | 0.2017 $\pm 0.0021$        | 0.5702 $\pm 0.0055$        | 1.3659 $\pm 0.0041$        | 1.1047 $\pm 0.0023$        | <b>0.00</b>                   | 19.39                       | <u>1.58%</u>         |

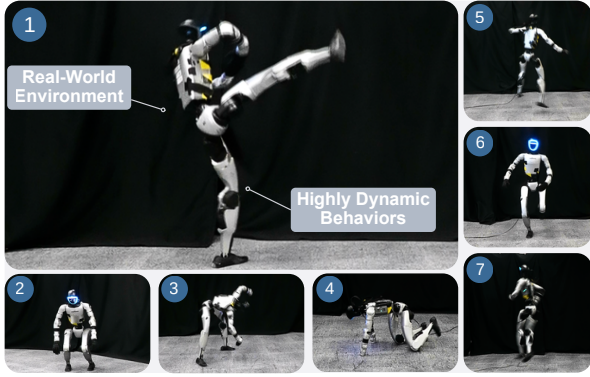


Fig. 3: Qualitative results on real robots demonstrating agile, whole-body motion generation across diverse behaviors.

rollouts, while keeping the diffusion generator frozen. At each iteration, conditioned on the current prefix, we sample a feasible 1-second continuation, and the simulator then performs end-to-end refinement on the concatenated window consisting of the prefix and the newly generated 1-second segment. We append the refined continuation to the prefix and repeat this receding-horizon procedure until reaching the designated suffix. By progressively extending the conditioned context under closed-loop filtering, the denoising process remains stable, and the resulting motions stay kinematically plausible and dynamically consistent over long horizons. Finally, we deploy the physically executable motions together with our fine-tuned GMT controller to the physical robot, achieving robust tracking of agile whole-body behaviors.

#### IV. EXPERIMENTS

We evaluate **PhyGile** through comprehensive offline and real-robot experiments, demonstrating stable tracking and execution of semantically aligned, high-difficulty agile humanoid motions. Additional qualitative results are provided in Fig. 3 and the accompanying video.

##### A. Datasets and Evaluation Metrics

**Datasets.** The motion generation model is evaluated on a robot-retargeted version of the standard HumanML3D [14] test split. The General Motion Tracking (GMT) policy is assessed on a retargeted AMASS [41] test set to examine the controller’s ability to track highly dynamic human motions.

**Evaluation Metrics.** We comprehensively evaluate our framework across both semantic generation quality and physical execution fidelity. The metrics are divided into two categories: Robot Motion Generation and Motion Tracking.

For robot motion generation, we evaluate semantic quality and physical realism. (1) **FID** measures feature-distribution fidelity between generated and real motions. (2) **R@3** evaluates text–motion alignment via Top-3 retrieval accuracy in the shared feature space. (3) **MM-Dist** quantifies cross-modal consistency by the distance between text and motion embeddings. (4) **Diversity** is computed as feature-level variance to reflect generation richness. (5) **Penetration** reports robot–environment interpenetration depth (mm). (6) **Floating** measures unintended airborne height (mm). (7) **Skating** measures tangential foot slippage during planted contacts. All generation metrics are trained in the same robot representation space (3 root translations + 4D root orientation + 29 DoF).

For Motion Tracking Evaluation, we quantify tracking accuracy and execution stability. (1) **MPJPE** ( $E_{mpjpe}$ , m) measures mean per-joint position error. (2) **MPJAE** ( $E_{mpjae}$ , rad) measures mean per-joint angle error. (3) **MPJVE** ( $E_{mpjve}$ , rad/s) measures mean per-joint velocity error. (4) **Success Rate** is the fraction of episodes without terminal failures: pelvic  $Z$  deviation  $> 0.3$ m, trunk gravity-projection difference  $> 0.8$ , or end-effector  $Z$  error  $> 0.3$ m.

##### B. Offline Evaluation

1) *Motion Generation Evaluation:* We compare PhyGile against state-of-the-art baselines across two paradigms: human motion generation models that map generated motions to the robot via General Motion Retargeting [11], and robot

TABLE II: Quantitative evaluation on General Motion Tracking. Each component of our two-stage curriculum contributes to robust motion tracking, and the full **PhyGile** pipeline achieves the most stable and reliable execution.

| Method         | $E_{mpjpe} \downarrow$ | $E_{mpjae} \downarrow$ | $E_{mpjve} \downarrow$ | Success $\uparrow$ |
|----------------|------------------------|------------------------|------------------------|--------------------|
| GMT [1]        | 0.6711                 | 0.1098                 | 0.6080                 | <u>0.8914</u>      |
| TextOp [7]     | <b>0.2427</b>          | 0.0927                 | <u>0.4824</u>          | 0.8888             |
| PhyGile-C      | 0.4960                 | 0.0910                 | 0.4948                 | 0.8537             |
| PhyGile-CF     | 0.4322                 | 0.0920                 | 0.4995                 | 0.8629             |
| PhyGile-CFM    | 0.4523                 | <u>0.0873</u>          | 0.5014                 | 0.8826             |
| <b>PhyGile</b> | <u>0.2566</u>          | <b>0.0720</b>          | <b>0.4222</b>          | <b>0.9401</b>      |

motion generation approaches (e.g., TextOp [7]) that synthesize motion directly within the robot’s kinematic space.

Table I shows a clear semantics–physics trade-off under re-targeting. Human-motion generators inherit strong alignment and diversity from large human datasets, but often produce severe physical artifacts (penetration/floating/skating) after re-targeting due to embodiment mismatch. Robot-space methods like TextOp and our **PhyGile** reduce these violations, yet the tighter feasible set induced by kinematic, balance, and contact constraints can hurt text alignment and diversity.

**PhyGile** resolves this tension by generating directly in robot space with TP-MOE: *Ours* achieves the best distribution matching (FID) while maintaining strong semantic fidelity (R@3, MM-Dist) and plausible physics. With simulator-based physical fine-tuning, *Ours [Fine-tuned]* further improves contact safety (penetration/skating) via simulator fine-tuning, at a minor cost to retrieval and distribution metrics. Removing TP-MOE (*Ours w/o TP-MOE*) degrades alignment and distribution matching, and Fig. 4(a) shows moderate expert activation in our TP-MOE is optimal (R@3 saturates around top- $k=6$ ).

2) *Motion Tracking Evaluation*: We evaluate the robustness of our motion tracking controller against the GMT [1] and TextOp [7] baselines, and conduct a systematic ablation study to examine the effect of each component in our two-stage training curriculum.

As summarized in Table II, the complete pipeline (**PhyGile**) achieves the best overall robustness on the test set, attaining the highest success rate and improved stability over the standard GMT [1] baseline. Moreover, **PhyGile** attains the lowest angular and velocity errors among all compared methods. Although TextOp [7] achieves a lower position error, the two methods exhibit different optimization profiles, with **PhyGile** showing stronger robustness and more consistent performance across metrics.

To isolate the contribution of each module, we introduce several variants. **PhyGile-C** employs only difficulty-stratified curriculum learning, yielding a clear improvement over GMT in both tracking accuracy. **PhyGile-CF** further incorporates the freeze-and-drop self-purification mechanism to filter physically incompatible reference motion, leading to additional gains in stability and overall performance. **PhyGile-CFM** introduces a Mixture-of-Experts (MoE) design during

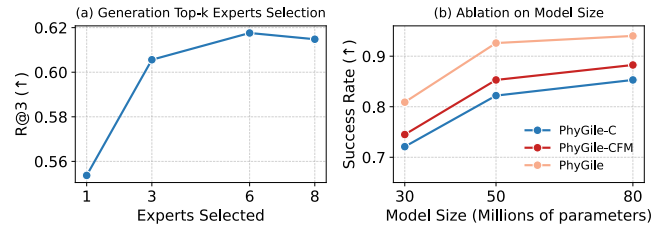


Fig. 4: Ablation on key design choices. (a) Generation module: varying the top- $k$  selected experts improves performance up to  $k=6$  (peak R@3), with a slight drop at  $k=8$ . (b) GMT module: increasing module size consistently raises the success rate; the full **PhyGile** outperforms **PhyGile-C** and **PhyGile-CFM** across all sizes.

Stage I, enabling specialization across difficulty tiers and improving the success rate relative to the curriculum-only setting. Finally, **PhyGile** applies global soft-MoE fine-tuning on unlabeled data in Stage II, consistently strengthening robustness and reducing joint errors. Moreover, scaling up the GMT module further boosts the success rate for all variants, while the full **PhyGile** remains the best across all sizes (Fig. 4(b)).

### C. Real-world Deployment

We deploy our system on a Unitree G1 humanoid robot with 29 degrees of freedom. The fine-tuned general motion tracking policy runs at 50 Hz using ONNX Runtime, taking proprioceptive states together with the current reference motion to output joint-level commands in real time. For deployment, we use the physics-prefix-guided, fine-tuned GMT controller to track a diverse set of agile, high-difficulty whole-body motions on hardware. All trajectories are first verified to be physically executable via sim-to-sim validation and are then streamed as references to the controller for real-time tracking. Notably, a single fine-tuned GMT policy generalizes across a wide range of complex behaviors without per-skill retraining, enabling stable and robust execution.

## V. CONCLUSION

In this paper, we introduced **PhyGile**, a physics-grounded framework for text-conditioned agile motion generation and execution on humanoid robots. The proposed framework couples a language-driven diffusion generator in robot-skeleton space with a general motion tracker, using physics-validated motion prefixes as a principled interface between them. By anchoring sampling with executable prefixes and fine-tuning the tracking policy with the generated motion clips, it narrows the gap between semantic intent and dynamic feasibility, yielding robust performance on challenging agile behaviors. Comprehensive evaluations in simulation and on hardware show that **PhyGile** achieves stronger text–motion consistency and more stable, higher-success execution than prior pipelines, establishing an effective route from open-vocabulary language commands to physically realizable whole-body skills.

## REFERENCES

- [1] Z. Chen, M. Ji, X. Cheng, X. Peng, X. B. Peng, and X. Wang, "Gmt: General motion tracking for humanoid whole-body control," *arXiv preprint arXiv:2506.14770*, 2025.
- [2] C. Yang, Y. Sun, P. Ye, X. Chen, C. Yu, and T. Chen, "Egm: Efficiently learning general motion tracking policy for high dynamic humanoid whole-body control," *arXiv preprint arXiv:2512.19043*, 2025.
- [3] Z. Wang, J. Wang, S. Yao, Y. Zhang, Z. Ding, M. Yang, Y. Wang, H. Jiang, C. Ma, X. Shi, *et al.*, "General humanoid whole-body control via pretraining and fast adaptation," *arXiv preprint arXiv:2602.11929*, 2026.
- [4] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano, "Human motion diffusion model," *arXiv preprint arXiv:2209.14916*, 2022.
- [5] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu, "Motiondiffuse: Text-driven human motion generation with diffusion model," *IEEE transactions on pattern analysis and machine intelligence*, vol. 46, no. 6, pp. 4115–4128, 2024.
- [6] X. Chen, B. Jiang, W. Liu, Z. Huang, B. Fu, T. Chen, and G. Yu, "Executing your commands via motion diffusion in latent space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 18 000–18 010.
- [7] W. Xie, J. Zheng, J. Han, J. Shi, W. Zhang, C. Bai, and X. Li, "Textop: Real-time interactive text-driven humanoid robot motion generation and control," *arXiv preprint arXiv:2602.07439*, 2026.
- [8] N. Jiang, Z. He, W. Yu, L. Pang, Y. Li, H. Li, J. Cui, Y. Li, Y. Wang, Y. Zhu, *et al.*, "Uniact: Unified motion generation and action streaming for humanoid robots," *arXiv preprint arXiv:2512.24321*, 2025.
- [9] Y. Shao, X. Huang, B. Zhang, Q. Liao, Y. Gao, Y. Chi, Z. Li, S. Shao, and K. Sreenath, "Langwbc: Language-directed humanoid whole-body control via end-to-end learning," *arXiv preprint arXiv:2504.21738*, 2025.
- [10] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.
- [11] J. P. Araujo, Y. Ze, P. Xu, J. Wu, and C. K. Liu, "Retargeting matters: General motion retargeting for humanoid motion tracking," *arXiv preprint arXiv:2510.02252*, 2025.
- [12] Z. Luo, J. Cao, A. W. Winkler, K. Kitani, and W. Xu, "Perpetual humanoid control for real-time simulated avatars," in *International Conference on Computer Vision (ICCV)*, 2023.
- [13] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.
- [14] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng, "Generating diverse and natural 3d human motions from text," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 5152–5161.
- [15] Y. Wang, H. Jiang, S. Yao, Z. Ding, and Z. Lu, "Sentinel: A fully end-to-end language-action model for humanoid whole body control," *arXiv preprint arXiv:2511.19236*, 2025.
- [16] G. Tevet, B. Gordon, A. Hertz, A. H. Bermano, and D. Cohen-Or, MotionCLIP: Exposing Human Motion Generation to CLIP Space.
- [17] J. Zhang, Y. Zhang, X. Cun, Y. Zhang, H. Zhao, H. Lu, X. Shen, and Y. Shan, "Generating human motion from textual descriptions with discrete representations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 14 730–14 740.
- [18] B. Jiang, X. Chen, W. Liu, J. Yu, G. Yu, and T. Chen, "Motiongpt: Human motion as a foreign language," *Advances in Neural Information Processing Systems*, vol. 36, pp. 20 067–20 079, 2023.
- [19] J. Zhang, Y. Zhang, X. Cun, S. Huang, Y. Zhang, H. Zhao, H. Lu, and X. Shen, T2M-GPT: Generating Human Motion from Textual Descriptions with Discrete Representations.
- [20] H. Liang, J. Bao, R. Zhang, S. Ren, Y. Xu, S. Yang, X. Chen, J. Yu, and L. Xu, "Omg: Towards open-vocabulary motion generation via mixture of controllers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 482–493.
- [21] W. Dai, L.-H. Chen, J. Wang, J. Liu, B. Dai, and Y. Tang, "Motionlcm: Real-time controllable motion generation via latent consistency model," in *European Conference on Computer Vision*, 2024, pp. 390–408.
- [22] M. Zhang, X. Guo, L. Pan, Z. Cai, F. Hong, H. Li, L. Yang, and Z. Liu, "Remodiffuse: Retrieval-augmented motion diffusion model," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 364–373.
- [23] Y. Shafir, G. Tevet, R. Kapon, and A. H. Bermano, "Human motion diffusion as a generative prior," *arXiv preprint arXiv:2303.01418*, 2023.
- [24] Y. Yuan, J. Song, U. Iqbal, A. Vahdat, and J. Kautz, "Physdiff: Physics-guided human motion diffusion model," 2023.
- [25] G. Tevet, S. Raab, S. Cohan, D. Reda, Z. Luo, X. B. Peng, A. H. Bermano, and M. van de Panne, "Closd: Closing the loop between simulation and diffusion for multi-task character control," *arXiv preprint arXiv:2410.03441*, 2024.
- [26] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu, "Avatarclip: Zero-shot text-driven generation and animation of 3d avatars," *arXiv preprint arXiv:2205.08535*, 2022.
- [27] P. Ding, J. Ma, X. Tong, B. Zou, X. Luo, Y. Fan, T. Wang, H. Lu, P. Mo, J. Liu, *et al.*, "Humanoid-vla: Towards universal humanoid control with visual integration," *arXiv preprint arXiv:2502.14795*, 2025.
- [28] Z. Liu, K. Ji, K. Yang, J. Yu, Y. Shi, and J. Wang, "Commanding humanoid by free-form language: A large language action model with unified motion vocabulary," *arXiv preprint arXiv:2511.22963*, 2025.
- [29] Z. Li, C. Chi, Y. Wei, B. Zhu, Y. Peng, T. Huang, P. Wang, Z. Wang, S. Zhang, and C. Xu, "From language to locomotion: Retargeting-free humanoid control via motion latent guidance," *arXiv preprint arXiv:2510.14952*, 2025.
- [30] Z. Li, C. Chi, Y. Wei, B. Zhu, T. Huang, Z. Sun, Y. Peng, P. Wang, Z. Wang, F. Liu, *et al.*, "Do you have freestyle? expressive humanoid locomotion via audio control," *arXiv preprint arXiv:2512.23650*, 2025.
- [31] Z. Luo, Y. Yuan, T. Wang, C. Li, S. Chen, F. Castaneda, Z.-A. Cao, J. Li, D. Minor, Q. Ben, *et al.*, "Sonic: Supersizing motion tracking for natural humanoid whole-body control," *arXiv preprint arXiv:2511.07820*, 2025.
- [32] H. Song, D. Qu, Y. Yao, Q. Chen, Q. Lv, Y. Tang, M. Shi, G. Ren, M. Yao, B. Zhao, *et al.*, "Hume: Introducing system-2 thinking in visual-language-action model," *arXiv preprint arXiv:2505.21432*, 2025.
- [33] H. Yang, J. Bao, Y. Xin, H. Song, Y. Tian, B. Zhao, D. Wang, and X. Li, "Zerowbc: Learning natural visuomotor humanoid control directly from human egocentric video," *arXiv e-prints*, pp. arXiv–2603, 2026.
- [34] Z. Sun, B.-S. Huang, Y. Peng, X. Li, J. Ma, Y. Sun, Z. Li, H. Jiang, B. Gao, Z. Bing, *et al.*, "Mosaic: Bridging the sim-to-real gap in generalist humanoid motion tracking and teleoperation with rapid residual adaptation," *arXiv preprint arXiv:2602.08594*, 2026.
- [35] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang, "Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit," 2025.
- [36] Y. Wang, M. Yang, Z. Ding, Y. Zhang, W. Zeng, X. Xu, H. Jiang, and Z. Lu, "From experts to a generalist: Toward general whole-body control for humanoid robots," *arXiv preprint arXiv:2506.12779*, 2025.
- [37] J. Li, B. Tang, F. Wu, and R. Cao, "Telegate: Whole-body humanoid teleoperation via gated expert selection with motion prior," *arXiv preprint arXiv:2602.09628*, 2026.
- [38] Y. Zhao, X. Wang, D. Wang, X. Liu, D. Lu, Q. Han, P. Liu, and C. Bai, "Towards adaptive humanoid control via multi-behavior distillation and reinforced fine-tuning," *arXiv preprint arXiv:2511.06371*, 2025.
- [39] Y. Li, Z. Luo, T. Zhang, C. Dai, A. Kanervisto, A. Tirinzoni, H. Weng, K. Kitani, M. Guzek, A. Touati, *et al.*, "Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning," *arXiv preprint arXiv:2511.04131*, 2025.
- [40] Z. Zhang, C. Chen, H. Xue, J. Wang, S. Liang, Y. Liu, Z. Zhang, H. Wang, and L. Yi, "Unleashing humanoid reaching potential via real-world-ready skill space," *IEEE Robotics and Automation Letters*, vol. 11, no. 2, pp. 2082–2089, 2025.
- [41] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *International Conference on Computer Vision*, Oct. 2019, pp. 5442–5451.
- [42] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 60–1, 2020.
- [43] M. V. Koroteev, "Bert: a review of applications in natural language processing and understanding," *arXiv preprint arXiv:2103.11943*, 2021.



### A. Motion Generation Module

The motion generation module is a diffusion-based generator that takes text prompts as input and outputs 262-dimensional robot-native motion sequences. We adopt a Transformer decoder architecture with AdaLN for timestep injection and TP-MoE for fine-grained text-motion alignment. The model is trained on retargeted HumanML3D data with ASFO to handle long-tailed action distributions.

#### 1. Network Architecture.

The denoising network is an 8-layer AdaLN Transformer decoder with latent dimension  $d = 512$ , feed-forward dimension 1024, 4 attention heads, and dropout rate 0.1. We use GELU activation throughout.

a) *Text Encoder and Summary Token*: We use frozen DistilBERT [43] as the text encoder, outputting variable-length 768-dim token embeddings. A learnable query vector performs multi-head attention pooling (4 heads) over these tokens to produce a single summary embedding, which is concatenated with the original tokens to form the cross-attention memory of shape  $[(1+N) \times d]$ . This provides both a global semantic signal and fine-grained per-token information.

b) *TP-MoE Configuration*: We use  $K = 12$  experts, each being a two-layer FFN with hidden dimension 1024. The gating network  $\mathcal{G}$  is a 3-layer MLP ( $768 \rightarrow d \rightarrow d \rightarrow K$ , with SiLU activations) that maps DistilBERT token embeddings to expert weights. The spatial mask parameters are  $\gamma = 24$  (sharpness) and  $\beta = 0.25$  (threshold ratio). The load-balancing loss weight is  $\lambda_{lb} = 0.01$ , computed as  $\mathcal{L}_{lb} = K \sum_{j=1}^K (\bar{p}_j - 1/K)^2$ , where  $\bar{p}_j$  is the mean routing probability for expert  $j$ .

#### 2. Feature Representation Details.

a) *Body Selection*: For the 262-dimensional representation, we select 12 informative bodies for  $p_t^{\text{ric}}$ : left/right elbow, wrist\_roll, knee, ankle\_pitch, ankle\_roll, and palm (body indices [7, 8, 12, 13, 17, 18, 19, 23, 24, 25, 28, 29] in the 30-body skeleton). Similarly,  $\hat{p}_t^{\text{local}}$  includes these 12 bodies plus the root (13 bodies total). This selection excludes nearly-static torso bodies (chest, abdomen, shoulders, hips) while retaining all articulations and end-effectors.

b) *Block-wise Normalization*: Z-score normalization is applied block-wise:  $\dot{\omega}_t^{\text{root}}$  (dims 0–2),  $\dot{v}_t^{\text{root}}$  (dims 3–5),  $z_t$  (dim 6),  $p_t^{\text{ric}}$  (dims 7–42), and  $\hat{p}_t^{\text{local}}$  (dims 217–255) are normalized;  $R_t^{\text{6d}}$  (dims 43–216) is *not* normalized since its elements naturally lie in  $[-1, 1]$ ; binary contact indicators  $c_t^{\text{foot}}$  (dims 256–259) and  $c_t^{\text{hand}}$  (dims 260–261) are preserved without normalization.

c) *Contact Detection*:  $c_t^{\text{foot}} \in \{0, 1\}^4$  (4 values for left/right ankle pitch/roll) is set to 1 when ankle height  $< 0.05$  m and horizontal velocity  $< 0.01$  m/s.  $c_t^{\text{hand}} \in \{0, 1\}^2$  uses a height threshold of 0.10 m.

d) *Canonical Heading*: All sequences are rotated so the first frame faces  $+X$  (yaw = 0). Only the yaw component is removed; pitch and roll are preserved, which is critical

for non-upright motions (e.g., rolling, crawling). The root xy position is also shifted to the origin at the first frame.

### 3. Classifier-Free Guidance with Negative Prompts.

We support both standard CFG and negative-prompt CFG. In standard mode:

$$\hat{m}_0 = \hat{m}_0^{\emptyset} + s \cdot (\hat{m}_0^l - \hat{m}_0^{\emptyset}), \quad (13)$$

where  $\hat{m}_0^{\emptyset}$  is the unconditional prediction (text embedding zeroed out via the 10% training dropout) and  $s = 2.5$ . When a negative prompt  $l^-$  is provided, its encoding replaces the unconditional output:

$$\hat{m}_0 = \hat{m}_0^{l^-} + s \cdot (\hat{m}_0^l - \hat{m}_0^{l^-}), \quad (14)$$

which steers generation *away* from the negative description (e.g., “no walking, no standing still”) while following the positive prompt.

### B. General Motion Tracking

This section provides additional implementation details of the General Motion Tracking (GMT) module, including the complete reward specification, observation and action spaces, future command encoder, adaptive sampling strategy, curriculum hyperparameters, domain randomization, early termination conditions, PPO settings, network architecture, simulation setup, and the neuro-control difficulty taxonomy.

#### 1. Reward Functions.

The reward function used for GMT training consists of task rewards for accurate motion imitation and regularization rewards for smooth and safe control. All task reward terms adopt an exponential kernel of the form

$$r = \exp\left(-\frac{e}{\sigma^2}\right), \quad (15)$$

where  $e$  denotes the squared error term and  $\sigma$  controls the reward sensitivity.

a) *Task rewards*: The task reward terms are summarized in Table III. Here,  $\mathbf{p}$  and  $\hat{\mathbf{p}}$  denote the reference and simulated positions, respectively;  $\mathbf{q} \ominus \hat{\mathbf{q}}$  denotes the quaternion error magnitude;  $\mathbf{v}$  and  $\boldsymbol{\omega}$  denote linear and angular velocities; and  $N$  is the number of tracked rigid bodies.

TABLE III: Task reward terms used in GMT training.

| Term                      | Expression                                                                                                        | Weight | $\sigma$ |
|---------------------------|-------------------------------------------------------------------------------------------------------------------|--------|----------|
| Global Anchor Position    | $\exp\left(-\frac{\ \mathbf{p}^{\text{anchor}} - \hat{\mathbf{p}}^{\text{anchor}}\ ^2}{\sigma^2}\right)$          | 0.8    | 0.2      |
| Global Anchor Orientation | $\exp\left(-\frac{\ \mathbf{q}^{\text{anchor}} \ominus \hat{\mathbf{q}}^{\text{anchor}}\ ^2}{\sigma^2}\right)$    | 0.5    | 0.4      |
| Relative Body Position    | $\exp\left(-\frac{\frac{1}{N} \sum_i \ \mathbf{p}_i - \hat{\mathbf{p}}_i\ ^2}{\sigma^2}\right)$                   | 1.0    | 0.3      |
| Relative Body Orientation | $\exp\left(-\frac{\frac{1}{N} \sum_i \ \mathbf{q}_i \ominus \hat{\mathbf{q}}_i\ ^2}{\sigma^2}\right)$             | 1.0    | 0.4      |
| Body Linear Velocity      | $\exp\left(-\frac{\frac{1}{N} \sum_i \ \mathbf{v}_i - \hat{\mathbf{v}}_i\ ^2}{\sigma^2}\right)$                   | 1.0    | 1.0      |
| Body Angular Velocity     | $\exp\left(-\frac{\frac{1}{N} \sum_i \ \boldsymbol{\omega}_i - \hat{\boldsymbol{\omega}}_i\ ^2}{\sigma^2}\right)$ | 1.0    | 3.14     |

The *Global Anchor Position* and *Global Anchor Orientation* terms measure pelvis tracking error in world coordinates. The *Relative Body Position* and *Relative Body*

*Orientation* terms are computed relative to the anchor frame and therefore capture body-pose consistency independently of global placement. The *Body Linear Velocity* and *Body Angular Velocity* terms encourage temporal consistency with the reference motion.

*b) Regularization rewards:* The regularization terms are listed in Table IV.

TABLE IV: Regularization rewards used in GMT training.

| Term                | Expression                            | Weight |
|---------------------|---------------------------------------|--------|
| Action Rate L2      | $\ a_t - a_{t-1}\ _2^2$               | -0.1   |
| Joint Limit Penalty | $\sum_j \mathbf{1}_{\text{out}}(q_j)$ | -10.0  |
| Undesired Contacts  | $\sum_b \mathbf{1}_{F_b > 1.0}$       | -0.1   |

The *Action Rate L2* term penalizes abrupt changes in joint-position targets to promote smooth control signals. The *Joint Limit Penalty* applies a large penalty whenever any joint exceeds its feasible position range, thereby improving mechanical safety. The *Undesired Contacts* term penalizes contact forces larger than 1.0 N on non-end-effector bodies. The following bodies are excluded from this penalty: `left_ankle_rolllink`, `right_ankle_rolllink`, `left_wrist_yawlink`, and `right_wrist_yawlink`.

## 2. Observation and Action Spaces

*a) Policy observation space:* The policy observation has 616 dimensions and is summarized in Table V. Additive uniform noise is applied during training for domain randomization:  $\pm 0.05$  on root orientation,  $\pm 0.2$  on angular velocity,  $\pm 0.01$  on joint positions, and  $\pm 0.5$  on joint velocities.

TABLE V: Policy observation space for our GMT.

| Component           | Description                                        | Dimension |
|---------------------|----------------------------------------------------|-----------|
| command             | Motion target (current + future frames)            | 520       |
| motion_anchor_ori_b | Root orientation error in body frame (6D rotation) | 6         |
| base_ang_vel        | Base angular velocity                              | 3         |
| joint_pos           | Joint positions (relative to default)              | 29        |
| joint_vel           | Joint velocities                                   | 29        |
| actions             | Previous actions                                   | 29        |

*b) Critic observation space:* The critic receives privileged, noise-free observations, as summarized in Table VI.

TABLE VI: Critic observation space for our GMT.

| Component           | Description                                        | Dimension |
|---------------------|----------------------------------------------------|-----------|
| command             | Motion target (current + future frames)            | 520       |
| motion_anchor_pos_b | Root position error in body frame                  | 3         |
| motion_anchor_ori_b | Root orientation error in body frame (6D rotation) | 6         |
| body_pos            | Key body positions (14 bodies $\times 3$ )         | 42        |
| body_ori            | Key body orientations (14 bodies $\times 6D$ )     | 84        |
| base_lin_vel        | Base linear velocity                               | 3         |
| base_ang_vel        | Base angular velocity                              | 3         |
| joint_pos           | Joint positions                                    | 29        |
| joint_vel           | Joint velocities                                   | 29        |
| actions             | Previous actions                                   | 29        |

*c) Motion command structure:* The motion command has 520 dimensions. Each frame contains 65 dimensions, including joint positions (29), joint velocities (29), root position (3), and root quaternion (4). The command structure is summarized in Table VII.

TABLE VII: Structure of the motion command used in GMT.

| Segment       | Frames | Per-Frame Dim | Total Dim |
|---------------|--------|---------------|-----------|
| Current       | 1      | 65            | 65        |
| Short-horizon | 2      | 65            | 130       |
| Long-horizon  | 5      | 65            | 325       |

The short-horizon frames are directly concatenated into the observation. The long-horizon frames are sampled with a stride  $k = 20$ , corresponding to  $5 \times 20 = 100$  simulation steps, i.e., approximately 2 seconds at 50 Hz, and are processed by the future command encoder.

*d) Action space:* The action is a 29-dimensional vector of joint position targets. A PD controller converts these targets into joint torques at each simulation step. The control frequency is 50 Hz, corresponding to a simulation time step of 0.005 s and a control decimation factor of 4.

## 3. Error-Driven Adaptive Sampling

In addition to curriculum-level scheduling, GMT employs file-level adaptive sampling within each difficulty level. For each motion file  $i$ , the following statistics are maintained using exponential moving averages (EMA):

$$E_i \leftarrow (1 - \alpha)E_i + \alpha \tilde{e}_i, \quad (16)$$

$$\hat{p}_i^{\text{succ}} = \frac{S_i}{S_i + F_i + \epsilon}, \quad (17)$$

where  $\tilde{e}_i$  is the mean tracking error for file  $i$  in the current batch,  $\alpha = 0.25$ , and the success-rate EMA uses decay  $\beta = 0.4$ .

The sampling score is defined as

$$r_i = (1 - w) \cdot \min(E_i/c, 1) + w \cdot (1 - \hat{p}_i^{\text{succ}}), \quad (18)$$

where  $c = 0.3$  is the error normalization constant and  $w = 0.15$  is the success-rate mixing weight, activated after a 6000-

iteration warmup. The final sampling distribution is given by

$$p_i = (1 - \varepsilon) \cdot \text{softmax} \left( \frac{\log(r_i + \epsilon)}{T} \right) + \frac{\varepsilon}{N}, \quad (19)$$

where  $T = 1.05$  and  $\varepsilon = 0.20$  ensures minimum coverage.

#### 4. Curriculum Learning Hyperparameters

*a) Metric convergence detection:* Early promotion is enabled when the relative improvement in MPJPE/MPJAE falls below 3% for 3 consecutive evaluations, subject to a minimum of 3000 iterations on the current level.

*b) Gradual file introduction:* When a new level is unlocked, files are introduced progressively according to

$$r(t) = r_{\text{start}} + \frac{\min(t - t_{\text{unlock}}, T_{\text{intro}})}{T_{\text{intro}}} \cdot (1 - r_{\text{start}}), \quad (20)$$

where  $r_{\text{start}} = 0.2$ . The associated hyperparameters are listed in Table VIII.

TABLE VIII: Gradual file introduction hyperparameters.

| Parameter                                       | Value |
|-------------------------------------------------|-------|
| Start ratio $r_{\text{start}}$                  | 0.2   |
| Base introduction iterations $T_{\text{intro}}$ | 3000  |
| Extra introduction iterations (level $\geq 4$ ) | +2000 |

*c) Freeze-and-drop parameters:* The freeze-and-drop mechanism is configured as shown in Table IX.

TABLE IX: Freeze-and-drop hyperparameters.

| Parameter                              | Value          |
|----------------------------------------|----------------|
| Error threshold $\tau_{\text{err}}$    | 0.1            |
| Success threshold $\tau_{\text{succ}}$ | 0.15           |
| Minimum attempts $n_{\text{min}}$      | 20000          |
| Freeze duration (iterations)           | 4000           |
| Freezes before permanent drop          | 2              |
| Check interval                         | 500 iterations |

#### 5. Domain Randomization

The domain randomization settings are summarized in Table X.

TABLE X: Domain randomization settings used in GMT training.

| Randomization                  | Range / Parameters                        | Mode     |
|--------------------------------|-------------------------------------------|----------|
| Ground friction (static)       | [0.3, 1.6]                                | Startup  |
| Ground friction (dynamic)      | [0.3, 1.2]                                | Startup  |
| Restitution                    | [0.0, 0.5]                                | Startup  |
| Joint default position offset  | [-0.01, 0.01] rad                         | Startup  |
| Torso CoM offset (x)           | [-0.025, 0.025] m                         | Startup  |
| Torso CoM offset (y)           | [-0.05, 0.05] m                           | Startup  |
| Torso CoM offset (z)           | [-0.05, 0.05] m                           | Startup  |
| External push (interval 1–3 s) | lin: $\pm 0.5$ m/s, ang: $\pm 0.78$ rad/s | Interval |

Additionally, observation noise is injected during training, as described in the policy observation space above.

#### 6. PPO Training Hyperparameters

The PPO hyperparameters used for GMT training are listed in Table XI.

TABLE XI: PPO training hyperparameters for GMT.

| Parameter                           | Value               |
|-------------------------------------|---------------------|
| Parallel environments               | 8192                |
| Steps per env per iteration         | 24                  |
| Learning epochs per iteration       | 5                   |
| Mini-batches per epoch              | 4                   |
| Initial learning rate               | $1 \times 10^{-3}$  |
| LR schedule                         | Adaptive (KL-based) |
| Desired KL divergence               | 0.01                |
| Discount factor $\gamma$            | 0.99                |
| GAE $\lambda$                       | 0.95                |
| Clip parameter                      | 0.2                 |
| Entropy coefficient                 | 0.005               |
| Value loss coefficient              | 1.0                 |
| Max gradient norm                   | 1.0                 |
| Initial action noise std            | 1.0                 |
| Advantage normalization             | Per mini-batch: off |
| Empirical observation normalization | On                  |
| Maximum iterations                  | 200,000             |
| Checkpoint save interval            | 1000 iterations     |

#### 7. Network Architecture

*a) Actor (MoE policy):* Each expert MLP uses hidden dimensions [2048, 1024, 512] with ELU activations. The gate network maps the 128-dimensional latent  $\mathbf{z}_t$  to  $K$  routing logits. During Stage I, the number of active experts grows from 1 to 10 as curriculum levels are unlocked. Stage II may dynamically add experts for persistently difficult motions.

*b) Critic:* The critic is a single MLP with hidden dimensions [2048, 1024, 512] and ELU activations, sharing the same future command encoder as the actor.

#### 8. Simulation Configuration

The simulation settings are summarized in Table XII.

TABLE XII: Simulation configuration for GMT training.

| Parameter           | Value                |
|---------------------|----------------------|
| Simulator           | Isaac Lab (IsaacSim) |
| Physics dt          | 0.005 s              |
| Control decimation  | 4 (control at 50 Hz) |
| Episode length      | 10 s                 |
| Environment spacing | 2.5 m                |

#### 9. Neuro-Control Difficulty Taxonomy

As discussed in the main text, motion difficulty is interpreted from a neuro-control perspective as the coordination load imposed on the central nervous system (CNS). We decompose this aggregate load into four interpretable sub-dimensions: motor planning load, predictive control load, balance and multisensory integration load, and contextual/environmental constraint load.

*a) Motor Planning Load:* This dimension reflects the complexity of an action’s temporal structure and sequential organization. Executing multi-step movement sequences relies on prospective planning and ordinal encoding in the primary motor cortex, premotor cortex, and supplementary motor area (SMA). Actions that require chaining multiple distinct motor primitives in a specific order impose higher planning demands than repetitive or single-phase motions.

TABLE XIII: Neuro-control difficulty taxonomy used in GMT curriculum design.

| Level         | Primary Sub-dimensions          | Representative Actions                  | Characteristics                                                                        |
|---------------|---------------------------------|-----------------------------------------|----------------------------------------------------------------------------------------|
| 1             | Low planning                    | Stand, raise arms, wave hands           | Near-static or upper-body only; minimal balance demand                                 |
| 2–4           | Planning                        | Walk, step, turn                        | Basic locomotion with stable rhythm and simple motor patterns                          |
| 5–7           | Planning + Predictive + Balance | Run, jump, walk backwards               | Combined locomotion and dynamic actions requiring higher coordination                  |
| 8             | Planning + Balance              | Crawl, kneel, crawl on knees            | Low-posture ground movements with increased body control                               |
| 9–10          | All four (maximal)              | Cartwheel, backflip, spinning, dance    | Acrobatic or high-skill actions with tightly coupled planning, prediction, and balance |
| 11 (excluded) | Contextual-dominant             | Stair climbing, stepping over obstacles | Requires external terrain or elevation changes                                         |
| 12 (excluded) | Contextual-dominant             | Swimming, flying                        | Physically unrealizable in flat-ground simulation                                      |

*b) Predictive Control Load:* This dimension captures the degree to which an action depends on feedforward prediction and error correction under sensory feedback delays or tight temporal constraints. Actions involving ballistic phases, rapid transitions, or precise timing place a greater demand on predictive control mechanisms.

*c) Balance and Multisensory Integration Load:* This dimension reflects the demand for integrating vestibular, proprioceptive, and visual information to maintain postural stability and spatial orientation. The load increases substantially during dynamic balance, turning, backward locomotion, and aerial phases.

*d) Contextual and Environmental Constraint Load:* This dimension reflects additional perceptual and control demands introduced by terrain structure, elevation changes, or environmental hazards. Such motions require continuous adjustment of gait parameters based on external context and are therefore not fully representable in a flat-ground simulation setting.

#### 10. Description aggregation and primary-action identification

Each motion instance may be associated with multiple textual annotations that describe the same underlying action from different perspectives or with different levels of detail. During difficulty assessment, these annotations are jointly interpreted to infer the action being performed and to estimate its overall motion complexity. In practice, we use the DeepSeek model to classify the action described by these annotations and to identify the dominant motion primitive underlying the sequence. If a description includes multiple consecutive movement components, we first identify the *primary action*, namely, the most essential or most difficult motion primitive in the sequence, and use it to determine the base difficulty level. Subsequent score adjustment is then based on the number and continuity of the remaining action components. Actions that explicitly depend on external structures (e.g., stairs, platforms, or narrow elevated surfaces) are assigned difficulty level 11, while actions that are physically infeasible under normal gravity conditions, such as swimming or flying, are assigned difficulty level 12

and treated as special cases.

The corresponding mapping to curriculum levels is summarized in Table XIII.