HETSKILLS: REUSABLE HETEROGENEOUS SKILL LEARNING FOR PHYSICS-BASED CONTROL

Anonymous authors
Paper under double-blind review



Figure 1: Overview of HetSkills. From left to right, the results illustrate four representative skill categories: motion tracking, text-to-motion generation, motion completion, and downstream task adaptation. These skills are learned from heterogeneous data sources, supervision signals, and task objectives, and are decoded by a shared physics-based motion controller, allowing learned skills to serve as reusable resources for subsequent skill acquisition and composition.

ABSTRACT

We propose HetSkills, a versatile and extensible framework for physics-based character control that supports modular acquisition and composition of heterogeneous skills without retraining learned skills or the low-level controller. The key idea is to preserve learned skills as executable capabilities and further reuse them as control foundations, expert teachers, or behavioral priors for acquiring new skills. HetSkills begins by learning a tracking skill that establishes a strong foundation in motion control and creates a shared motion decoder, which can be reused across tasks without the need for retraining or separate controllers. To prevent the text-to-motion skill from exploiting shortcut pathways instead of learning language semantics, we introduce motion intuition distillation to ground text-to-motion generation in language semantics and a task-guidance module that dynamically adjusts actions based on high-level language instructions. This enables HetSkills to preserve natural motion while continuously expanding its skill repertoire, making it highly adaptable for long-horizon tasks. Experimental results demonstrate the effectiveness in motion tracking, text-to-motion generation, motion completion, and downstream task adaptation, achieving impressive success rates even under challenging conditions.

1 INTRODUCTION

Learning physics-based character skills that are natural, composable, and reusable is a fundamental goal in animation, gaming, and robotics (Peng et al., 2018). A versatile character, however, must acquire a broad repertoire of skills rather than a single behavior, often from heterogeneous data sources, conditioning modalities, supervision signals, and task objectives. Recent advances in large-scale motion priors and downstream adaptation (Luo et al., 2023a; 2026; Shi et al., 2026) have established strong reusable motor foundations. Building on these foundations, a more scalable approach to skill learning is to retain acquired skills and draw on relevant prior experience when learning new ones, much as humans do. From this perspective, each learned skill becomes part of the character's accumulated experience rather than an isolated outcome, remaining available for later adaptation and flexible composition.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing general motion controllers and downstream adaptation methods have enabled a pretrained motor foundation to support diverse control interfaces and task objectives, without requiring a separate low-level controller for each new task (Tessler et al., 2024; Wu et al., 2025a). However, prior work has mainly focused on how different tasks can reuse the same motor foundation. Less attention has been paid to how heterogeneous skills can be acquired from different data sources and supervision signals on top of this foundation, or how a previously acquired skill can in turn facilitate the learning of a subsequent one. Consequently, even systems that support multiple tasks often provide limited support for reusing prior experience during new skill acquisition and for flexibly composing heterogeneous skills into coherent long-horizon behaviors.

To enable reusable learning and composition of heterogeneous skills in physics-based character control, we propose HetSkills, a physics-based character control framework that preserves acquired skills and leverages them for subsequent skill learning. Specifically, heterogeneous skills refer to capabilities acquired from different data sources, supervision forms, and task objectives, including motion tracking, text-to-motion generation, motion completion, and downstream task adaptation. Instead of training isolated controllers for different capabilities, HetSkills allows each newly learned skill to build upon previously acquired knowledge while remaining an independently executable capability. Through a shared execution framework, these heterogeneous skills can be reused and organized for flexible adaptation and long-horizon composition.

More specifically, HetSkills follows the idea of a shared executable control interface used in previous motion-prior methods (Peng et al., 2022; Tessler et al., 2023), but extends it toward reusable skill organization. After learning a skill, HetSkills preserves it as an executable module and represents it through a standardized task description, including its initialization, condition, and termination criteria. This allows skills acquired from different data sources and supervision forms to remain independently executable while being composed into long-horizon behaviors. To build this framework, we first learn a tracking skill with a part-wise character decomposition (Bae et al., 2023; 2025), which provides a reusable motor foundation for subsequent skills. We then learn text-to-motion and motion completion skills on top of this foundation, introducing Motion Intuition Distillation (MID) to improve language grounding. Finally, downstream tasks are solved through lightweight adaptation by combining reusable language-conditioned priors with task-specific guidance, without requiring task demonstrations or retraining the low-level controller.

Experiments demonstrate the effectiveness of HetSkills from three perspectives. First, HetSkills enables heterogeneous skills learned from different data sources and supervision forms to be independently acquired, reused, and organized through a standardized skill interface, allowing diverse capabilities to be composed into long-horizon behaviors. Second, HetSkills achieves robust text-to-motion generation under both standard and challenging initializations, reaching a success rate of 96.9% under standard initialization and 81.7% from a neutral pose. Third, HetSkills transfers effectively to novel downstream tasks through lightweight adaptation, without requiring task-specific demonstrations or retraining the low-level controller. These results demonstrate the potential of reusable heterogeneous skill learning for building scalable physics-based character control systems.

2 Related Work

Physics-based Motion Imitation and Skill Learning. Physics-based character control commonly combines motion capture data with reinforcement learning to acquire physically plausible behaviors. Reference-tracking approaches learn policies that reproduce motion clips, with scalable expert training and distillation extending this paradigm to diverse skill collections (Peng et al., 2018; Won et al., 2020). To reduce reliance on explicit motion matching, adversarial approaches learn motion rewards from demonstrations (Peng et al., 2021; Zhang et al., 2025), while latent skill embeddings further support behavioral diversity and high-level control (Peng et al., 2022; Tessler et al., 2023; Dou et al., 2023). In parallel, VAE-based methods (Kingma & Welling, 2013) model motion through continuous or discrete latent variables, learning structured skill representations that support diverse motion generation and downstream task reuse (Merel et al., 2018; Ling et al., 2020; Yao et al., 2022; Zhu et al., 2023). Diffusion-based methods broaden this toolkit through policy learning, look-ahead control, and motion generation (Truong et al., 2024; Huang et al., 2025), as well as reusable motion priors for task learning (Mu et al., 2025). These approaches establish motion imitation and generative modeling as complementary foundations for learning natural and reusable character behaviors.

Large-scale Motion Priors and Language-conditioned Control. Large-scale motion tracking and latent representation learning improve coverage and generalization by providing reusable motor foundations for downstream applications (Luo et al., 2023b;a; Yao et al., 2024). The acquired skills can be distilled or encoded into compact control spaces, enabling reuse without training separate low-level controllers. This paradigm has been further extended by recent work. SONIC (Luo et al., 2026) scales motion tracking and supports multiple motion interfaces through a shared latent representation, whereas GPC (Shi et al., 2026) combines skill tokenization with autoregressive generative pretraining and downstream adaptation. In parallel, language-conditioned approaches connect high-level instructions to physical behaviors through language-directed policies or instruction-conditioned motion planning (Juravsky et al., 2022; 2024; Ren et al., 2023). Richer formulations integrate language, sparse motion constraints, and spatial goals into unified control models (Tessler et al., 2024; Wu et al., 2025a). These advances demonstrate the value of shared motor representations and flexible control interfaces. Building on these foundations, HetSkills focuses on learning a modular repertoire of heterogeneous skills that remain independently executable and can in turn support subsequent skill acquisition.

Heterogeneous Skill Learning and Compositional Control. Skill-based control addresses complex tasks by reusing knowledge across behaviors and coordinating their execution. Multi-skill training promotes knowledge sharing across interaction tasks, while task-specific encoders adapt pretrained behaviors to new objectives (Pan et al., 2025; Vainshtein et al., 2025). Hierarchical and compositional methods further exploit acquired capabilities by learning to combine pretrained motor primitives for more complex control objectives (Peng et al., 2019; Kuang et al., 2025). For multi-stage tasks, such composition also requires coordinating skill selection and transitions. Learned scheduling selects control fragments during execution (Liu & Hodgins, 2017), while policy integration uses pretrained policies to guide the learning of intermediate skills that connect constituent behaviors (Xu et al., 2025). At the task level, language-conditioned planning and large language model (LLM)-based decomposition organize behaviors through motion plans, goal sequences, or executable scripts, connecting high-level instructions to long-horizon physical interactions (Tevet et al., 2024; Wang et al., 2025; Wu et al., 2025b; Fu et al., 2024). HetSkills complements these approaches by learning skills from heterogeneous data and supervision while organizing their execution through a standardized task description. This interface specifies initialization, conditioning, and termination independently of skill-specific training, allowing acquired capabilities to be composed into long-horizon behaviors without retraining the skill modules or shared controller.

3 PRELIMINARIES

We formulate physics-based character control as a goal-conditioned Markov decision process (Liu et al., 2022). At time t, s^t denotes the character state, g^t the skill-specific condition, and a^t the action, with physical dynamics $s^{t+1} \sim p(\cdot \mid s^t, a^t)$. Actions specify target joint rotations, which are converted into joint torques by proportional-derivative (PD) controllers (Tan et al., 2011). For skills trained with reinforcement learning, the objective is to maximize the expected discounted return $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{T} \gamma^t r^t]$, where π is the policy, r^t is the task reward, and $\gamma \in [0, 1)$ is the discount factor (Sutton & Barto, 1998).

In HetSkills, each learned skill is retained as an independently executable module and invoked through a standardized task description $\mathcal{T}_i = (\mathrm{Initialization}_i, \mathrm{Condition}_i, \mathrm{Terminate}_i)$. *Initialization* specifies the setup required before execution, including any internal memory initialization; *Condition* supplies the skill-specific input g^t, such as reference motion, language, sparse motion constraints, or task goals; and *Terminate* specifies when execution ends. For example, a path-following skill receives a target path and terminates upon reaching its destination, whereas a text-to-motion skill receives a language instruction and can run for a specified duration. These descriptions provide a common execution interface for skills acquired from different data sources and supervision signals. A high-level controller can organize learned skills as $\mathrm{Seq}(\mathcal{T}_1, \ldots, \mathcal{T}_N)$, invoking the next skill when the current one terminates and carrying forward the simulated state. At a transition, the next skill uses this continuing physical state together with its own conditioning input and memory setup. This allows learned skills to be called independently or composed into long-horizon behaviors without retraining the constituent modules.

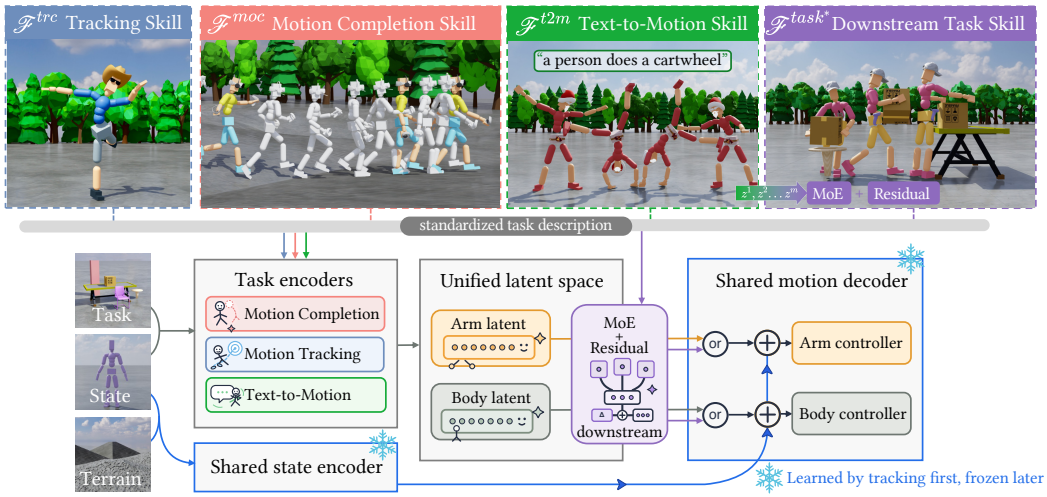


Figure 2: Overview of reusable heterogeneous skill learning in HetSkills. Motion tracking establishes a shared physics-based controller that is frozen and reused by later skills. Each acquired skill retains its conditioning module and remains independently executable through a standardized task description. The text-to-motion skill provides behavioral priors for downstream adaptation, where task-specific routing and residual modules combine and refine part-wise latent commands.

4 METHOD

As shown in Fig. 2, HetSkills learns a tracking skill $\mathscr{F}^{trc}$ and freezes its state encoder and part-wise decoders as a shared control foundation for subsequent skills. Text-to-motion $\mathscr{F}^{t2m}$ reuses the tracker as an expert teacher, with Motion Intuition Distillation (MID) promoting language grounding. Motion completion $\mathscr{F}^{moc}$ uses tracker supervision for VR tracking and motion in-betweening, and reinforcement learning for human-scene interaction. The learned text-to-motion skill then supplies frozen language-conditioned priors for downstream adaptation $\mathscr{F}^{task^*}$, where task rewards train lightweight gating and residual modules without task-specific demonstrations. Acquired skills remain independently executable and can be composed through the standardized task description in Section 3.

4.1 SHARED PHYSICAL CONTROLLER

We learn the control foundation by tracking reference motions with reinforcement learning (Peng et al., 2018). Following part-wise control (Bae et al., 2023; 2025), we partition the character into the arms and hands ($p = a$) and the remaining body joints ($p = b$). Both branches condition on the full-body state feature $c^t = \mathcal{E}_s(s^t, a^{t-1})$, providing a common physical context for coordination. Given a condition g^t containing two future reference frames, the tracking encoders predict deterministic latent commands, which the corresponding decoders convert into joint actions:

$$\begin{aligned} z_p^t &= \mathscr{F}_p^{trc}(s^t, g^t, a^{t-1}), \qquad p \in \{a, b\}, \\ a^t &= \left[\mathcal{D}_a(c^t, z_a^t),\ \mathcal{D}_b(c^t, z_b^t)\right]. \end{aligned} \tag{1}$$

We jointly train the tracking encoders, state encoder, and decoders on AMASS (Mahmood et al., 2019) using PPO (Schulman et al., 2017) with motion-tracking rewards. During tracking training, exploration noise is applied to the final joint actions. Latent magnitude and AR(1)-based temporal regularization (Merel et al., 2018) encourage compact, smooth commands. After training, $\mathcal{E}_s$, $\mathcal{D}_a$, and $\mathcal{D}_b$ are frozen, and subsequent skills learn new latent mappings into this controller. The tracking skill is retained for direct execution and expert supervision. Reward and regularization details are provided in Appendix C.

4.2 Learning Skills from Heterogeneous Supervision

Text-to-motion. We reuse the tracking skill as an expert teacher to learn text-conditioned physical motion from paired text-motion data. Transformer encoders (Vaswani et al., 2017) predict $z_p^t = \mathscr{F}_p^{t2m}(s^t, a^{t-1}, H^t, e)$, where H^t contains recent proprioceptive history and e comes from a frozen TMR text encoder (Petrovich et al., 2023). At each state visited by the student, the frozen tracker uses future frames of the paired reference motion to provide expert joint actions (Ross et al., 2011). We train the student with behavior cloning on the decoded actions from Eq. (1), propagating gradients through the frozen controller to update only the student encoders.

We introduce Motion Intuition Distillation (MID) to strengthen language grounding across varied initial states and motion histories. Future reference frames are available only to the teacher; the student predicts from text, the current state, previous action, and past history. When the initial state and history match the target clip, they can already provide strong cues for how its motion continues. To train language conditioning beyond these matched contexts, we propose Randomized Memory Initialization (RMI). We sample another motion clip, use its terminal frame as the character's initial state, and fill H^t with the corresponding preceding history, while retaining the target text-motion pair. The teacher uses the target reference motion to supervise actions at the states visited from this new initialization. The student is thus trained to produce the requested motion from a preceding context that need not match the text, encouraging it to use language. We also adopt Reference State Initialization (RSI) (Peng et al., 2018) to expose the student to different motion phases by sampling an initial state from a random frame of the target clip.

Motion completion. Motion completion recovers physically plausible full-body motion from spatially or temporally sparse targets. (1) *VR tracking* reconstructs full-body motion from the movements of the head and both hands. (2) *Motion in-betweening* fills temporal gaps between full-body keyframes (Harvey et al., 2020) using the next target pose, its time offset, and recent history. (3) *Human-scene interaction* applies temporal completion to object-interaction motions (Hwang et al., 2025), conditioned on future interaction poses, time offsets, and recent history. We train these skills independently using different forms of supervision: VR tracking and motion in-betweening use behavior cloning of the tracking teacher's decoded joint actions, while human-scene interaction uses PPO with tracking-style rewards on SAMP (Hassan et al., 2021), which contains interaction motions outside the tracker's training distribution.

4.3 Task Adaptation with Language Priors

We adapt the text-to-motion skill to downstream tasks by learning to combine its behavior priors and refine them for task objectives. For each body part, we specify language instructions represented by a fixed embedding set $\mathcal{I}_p = \{e_p^m\}_{m=1}^{M_p}$. These instructions describe useful behaviors, such as lifting and carrying for object manipulation, and guide exploration toward relevant motions. The instruction counts M_p can differ between parts. At each step, the text-to-motion skill produces state-dependent candidates $z_p^{t,m} = \mathscr{F}_p^{t2m}(s^t, a^{t-1}, H^t, e_p^m)$.

A gating network $\mathcal{G}_p$ adjusts each candidate's contribution as the task context changes. A residual network $\mathcal{R}_p$ then refines the mixture to address task requirements not fully specified by language. Given task-specific observations o_{task}^t, the routing weights and adapted latents are

$$\begin{aligned} \boldsymbol{\alpha}_p^t &= \mathrm{softmax}\big(\mathcal{G}_p(s^t, a^{t-1}, o_{\text{task}}^t)\big), \qquad p \in \{a, b\}, \\ \hat{z}_p^t &= \sum_{m=1}^{M_p} \alpha_p^{t,m} z_p^{t,m} + \mathcal{R}_p(s^t, a^{t-1}, o_{\text{task}}^t). \end{aligned} \tag{2}$$

The adapted commands are executed through Eq. (1). Task rewards train the gating and residual modules using PPO without task-specific motion demonstrations, while the text-conditioned priors and shared controller remain frozen. During PPO training, exploration noise is added to the adapted latents before decoding. A magnitude penalty discourages residual values outside a prescribed bound, while a temporal smoothness penalty discourages abrupt changes in the adapted latents; details are provided in Appendix C. After training, each instruction set and its guidance modules are retained as a new skill, independently executable or sequentially composable through the interface in Section 3.

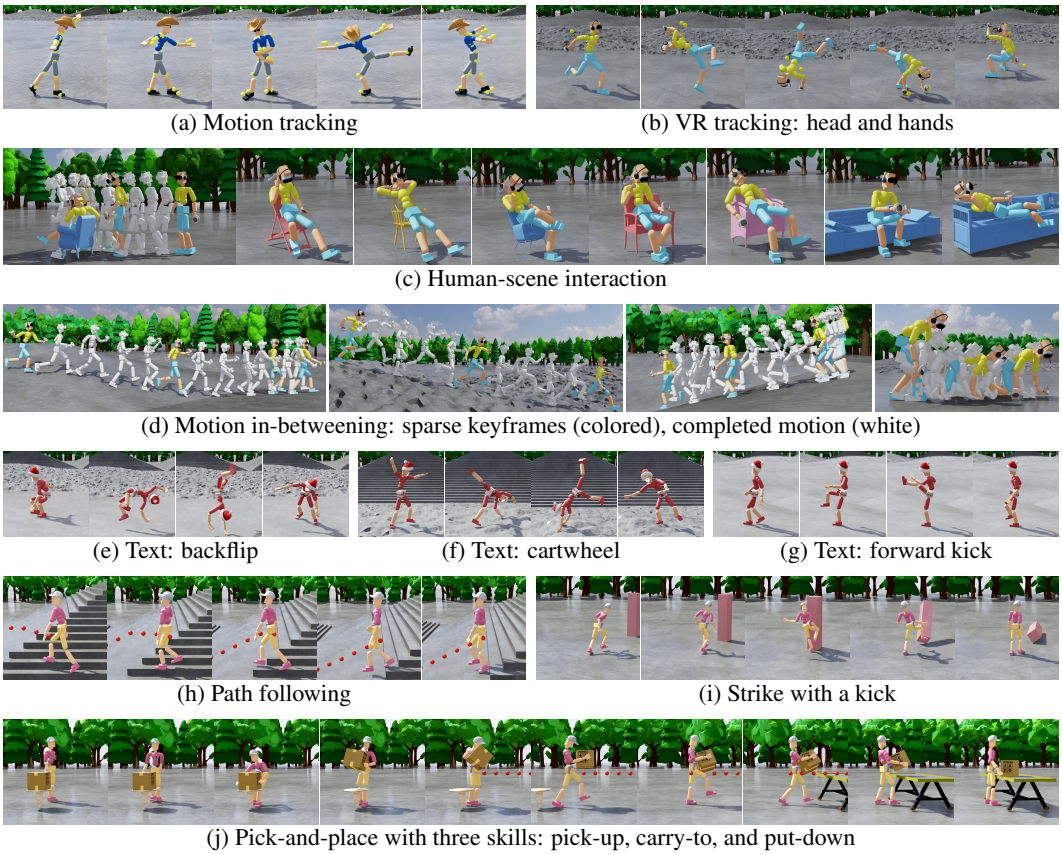
5 EXPERIMENTS

We study how HetSkills acquires heterogeneous skills, reuses a learned text-conditioned skill for downstream adaptation, and composes independently trained modules into continuous behaviors. Figure 3 presents the resulting repertoire, covering reference tracking, sparse motion completion, language-conditioned generation, and reward-driven tasks.

5.1 EXPERIMENTAL SETUP

Experiments use Isaac Lab (Mittal et al., 2025) and ProtoMotions (Tessler et al., 2025), with physics at 120 Hz and control at 30 Hz. Tracking uses AMASS (Mahmood et al., 2019) with PHC filtering (Luo et al., 2023b), which removes clips with non-physical artifacts, such as limb penetration and body floating, or interactions with unmodeled objects; sparse observations from these motions train VR tracking and motion in-betweening. We train the text-to-motion skill on the training split of HumanML3D (Guo et al., 2022). Following SuperPADL (Juravsky et al., 2024), we discard training clips shorter than 2 seconds or longer than 9 seconds to limit idle poses, redundant pauses, and compound actions. SAMP (Hassan et al., 2021) supplies scene-interaction supervision.

Tracking, in-betweening, and SAMP interaction fail if full-body mean per-joint position error (MPJPE) exceeds 0.5 m at any frame. VR applies the same test to mean head-and-hand error, and text-to-motion to paired-reference full-body MPJPE at 0.3 and 0.5 m. TMR retrieval (Petrovich et al., 2023) provides complementary text-to-motion scores: R@N and reported MedR. Network, reward, and training details are provided in the supplementary material.



(a) Motion tracking (b) VR tracking: head and hands

(c) Human-scene interaction

(d) Motion in-betweening: sparse keyframes (colored), completed motion (white)

(e) Text: backflip (f) Text: cartwheel (g) Text: forward kick

(h) Path following (i) Strike with a kick

(j) Pick-and-place with three skills: pick-up, carry-to, and put-down

Figure 3: Heterogeneous skills executed with the shared controller, spanning motion tracking, sparse motion completion, text-conditioned generation, and reward-driven task adaptation. Text-conditioned motions start from a neutral pose; downstream tasks reuse frozen text skills without task-specific motion demonstrations.

5.2 HETEROGENEOUS SKILL ACQUISITION

HetSkills acquires heterogeneous skills within one system by decomposing complex character control into progressively learned, reusable capabilities. This organization accommodates different data sources, conditioning modalities, and supervision signals. Across most quantitatively benchmarked tasks, performance is comparable to or better than the evaluated state-of-the-art baselines.

Table 1: AMASS tracking: success (%, 0.5 m threshold) and MPJPE (mm).

	Train		Test	
Method	Succ.	MPJPE	Succ.	MPJPE
MaskedMimic[1]	99.6	37.9	99.3	53.8
PULSE	99.8	39.2	97.1	54.1
HetSkills-1step	99.9	29.2	99.3	43.6
HetSkills-2step	99.9	28.2	100	39.4
HetSkills-1part	99.9	31.8	100	42.3
HetSkills-5part	100	26.0	100	35.5

Table 2: TMR retrieval following the HumanML3D evaluation protocol. Higher R@N (%) and lower reported MedR are better.

	Neutral-pose			First-frame		
Method	R@1	R@3	MedR	R@1	R@3	MedR
HetSkills	65	88	1.03	74	94	1.01
CLoSD	41	67	2.08	56	79	1.35
MaskedMimic	38	62	2.41	73	94	1.01

Table 3: Text-to-motion success (%) on HumanML3D at two MPJPE thresholds. First/Neutral denote initialization.

	0.3 m		0.5 m	
Method	First	Neutral	First	Neutral
MaskedMimic	76.1	19.8	83.5	34.0
CLoSD	30.9	27.1	44.2	38.5
HetSkills	92.2	60.2	96.9	81.7
w/o RMI	92.6	48.5	97.7	70.0

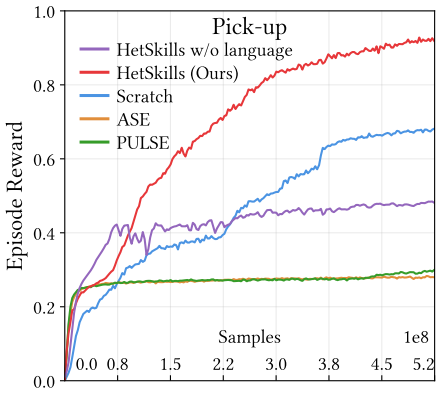


Figure 4: Pick-up learning curves.

Tracking and motion completion. Table 1 reports AMASS tracking results. The default two-part tracker uses two future reference frames; one-frame inference duplicates its input. HetSkills achieves 99.3% test success with one frame and 100% with two. Both variants outperform MaskedMimic (Tessler et al., 2024) and PULSE (Luo et al., 2023a) in test MPJPE, with comparable or higher success rates. The one-part, two-part (2step), and five-part configurations all achieve 100% test success, with MPJPE of 42.3, 39.4, and 35.5 mm, respectively. Tracking precision thus distinguishes configurations even when threshold-based success saturates. We retain the two-part arm/body interface for separate conditioning and downstream routing.

Figure 3(b–d) shows motion-completion skills with different conditions. VR tracking infers full-body motion from head and hand targets, reaching 97.8% test success. For in-betweening, each input specifies one target full-body pose and the time remaining to reach it; successive keyframes are sampled 5–30 frames apart. This skill achieves 100% AMASS test success. On SAMP, the interaction module achieves 96.1% success with the controller frozen. Detailed VR comparisons remain in the supplementary material.

Text-conditioned motion. Following the HumanML3D evaluation protocol (Guo et al., 2022), we evaluate motions initialized from the paired reference's first pose or a neutral pose. Tables 2 and 3 report semantic retrieval and reference adherence, respectively. A text can admit multiple valid motions, so departure from one reference does not itself imply semantic error or physical failure. In retrieval, generated motions query candidate texts; R@N measures top-N retrieval. First-frame initialization tests matched starting conditions; neutral initialization tests robustness to mismatched poses.

At the 0.5 m threshold, HetSkills achieves 96.9% first-frame and 81.7% neutral-pose success, exceeding MaskedMimic and CLoSD (Tevet et al., 2024) under the reported protocols. RMI improves

[1]We evaluate MaskedMimic on the test split defined by PHC, using its official model weights and 10 future frames as conditioning.

neutral success by 11.7 percentage points. By initializing from another motion's terminal state and history, it prepares the text skill to execute new instructions following a preceding behavior; the supplementary video illustrates such skill transitions. These results support MID's use of varied motion contexts to improve the robustness of text-conditioned motion generation. First-frame retrieval is comparable to MaskedMimic. Neutral-pose R@1 reaches 65%, versus 38% for MaskedMimic and 41% for CLoSD. These retrieval results indicate semantic consistency under neutral-pose initialization. Figure 3(e–g) shows backflips, cartwheels, and forward kicks generated from this starting pose.

5.3 Skill Reuse and Composition



Prompt: Long-Horizon Skill Composition

Follow the Standardized Task Description to generate an executable long-horizon skill program for a SMPL character: retrieve a box from a source table, carry it to a nearby target platform, approach and interact with a chair, and finish with open-ended text-conditioned motion.

Pick-up Skill. Initialization: Spawn the humanoid next to the source table with the box. Condition: Provide the box position. Terminate: End when the humanoid stably lifts the box for several consecutive frames.

Carry-to Skill. Initialization: Use the final pick-up state as the initial state. Condition: Generate and follow a path from the current position to the front of the target platform. Terminate: End when the humanoid reaches the neighborhood of the target platform.

Put-down Skill. Initialization: Use the final carry-to state. Condition: Provide the target platform position. Terminate: End when the box remains successfully placed for several consecutive frames.

Path Follow Skill. Initialization: Start from the final state of the put-down skill. Condition: Generate and follow a walking path toward a target location in front of the chair. Terminate: End when the humanoid reaches the target location in front of the chair.

Human-Scene Interaction Skill. Initialization: Start from the final state of the path follow skill. Condition: Specify a scheduled interaction sequence, including turning toward the chair, sitting down, and standing up, and assign a duration to each target pose. Terminate: End when the interaction sequence is completed.

Text-to-Motion Skill. Initialization: Use the final HSI state. Condition: Set both prompts to "a person doing cartwheel". Terminate: When externally stopped.

Figure 5: Long-horizon skill composition. Top: continuous execution of six heterogeneous skills. Bottom: the corresponding LLM prompt specifying initialization, conditions, and termination. Each stage passes the simulated state to the next skill.

Executable skill programs. Figure 5 combines six heterogeneous skills into one continuous behavior. The LLM (Singh et al., 2025) organizes skill descriptions into executable task units (Wang et al., 2023). The shared interface accommodates different conditioning modalities and training objectives. Changing target conditions or skill order enables alternative behaviors using the learned modules without training a transition controller. Across 1,000 executions of the composed program, cumulative success rates reach 99.7% for placing the box on the platform, 99.6% for subsequently sitting on the chair, and 99.5% for completing the full sequence. These results demonstrate reliable composition of heterogeneous skills. Appendix A.3 presents further combinations.

Language-guided task adaptation. Figure 3(h–j) illustrates the downstream tasks. The path-following task tracks online waypoints, while the strike task requires knocking over a pillar. Pick-and-place combines three separate skills: pick-up, carry-to, and put-down. We use simple descriptions of the desired behaviors without elaborate prompt engineering; the instruction sets are provided

in Appendix B. For these three skills, the arm branch uses lifting and carrying descriptions, while the body branch uses standing and directional walking descriptions. Each downstream task is trained within five hours on a single RTX 5090 GPU.

Through motion priors supplied by the learned text skill, language specifies how a task is performed: kicking and pushing instructions support distinct ways to knock over the same target. Each variant learns its own task guidance while reusing the frozen text skill and controller, adapting learned motions to new interaction goals without task-specific demonstrations. The pick-up ablation (Figure 4) further shows that language guidance helps HetSkills converge to effective lifting behavior, while its removal leads to a lower-reward plateau, as with the baselines. This suggests that language-conditioned priors make coordinated bimanual lifting easier to discover and help avoid suboptimal solutions during exploration. Additional learning curves for path following and strike push are provided in Appendix A.2.

Reusing part-wise motion priors. Figure 6 examines how the learned text skill is reused during pick-and-place. The arm branch shifts from lifting-related commands toward carrying-related commands as transport begins, while the body branch redistributes weights among directional walking and standing descriptions. Their distinct temporal profiles show that task guidance routes the two body partitions separately. In this example, the three tasks use the same two arm prompts and three body prompts, while retaining task-specific guidance modules. Each prompt queries the same frozen T2M mapping with the current state and history, so candidate commands evolve even when the instruction remains fixed. The gating weights mix these commands before decoding, while task-conditioned residuals refine the mixture.

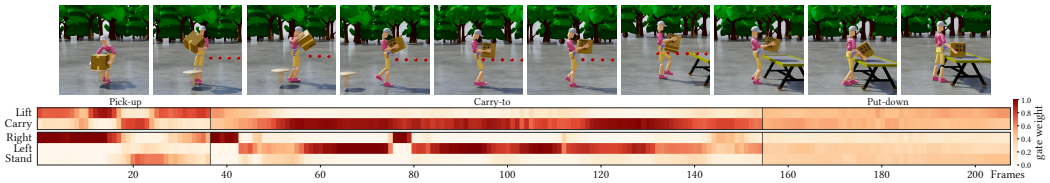


Figure 6: Part-wise gating during pick-and-place, with snapshots every 20 frames. Rows show arm (Lift/Carry) and body (Right/Left/Stand) prior weights; darker red indicates higher weight (0–1), and vertical lines delimit stages.

The same text skill thus serves two roles: it supplies behavior priors while downstream guidance is trained and remains directly executable afterward, as the final text-conditioned stage of Figure 5 illustrates. Once trained, a downstream skill retains its instruction set and guidance modules for later execution. The combined rollout reuses these acquired modules with fixed parameters, so the LLM's role is limited to organizing their calls. These roles connect the acquisition of a skill to its continued use within an expanding repertoire.

6 Conclusions and Limitations

We presented HetSkills, a physics-based character control framework for acquiring and reusing heterogeneous skills. Each learned skill remains independently executable and available for composition with other skills. These skills can also serve as control foundations, expert teachers, or behavioral priors for subsequent learning. A shared latent interface accommodates skills from different data sources, supervision forms, and task objectives, while a standardized task description organizes their invocation and long-horizon composition. Through part-wise control, MID, and language-guided adaptation, HetSkills combines diverse skill acquisition with robust text-conditioned generation and downstream learning without task-specific motion demonstrations. These results provide a practical foundation for reusable, composable, and extensible character control.

Although HetSkills demonstrates progressive integration across several representative skill categories, the current experiments do not yet fully cover larger-scale skill accumulation, more complex long-horizon composition, or more open-ended human-object and human-scene interactions. MID improves robustness to different initial states and motion histories, but the text-to-motion skill remains affected by the quality of language-motion annotations.

AI USE STATEMENT

We used generative AI tools to assist with coding, manuscript drafting, and language polishing. We manually reviewed all AI-assisted text and code. All references were identified and checked manually without AI assistance. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Implementation details and evaluation protocols are provided in the paper and appendix to support reproducibility. A concise implementation of our method is available in our anonymous repository: `https://anonymous.4open.science/r/HetSkills-F4E7/README.md`. Complete video materials are included in the supplementary materials submitted with this paper, and we encourage reviewers to view them alongside the manuscript. We will release our code, training configurations, and evaluation scripts as open source.

REFERENCES

Jinseok Bae, Jungdam Won, Donggeun Lim, Cheol-Hui Min, and Young Min Kim. PMP: Learning to physically interact with environments using part-wise motion priors. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 1–10, 2023.

Jinseok Bae, Younghwan Lee, Donggeun Lim, and Young Min Kim. PLT: Part-wise latent tokens as adaptable motion priors for physically simulated characters. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 1–10, 2025.

Zhiyang Dou, Xuelin Chen, Qingnan Fan, Taku Komura, and Wenping Wang. C·ASE: Learning conditional adversarial skill embeddings for physics-based characters. In *ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia)*, pp. 1–11, 2023.

Haotian Fu, Pratyusha Sharma, Elias Stengel-Eskin, George Konidaris, Nicolas Le Roux, Marc-Alexandre Côté, and Xingdi Yuan. Language-guided skill learning with temporal variational inference. *arXiv preprint arXiv:2402.16354*, 2024.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3D human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5152–5161, 2022.

Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020.

Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. Stochastic scene-aware motion prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11374–11384, 2021.

Xiaoyu Huang, Takara Truong, Yunbo Zhang, Fangzhou Yu, Jean Pierre Sleiman, Jessica Hodgins, Koushil Sreenath, and Farbod Farshidian. Diffuse-CLoC: Guided diffusion for physics-based character look-ahead control. *ACM Transactions on Graphics (TOG)*, 44(4):1–12, 2025.

Inwoo Hwang, Bing Zhou, Young Min Kim, Jian Wang, and Chuan Guo. SceneMI: Motion in-betweening for modeling human-scene interaction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6034–6045, 2025.

Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. PADL: Language-directed physics-based character control. In *ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia)*, pp. 1–9, 2022.

Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. SuperPADL: Scaling language-directed physics-based control with progressive supervised distillation. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 1–11, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yuxuan Kuang, Haoran Geng, Amine Elhafsi, Tan-Dzung Do, Pieter Abbeel, Jitendra Malik, Marco Pavone, and Yue Wang. SkillBlender: Towards versatile humanoid whole-body loco-manipulation via skill blending. *arXiv preprint arXiv:2506.09366*, 2025.

Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character controllers using motion VAEs. *ACM Transactions on Graphics (TOG)*, 39(4):40–1, 2020.

Libin Liu and Jessica Hodgins. Learning to schedule control fragments for physics-based characters using deep Q-learning. *ACM Transactions on Graphics (TOG)*, 36(3):1–14, 2017.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions On Graphics (TOG)*, 34(6): 248:1–248:16, October 2015.

Zhengyi Luo, Ye Yuan, and Kris M Kitani. From universal humanoid control to automatic physically valid character creation. *arXiv preprint arXiv:2206.09286*, 2022.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582*, 2023a.

Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10895–10904, 2023b.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Jinhyung Park, David Sami, Zi Wang, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi "Jim" Fan, and Yuke Zhu. SONIC: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11 (117):eaed4592, 2026.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. AMASS: Archive of motion capture as surface shapes. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5442–5451, 2019.

Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control. *arXiv preprint arXiv:1811.11711*, 2018.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Munoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac Lab: A GPU-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

Yuxuan Mu, Ziyu Zhang, Yi Shi, Minami Matsumoto, Kotaro Imamura, Guy Tevet, Chuan Guo, Michael Taylor, Chang Shu, Pengcheng Xi, et al. SMP: Reusable score-matching motion priors for physics-based character control. *arXiv preprint arXiv:2512.03028*, 2025.

Liang Pan, Zeshi Yang, Zhiyang Dou, Wenjia Wang, Buzhen Huang, Bo Dai, Taku Komura, and Jingbo Wang. TokenHSI: Unified synthesis of physical human-scene interactions through task tokenization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5379–5391, 2025.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. MCP: Learning composable hierarchical control with multiplicative compositional policies. *Conference on Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)*, 40(4):1–20, 2021.

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

Mathis Petrovich, Michael J Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9488–9497, 2023.

Jiawei Ren, Mingyuan Zhang, Cunjun Yu, Xiao Ma, Liang Pan, and Ziwei Liu. InsActor: Instruction-driven physics-based characters. *Conference on Neural Information Processing Systems (NeurIPS)*, 36:59911–59923, 2023.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Yi Shi, Yifeng Jiang, Chen Tessler, and Xue Bin Peng. GPC: Large-scale generative pretraining for transferable motor control. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2026.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*, volume 1. MIT press Cambridge, 1998.

Jie Tan, Karen Liu, and Greg Turk. Stable proportional-derivative controllers. *IEEE Computer Graphics and Applications*, 31(4):34–44, 2011.

Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. CALM: Conditional adversarial latent models for directable virtual characters. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 1–9, 2023.

Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. MaskedMimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions On Graphics (TOG)*, 43(6):1–21, 2024.

Chen Tessler, Yifeng Jiang, Xue Bin Peng, Erwin Coumans, Yi Shi, Haotian Zhang, Davis Rempe, Gal Chechik, and Sanja Fidler. ProtoMotions3: An open-source framework for humanoid simulation and control. `https://github.com/NVLabs/ProtoMotions/`, 2025.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit H Bermano, and Michiel van de Panne. CLoSD: Closing the loop between simulation and diffusion for multi-task character control. *arXiv preprint arXiv:2410.03441*, 2024.

Takara Everest Truong, Michael Piseno, Zhaoming Xie, and Karen Liu. PDP: Physics-based character animation via diffusion policy. In *ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia)*, pp. 1–10, 2024.

Ron Vainshtein, Zohar Rimon, Shie Mannor, and Chen Tessler. Task tokens: A flexible approach to adapting behavior foundation models. *arXiv preprint arXiv:2503.22886*, 2025.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(11), 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Conference on Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Wenjia Wang, Liang Pan, Zhiyang Dou, Jidong Mei, Zhouyingcheng Liao, Yuke Lou, Yifan Wu, Lei Yang, Jingbo Wang, and Taku Komura. SIMS: Simulating stylized human-scene interactions with retrieval-augmented script generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14117–14127, 2025.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics (TOG)*, 39(4): 33–1, 2020.

Yan Wu, Korrawe Karunratanakul, Zhengyi Luo, and Siyu Tang. UniPhys: Unified planner and controller with diffusion for flexible physics-based character control. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13214–13224, 2025a.

Zhen Wu, Jiaman Li, Pei Xu, and C Karen Liu. Human-object interaction from human-level instructions. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11176–11186, 2025b.

Pei Xu, Zhen Wu, Ruocheng Wang, Vishnu Sarukkai, Kayvon Fatahalian, Ioannis Karamouzas, Victor Zordan, and C Karen Liu. Learning to ball: Composing policies for long-horizon basketball moves. *ACM Transactions on Graphics (TOG)*, 44(6):1–14, 2025.

Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. ControlVAE: Model-based learning of generative controllers for physics-based characters. *ACM Transactions on Graphics (TOG)*, 41 (6):1–16, 2022.

Heyuan Yao, Zhenhua Song, Yuyang Zhou, Tenglong Ao, Baoquan Chen, and Libin Liu. MoConVQ: Unified physics-based motion control via scalable discrete representations. *ACM Transactions on Graphics (TOG)*, 43(4):1–21, 2024.

Ziyu Zhang, Sergey Bashkirov, Dun Yang, Yi Shi, Michael Taylor, and Xue Bin Peng. Physics-based motion imitation with adversarial differential discriminators. In *ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia)*, pp. 1–12, 2025.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5745–5753, 2019.

Qingxu Zhu, He Zhang, Mengting Lan, and Lei Han. Neural categorical priors for physics-based character control. *ACM Transactions on Graphics (TOG)*, 42(6):1–16, 2023.

Appendix Contents

A Additional Experimental Results 14
B Downstream Instruction Sets 17
C Implementation Details 18
D Task Rewards and Settings 21
E Training Time and Compute 23

A Additional Experimental Results

A.1 Tracking, Motion Completion, and Text–Motion

Table 4 reports VR-driven body tracking on the training and test splits of AMASS. Only the head and both hands are provided as goal constraints. A rollout is considered a failure if the mean tracking error of these three end-effectors exceeds 0.5 m at any frame, while MPJPE measures full-body reconstruction error. When all methods are conditioned on a single future frame for a fair comparison, HetSkills achieves the best performance.[2] With only three upper-body endpoints as constraints, the lower-body motion is underdetermined: multiple plausible configurations can satisfy the observations, so full-body reconstruction error does not fully characterize motion quality.

Table 4: VR tracking result on AMASS.

	Train		Test	
Method	Succ.	MPJPE	Succ.	MPJPE
HetSkills	99.9	39.9	97.8	82.3
PULSE	99.5	57.8	93.4	88.6
MaskedMimic-1step	91.4	118.2	89.9	154.9
MaskedMimic-10step	98.6	50.0	98.1	58.1
ASE	79.8	103.0	37.6	120.5

Table 5: TMR six-class Top-1 accuracy (%) under same- and different-class histories.

History	Method	Top-1 ↑
Same-class history	HetSkills	90.83
	w/o RMI	89.17
Different-class history	HetSkills	92.50
	w/o RMI	82.33

For motion in-betweening, HetSkills achieves success rates of 99.9% on the AMASS training split and 100% on the test split, using a 0.5 m full-body MPJPE failure threshold. Success rate alone is not a comprehensive measure, since motion in-betweening admits a range of plausible intermediate trajectories. A rollout that deviates from the reference is not necessarily incorrect.

Table 5 tests whether text instructions can override motion history. At equal training steps, both models start from identical physical states and two-second histories. Six-second rollouts with twelve descriptions from six action classes yield 120 same-class and 600 different-class cases per model. Frozen TMR embeddings select the class with the highest mean similarity to its two descriptions; Top-1 measures category matching rather than exact-caption retrieval. HetSkills and w/o RMI perform similarly with same-class histories (90.83% vs. 89.17%). With different-class histories, HetSkills improves accuracy from 82.33% to 92.50%, supporting better switching to the requested action category in this test.

A.2 Downstream Task Learning

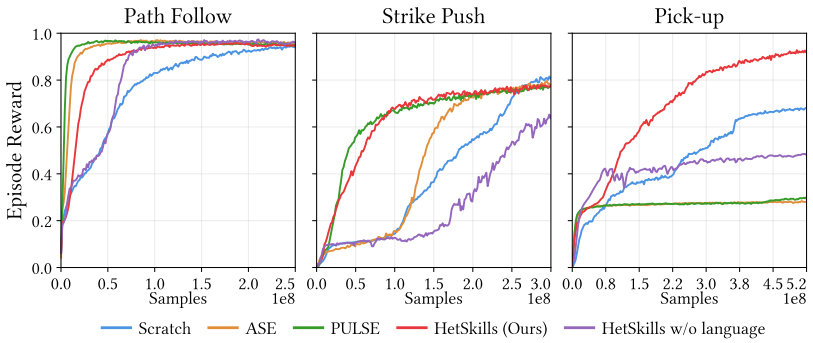


Figure 7: Downstream episode rewards versus environment samples, averaged over three random seeds. "HetSkills w/o language" ablates language guidance.

[2] MaskedMimic uses 10 future frames by default in its official evaluation.

Figure 7 compares HetSkills with Scratch, ASE, and PULSE, which optimize task rewards without language conditioning. Path-following rewards are comparable, although ASE and PULSE occasionally walk backward. On strike push, the baselines can knock over the target but often adopt awkward postures, whereas HetSkills reuses natural pushing motions from its text-conditioned skill. HetSkills achieves the highest pick-up reward within the reported sample budget.

Language guidance ablation. Without language guidance, HetSkills eventually reaches a similar reward on path follow, but learns strike push more slowly and attains a lower reward within the same sample budget. The largest gap occurs on pick-up, where the ablated variant plateaus well below the full model. This suggests that language guidance helps direct exploration toward task-relevant behaviors, especially coordinated bimanual lifting.

A.3 Long-Horizon Skill Composition Examples

More long-horizon skill composition examples are provided to show different combinations and task sequences. Each example follows the same standardized task interface, where the overall behavior is decomposed into a sequence of executable skill units with specified initialization, condition, and termination criterion. The examples involve different combinations and orderings of heterogeneous skills, further illustrating that HetSkills can flexibly organize skills learned from different supervision forms and task settings into coherent long-horizon behaviors.

The two-box task in Figure 8 requires placing both boxes on the same target desk in sequence. Across 1,000 executions, cumulative success rates reach 99.7% for the first placement and 99.1% for completing both placements. Six trials that successfully complete the first placement fail during the second, mainly because the two boxes interfere with each other on the desk. The high overall success rate demonstrates reliable skill reuse and composition in sequential object manipulation.



Prompt A: Two-Box Sequential Placement

Please follow the Standardized Task Description to generate an executable long-horizon skill program for a SMPL character that moves two boxes, one at a time, from separate source tables to the same target desk.

Pick-up Skill (Box 1). Initialization: Start beside the first source table. Condition: Provide Box 1's position. Terminate: End after a stable lift for several consecutive frames.

Carry-to Skill (Box 1). Initialization: Continue from pick-up. Condition: Generate and follow a path to the target desk while carrying Box 1. Terminate: End near the desk.

Put-down Skill (Box 1). Initialization: Continue from carry-to. Condition: Provide Box 1's target position on the desk. Terminate: End after successful placement for several consecutive frames.

Path Follow Skill. Initialization: Start from the final state of the first put-down skill. Condition: Generate and follow a walking path toward the second source table. Terminate: End when the humanoid reaches the vicinity of the second source table.

Pick-and-Place Skill (Box 2). Initialization: Start from the final state of the path follow skill, with the second box on the second source table. Condition: Reuse the same pick-up, carry-to, and put-down procedure for Box 2 by providing the second box position, the path from the second source table to the desk, and the second target placement position. Terminate: End when the second box is successfully placed on the desk.

Figure 8: Long-horizon skill composition results for the two-box sequential placement task. The humanoid first picks up the first box from the source table and places it on the target desk, then walks to the second source table, picks up the second box, carries it back to the desk, and places it at the target position.



Figure 9: Long-horizon skill composition results for the sit-strike-celebrate task. The humanoid walks to the chair, sits down and stands up, approaches the target object, strikes it down, and then performs a jumping celebration motion.

Prompt B: From Sitting to Striking and Celebration

Please follow the Standardized Task Description to generate an executable long-horizon skill program for a SMPL character. The task depicts a humanoid that walks to a chair, sits down and stands up, moves toward a target object and strikes it down, and then celebrates with an open-ended text-conditioned jumping motion. The detailed skill sequence is specified below.

Path Follow Skill. Initialization: Spawn the humanoid at the scene origin. Condition: Generate and follow a straight walking path toward a target location in front of the chair. Terminate: End when the humanoid reaches the target location in front of the chair.

Human-Scene Interaction Skill. Initialization: Start from the final state of the path follow skill. Condition: Provide a scheduled inpainting target sequence, including sitting down on the chair, holding the seated pose, and standing back up, with a specified duration for each target frame. Terminate: End when the full interaction schedule is completed.

Strike Skill. Initialization: Start from the final state of the human-scene interaction skill. Condition: Provide the target object position, and generate a full-body striking motion that moves the humanoid toward the target object and knocks it down. Terminate: End when the target object remains in a knocked-down orientation.

Text-to-Motion Skill. Initialization: Start from the final state of the strike skill. Condition: Set the text prompt to "a person jumping up and down, with their hands above their head." Terminate: Continue until the episode is externally stopped.



Figure 10: Long-horizon rollout of sequential text-conditioned motion generation. The humanoid switches across three language conditions and performs a left punch, a forward kick, and a cartwheel in sequence.

Prompt C: Text-Conditioned Motion Sequence

Please follow the Standardized Task Description to generate an executable long-horizon skill program for a SMPL character. The task depicts a humanoid that sequentially performs three text-conditioned motions, including punching, kicking, and cartwheeling, each for a specified duration. The detailed skill sequence is specified below.

Text-to-Motion: Punch. Condition: Set the language embedding to "person was fighting with a left punch". Terminate: End after 5 seconds.

Text-to-Motion: Kick. Condition: Switch the language embedding to "a person kicks forward." Terminate: End after 3 seconds.

Text-to-Motion: Cartwheel. Condition: Switch the language embedding to "a person does a cartwheel." Terminate: End after 5 seconds.

A.4 Additional Motion Examples

Figure 11 provides additional motion examples. The human-scene interaction examples show sitting, reclining, and other interactions with chairs, sofas, and tables. The text-to-motion examples cover circular walking, dance, crouching, and lying down, all generated from a neutral pose initialization.



(a) Over a hundred characters interacting with diverse furniture simultaneously.



(b) A person walks in a circle.



(c) A person dances the waltz.



(d) A person crouches and then lies down.



(e) A person dances ballet.

Figure 11: Additional motion examples. The top panel shows human-scene interactions with diverse furniture. The remaining sequences show text-to-motion generation from a neutral pose initialization.

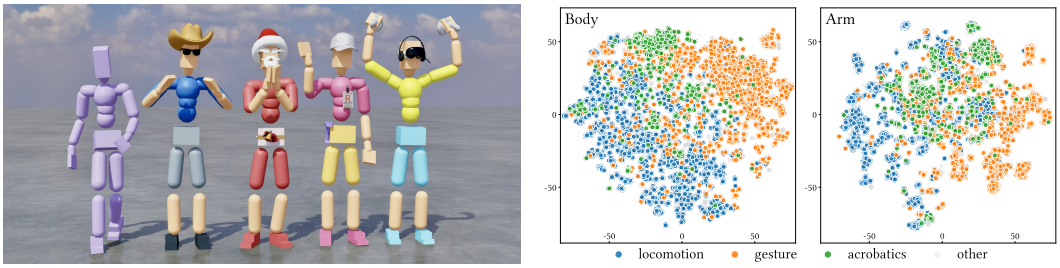
B Language Instruction Sets for Downstream Tasks

For downstream task adaptation, we provide fixed language instruction sets for the arm and body branches separately. Each instruction specifies a candidate semantic motion prior in the text-to-motion latent space, such as lifting, carrying, standing, or moving in a particular direction. Given the current task context, the gating network $\mathcal{G}_p$ predicts part-wise mixture weights over these candidate instructions and dynamically blends the corresponding motion priors. This design allows the downstream policy to compose task-relevant behaviors from reusable language-conditioned priors, while keeping the pretrained motion decoder fixed. The instruction sets used for each downstream task are listed in Table 6.

Table 6: Language-prior instruction sets for downstream adaptation. The table lists the natural-language prompts used by $\mathscr{T}^{t2m}$ to construct candidate motion priors for each downstream task and body branch. The arm and body prompt sets provide part-specific semantic priors for tasks such as pick-and-place, strike, and path follow, while the MoE gating network dynamically blends these priors according to the current task observation.

Task	Branch	Instructions
Pick-and-Place (all stages)	Arm	"a man lifts something on his left and places it down on his right" (Lift) "the person is moving or carrying something" (Carry)
	Body	"a person walks to his right" (Right) "the person was walking forward then turn left" (Left) "a man stands still" (Stand)
Strike (kick)	Arm & Body	"a person does a high kick with their right leg" "this person kicked with their right leg" "a man runs forward fast"
Strike (push)	Arm & Body	"a person pushes something forward with both hands" "a person walks up to an object and shoves it" "a man runs forward fast"
Path Follow	Arm & Body	"a person walks to his right" "the person was walking forward then turn left" "a person runs to the right then runs to the left then back to the middle"

C Implementation Details



(a) Physical model and skill appearances.

(b) Body and arm latent distributions.

Figure 12: **Character visualization and learned latents.** (a) The neutral humanoid is the simulated model; the cowboy, Santa, courier, and gamer distinguish tracking, text-to-motion, downstream adaptation, and motion completion, respectively. Their appearances do not change the underlying physical character or controller. (b) t-SNE projections (van der Maaten & Hinton, 2008) reveal category-dependent structure in the body and arm latents, learned without motion-category supervision.

All skills control the same physically simulated neutral SMPL humanoid (Loper et al., 2015) with 69 degrees of freedom, following prior physics-based character control works (Luo et al., 2022; 2023b; Tessler et al., 2024). Figure 12 shows the visual appearances used to distinguish skill families and the arm/body latent distributions learned through tracking.

Shared settings. We optimize trainable policy and prior modules with Adam (Kingma & Ba, 2014), using learning rates of 2×10^{-5} for policies and priors and 10^{-4} for critics. Each arm/body latent has $d_p = 64$ dimensions, and Transformer tokens have 256 dimensions. Unless specified otherwise, latent regularization uses $\lambda_{\text{mr}} = 10^{-3}$, $\lambda_{\text{ar}} = 5 \times 10^{-3}$, and $\phi = 0.99$.

C.1 TRACKING

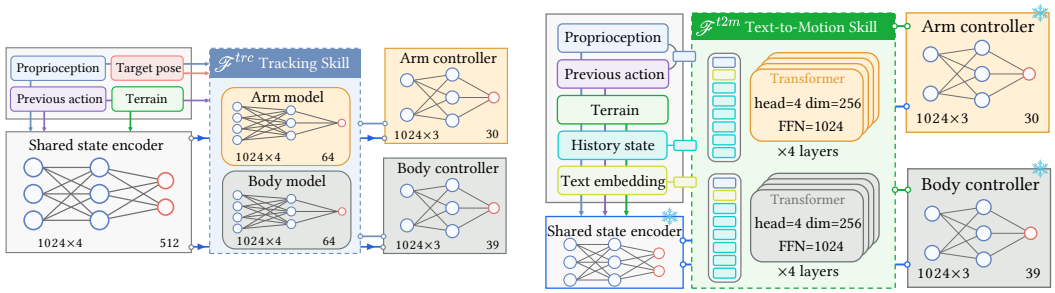
State and goal representation. All rotations use the continuous 6D parameterization (Zhou et al., 2019). We write $\delta_r(\cdot)$ for a quantity expressed in a frame centered at the current root position and aligned with its heading, and $\delta_g(\cdot)$ for its target-minus-current difference. The state is

$$s^t = \big(\delta_r(p^t),\ \delta_r(\theta^t),\ \delta_r(\dot{\theta}^t),\ \delta_r(v^t),\ h^t_{\text{root}},\ o^t_{\text{terrain}}\big), \tag{3}$$

where p^t, θ^t, $\dot{\theta}^t$, and v^t denote body positions, rotations, angular velocities, and linear velocities, respectively. The root entry is omitted from the position feature. h^t_{root} denotes root height above the ground, and o^t_{terrain} denotes sampled terrain heights. The goal $g^t = [\hat{f}^{t+1}, \ldots, \hat{f}^{t+K}]$ contains future reference poses represented as

$$\hat{f} = \big(\delta_r(\hat{p}),\ \delta_g(p),\ \delta_r(\hat{\theta}),\ \delta_g(\theta),\ \delta_g(v),\ \delta_g(\dot{\theta})\big). \tag{4}$$

These terms encode target body positions and position errors in the root frame, target rotations and rotation errors, and linear/angular velocity differences. The tracker receives s^t, g^t, and a^{t-1} and predicts latent commands decoded through Eq. (1). The resulting target joint rotations are converted to torques by PD controllers (Tan et al., 2011).



(a) Tracking architecture.

(b) Text-to-motion architecture.

Figure 13: **Tracking and text-to-motion networks.** (a) Part-wise tracking encoders produce arm/body latents, decoded with a shared state feature into joint actions. (b) Independent Transformer branches replace the tracking encoders to predict latents from text and state history; snowflakes mark the frozen state encoder and controllers. Layer widths, depths, and output dimensions are shown in the panels.

Architecture. Figure 13a gives the network dimensions. The arm branch controls both arms and hands, while the body branch controls the remaining joints. Both decoders receive the shared full-body feature $c^t = \mathcal{E}_s(s^t, a^{t-1})$ for coordination. We train with $K = 2$ future frames for tracking and expert supervision; at inference, a single reference frame can be duplicated to fill this input. After training, $\mathcal{E}_s$, $\mathcal{D}_a$, and $\mathcal{D}_b$ are frozen for reuse by later skills, while the tracker remains available for execution and teaching.

Training and rewards. We use PPO with exploration only in the final joint-action space, using a diagonal Gaussian with fixed log standard deviation -2.9. Episodes terminate when tracking error exceeds $0.5\,\text{m}$. Reference State Initialization (RSI) samples a reference frame with probability 0.8; otherwise, the episode starts at the first frame. Prioritized motion sampling draws motions with higher failure rates more frequently (Luo et al., 2023a).

The reward tracks global joint position (gp), global joint rotation (gr), joint velocity (jv), joint angular velocity (jav), and root height (rh):

$$r^t = \sum_{j \in \mathcal{J}} w_j \exp(-k_j e^t_j) + w_{\text{ct}} r^{t,\text{ct}} + w_{\text{sm}} r^{t,\text{sm}} + w_{\text{eg}} r^{t,\text{eg}}, \tag{5}$$

where e^t_j is the mean squared error over all scalar components for $j \in \{\text{gp}, \text{jv}, \text{jav}, \text{rh}\}$. For global joint rotations, e^t_{gr} is the mean squared geodesic angle between the current and reference rotations

over all joints, with angles measured in radians. In the order $\mathcal{J} = (\mathrm{gp}, \mathrm{gr}, \mathrm{jv}, \mathrm{jav}, \mathrm{rh})$, the weights are $(0.5, 0.3, 0.1, 0.1, 0.2)$ and exponential coefficients are $(100, 5, 0.5, 0.1, 100)$. Contact mismatch, action smoothness, and energy penalties use $w_{\mathrm{ct}} = -0.1$, $w_{\mathrm{sm}} = -0.02$, and $w_{\mathrm{eg}} = -10^{-5}$. The energy penalty is clipped below at -0.5; reference contacts are smoothed over 7 frames.

The encoders produce deterministic latents, regularized for compactness and AR(1) temporal consistency (Merel et al., 2018):

$$\begin{aligned} \mathcal{L}_{\mathrm{mr}} &= \sum_{p\in\{a,b\}} \mathbb{E}\left[\frac{1}{d_p}\|z_p^t\|_2^2\right], \\ \mathcal{L}_{\mathrm{ar}} &= \sum_{p\in\{a,b\}} \mathbb{E}\left[\|z_p^t - \phi z_p^{t-1}\|_2\right], \\ \mathcal{L}_{\mathrm{trc}} &= \mathcal{L}_{\mathrm{ppo}} + \lambda_{\mathrm{mr}}\mathcal{L}_{\mathrm{mr}} + \lambda_{\mathrm{ar}}\mathcal{L}_{\mathrm{ar}}. \end{aligned} \tag{6}$$

C.2 Text-to-Motion

Inputs and initialization. The two Transformer branches in Fig. 13b have separate parameters. Each receives one current-state token encoding (s^t, a^{t-1}), one language token from the frozen TMR text encoder (Petrovich et al., 2023), and 6 history tokens uniformly sampled from a 60-step window (2 seconds at 30 Hz). They predict $z_p^t = \mathscr{F}_p^{t2m}(s^t, a^{t-1}, H^t, e)$ for the frozen controller. For MID initialization (Section 4.2), RSI ($p_{\mathrm{rsi}} = 0.7$) samples a frame uniformly from the target clip. RMI ($p_{\mathrm{rmi}} = 0.2$) instead samples another clip uniformly, initializes from its terminal frame, and fills the history buffer with its preceding states, retaining the target text-motion pair. In the remaining 10% of training episodes, the character is initialized from the first frame of the target clip.

Expert supervision. At each state visited by the student, the frozen tracker uses the target motion's future reference frames to supply an expert action $\hat{a}^t$ (Ross et al., 2011); those frames are unavailable to the student. We supervise the decoded joint actions and apply the latent regularizers from Eq. (6):

$$\begin{aligned} \mathcal{L}_{\mathrm{BC}} &= \mathbb{E}\left[\|a^t - \hat{a}^t\|_2\right], \\ \mathcal{L}_{\mathrm{t2m}} &= \mathcal{L}_{\mathrm{BC}} + \lambda_{\mathrm{mr}}\mathcal{L}_{\mathrm{mr}} + \lambda_{\mathrm{ar}}\mathcal{L}_{\mathrm{ar}}. \end{aligned} \tag{7}$$

Gradients pass through the frozen controller to update only the student encoders.

C.3 Motion Completion

Sparse targets are represented as $\mathcal{O}^t = \{(t_k, \tilde{g}_k^t)\}_{k=1}^K$, where t_k is a relative time offset and $\tilde{g}_k^t$ specifies partial-body states or a future full-body pose. Each completion encoder conditions on the current proprioception, previous action, and sparse targets. We train the following three instances independently, all reusing the frozen state encoder and decoders.

VR tracking. The condition contains the full kinematic states of the head and both hands from the next reference frame, with no history input. A four-layer MLP with 1024 hidden units per layer outputs 128 values, split into the arm and body latents. Training uses the tracking teacher with $\mathcal{L}_{\mathrm{BC}} + \lambda_{\mathrm{mr}}\mathcal{L}_{\mathrm{mr}}$. We omit AR(1) regularization to prioritize immediate fidelity to the spatial targets.

Motion in-betweening. Two part-wise Transformer branches receive a future full-body target pose, its time offset, and 3 historical states uniformly sampled from a 30-step window, in addition to the current state and previous action. The target is sampled uniformly from 5–30 frames ahead and resampled once reached. Training uses the tracking teacher with the loss in Eq. (7), including both latent regularizers.

Human-scene interaction. We use the same temporal conditioning, history window, and target-sampling scheme for interaction poses from SAMP (Hassan et al., 2021). These motions are outside the tracker's training distribution, so we train with PPO using 6 mini-epochs per update. The reward follows Eq. (5) with the same coefficients, omitting the root-height and contact-mismatch terms.

C.4 Downstream Adaptation

Prior reuse and optimization. Each part's fixed instruction set $\mathcal{I}_p = \{e_p^m\}_{m=1}^{M_p}$ queries the same frozen text-to-motion skill to obtain state-dependent candidate latents. The counts M_p may differ between parts. Task-specific gating and residual networks combine and refine the candidates through Eq. (2); only these modules and the critic are trained. Instruction sets and task observations/rewards are provided in Sections B and D. We use PPO with 32-step rollouts and 4 mini-epochs per update. Exploration samples latent commands around the adapted mean before decoding, with log standard deviation annealed from -2.5 to -3.0 between epochs 500 and 1000. Algorithm 1 summarizes training; the resulting instruction sets and guidance modules are retained as an independently executable skill.

ALGORITHM 1: Language-Guided Downstream Adaptation

Input: Frozen text priors $\mathscr{F}_p^{t2m}$, state encoder $\mathcal{E}_s$, and decoders $\mathcal{D}_p$; instruction sets $\mathcal{I}_p = \{e_p^m\}_{m=1}^{M_p}$, $p \in \{a, b\}$; trainable gating networks $\mathcal{G}_p$, residual networks $\mathcal{R}_p$, and critic V_ψ

for *each training iteration* **do**
  for *each rollout step, in parallel environments* **do**
    Observe s^t, a^{t-1}, H^t, and o_{task}^t;
    Compute $z_p^{t,m} = \mathscr{F}_p^{t2m}(s^t, a^{t-1}, H^t, e_p^m)$ for all p, m;
    Compute routing weights and adapted means $\hat{z}_p^t$ via Eq. (2);
    Sample $z^t \sim \mathcal{N}([\hat{z}_a^t; \hat{z}_b^t], \mathrm{diag}(\sigma^2))$; record its log probability;
    Decode z^t into joint actions a^t using Eq. (1);
    Execute a^t and store the transition, reward, and next state;
  end
  Compute advantages and returns using generalized advantage estimation;
  For each PPO mini-epoch and mini-batch, update $\mathcal{G}_p$ and $\mathcal{R}_p$ with Eq. (8) and fit V_ψ to returns;
end
return retained skill $\mathscr{F}^{task^*} = \{\mathcal{I}_p, \mathcal{G}_p, \mathcal{R}_p\}_{p\in\{a,b\}}$;

Latent regularization. Let $\Delta z_p^t = \mathcal{R}_p(s^t, a^{t-1}, o_{\text{task}}^t)$ denote the residual for part p, and let $\hat{z}^t = [\hat{z}_a^t; \hat{z}_b^t]$ be the adapted mean after prior mixing and residual refinement. We penalize residual magnitudes beyond a prescribed bound and abrupt changes in the adapted mean:

$$\begin{aligned} \mathcal{L}_{\text{lmp}} &= \sum_{p\in\{a,b\}} \mathbb{E}\left[\frac{1}{d_p}\left\|\max\big(|\Delta z_p^t| - z_{\text{bound}}, 0\big)\right\|_2^2\right], \\ \mathcal{L}_{\text{smooth}} &= \mathbb{E}\big[\|\hat{z}^t - \hat{z}^{t-1}\|_2^2\big], \\ \mathcal{L}_{\text{task}^*} &= \mathcal{L}_{\text{ppo}} + \lambda_{\text{lmp}}\mathcal{L}_{\text{lmp}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}}. \end{aligned} \tag{8}$$

The absolute value and maximum are applied elementwise, with $z_{\text{bound}} = 0.1$, $\lambda_{\text{lmp}} = 10.0$, and $\lambda_{\text{smooth}} = 0.1$. The magnitude penalty acts on the residual, while the smoothness penalty acts on the adapted mean.

D Downstream Task Rewards and Settings

This section describes the reward functions and environment settings used for downstream task adaptation. All rewards are designed to encourage task completion while regularizing the motion with energy penalties, so that the adapted behaviors remain physically plausible and compatible with the pretrained motion prior.

Path Follow. The path follow reward encourages the character root to track an online-generated target path in the horizontal plane:

$$r = \exp\big(-\|p_{\text{target}}^{xy} - p_{\text{root}}^{xy}\|^2\big) - \lambda \cdot \mathcal{P}_{\text{leg}} \tag{9}$$

where p_{target}^{xy} and p_{root}^{xy} denote the horizontal positions of the current path target and the character root, respectively. The leg energy term is defined as

$$\mathcal{P}_{\text{leg}} = \sum_i |\tau_i \dot{q}_i|, \tag{10}$$

where the summation is taken over the 24 leg DoFs, including the bilateral hip, knee, ankle, and toe joints. We set $\lambda = 10^{-5}$ and clip the resulting energy penalty to $[-0.1, 0]$. The path is generated online with a maximum speed of $5\,\mathrm{m/s}$ and a maximum acceleration of $2\,\mathrm{m/s^2}$. An episode terminates if the horizontal distance between the root and the current path target exceeds $1.5\,\mathrm{m}$.

Strike. The strike reward encourages the character to approach the target, face it, move toward it, and apply a motion that tilts the target:

$$r = 0.6\, r_{\text{rot}} + 0.2\, r_{\text{vel}} + 0.1\, r_{\text{prog}} + 0.1\, r_{\text{toward}} - \lambda \mathcal{P}. \tag{11}$$

The reward components are defined as

$$\begin{aligned} r_{\text{rot}} &= \max(1 - \mathbf{u}^\top R_{\text{target}} \mathbf{u},\ 0), & r_{\text{vel}} &= \exp\!\big(-4(2.5 - v_{\text{along}})^2\big)\,, \\ r_{\text{prog}} &= \text{clamp}\!\left(\frac{D_0 - D_t}{D_0 + \epsilon},\ 0,\ 1\right), & r_{\text{toward}} &= \text{clamp}(\mathbf{d}_{\text{face}} \cdot \mathbf{d}_{\text{target}},\ 0,\ 1)\,. \end{aligned} \tag{12}$$

Here, $\mathbf{u} = [0, 0, 1]^\top$ is the world up vector, R_{target} is the target orientation, and v_{along} is the root velocity projected onto the horizontal direction toward the target. D_0 and D_t denote the initial and current horizontal distances to the target, respectively. $\mathbf{d}_{\text{face}}$ is the character heading direction, and $\mathbf{d}_{\text{target}}$ is the direction from the character to the target in the horizontal plane. Once the target tilt exceeds approximately 78°, the strike reward is set to 1. The energy term is defined as $\mathcal{P} = \sum_i |\tau_i \dot{q}_i|$ over all joints. We set $\lambda = 10^{-5}$ and clip the resulting energy penalty to $[-0.1, 0]$. The target is initialized at a random horizontal distance between $0.5\,\mathrm{m}$ and $10.0\,\mathrm{m}$ from the character.

Pick-and-Place. The pick-and-place task is decomposed into three stages, including pick-up, carry-to, and put-down. Each stage uses a stage-specific reward while sharing the same energy regularization form. The pick-up reward encourages the character to face the box, grasp it with both hands, apply sufficient contact force, and lift it from the source platform:

$$r = 0.4\, r_{\text{lift}} + 0.3\, r_{\text{grasp}} + 0.2\, r_{\text{force}} + 0.1\, r_{\text{face}} - \lambda \cdot \mathcal{P} \tag{13}$$

The reward components are

$$\begin{aligned} r_{\text{lift}} &= \text{clamp}\!\left(\frac{z_{\min} - h_{\text{src}}}{h_{\text{target}} - h_{\text{src}}},\ 0,\ 1\right), \\ r_{\text{grasp}} &= \frac{1}{2}\left(e^{-3d_R} + e^{-3d_L}\right), \\ r_{\text{force}} &= \text{clamp}\!\left(\frac{\min(\|F_R\|, \|F_L\|)}{10},\ 0,\ 1\right) \cdot \mathbf{1}[\text{both hands close}], \\ r_{\text{face}} &= \text{clamp}(\mathbf{d}_{\text{face}} \cdot \mathbf{d}_{\text{box}},\ 0,\ 1)\,. \end{aligned} \tag{14}$$

Here, $z_{\min}$ is the lowest corner height of the box, h_{src} is the source platform surface height, and $h_{\text{target}} = 0.8\,\mathrm{m}$ is the target lifting height. d_R and d_L are the distances from the right and left hands to their nearest grasp points on opposite sides of the box. F_R and F_L are the right and left hand contact forces. The indicator $\mathbf{1}[\text{both hands close}]$ equals 1 only when both hands are within $0.1\,\mathrm{m}$ of their corresponding grasp points. The energy term is computed over all joints with $\lambda = 10^{-5}$, and the resulting energy penalty is clipped to $[-0.1, 0]$. The source platform is initialized at a random horizontal distance between $0.5\,\mathrm{m}$ and $1.5\,\mathrm{m}$, with a surface height between $0.1\,\mathrm{m}$ and $0.5\,\mathrm{m}$.

The carry-to reward maintains the grasp while guiding the character toward the target platform:

$$r = 0.4\, r_{\text{path}} + 0.1\, r_{\text{face}} + 0.2\, r_{\text{grasp}} + 0.15\, r_{\text{force}} + 0.15\, r'_{\text{lift}} - \lambda \cdot \mathcal{P} \tag{15}$$

The path follow and lift-maintenance terms are defined as

$$r_{\text{path}} = \exp\!\big(-2\|p^{xy}_{\text{target}} - p^{xy}_{\text{root}}\|^2\big)\,, \qquad r'_{\text{lift}} = \text{clamp}\!\left(\frac{z_{\min}}{1.0},\ 0,\ 1\right). \tag{16}$$

The grasp and force terms are reused from the pick-up reward. We set $\lambda = 2 \times 10^{-5}$ and clip the resulting energy penalty to $[-0.2, 0]$.

The put-down reward guides the character to place the box on the target platform and release it after stable placement:

$$r = 0.3\, r_{\text{prog}} + 0.2\, r_{\text{place}} + 0.2\, r_{\text{succ}} + 0.1\, r_{\text{grasp}} + 0.1\, r_{\text{force}} + 0.1\, r'_{\text{lift}} - \lambda \mathcal{P}. \tag{17}$$

The progress and placement terms are defined as

$$r_{\text{prog}} = \text{clamp}\left(\frac{D_0 - D_t}{D_0 + \epsilon},\ 0,\ 1\right), \qquad r_{\text{place}} = \exp(-5|z_{\text{box}} - z_{\text{target}}|)\,. \tag{18}$$

Here, r_{prog} rewards horizontal progress toward the target platform, and r_{place} is activated within $0.5\,\text{m}$ of the target platform to encourage vertical alignment. The success term r_{succ} is a binary reward triggered when the box is stably placed on the target platform and both hands have released it. Upon success, all sub-rewards are frozen at their maximum values. We set $\lambda = 2 \times 10^{-5}$ and clip the resulting energy penalty to $[-0.2, 0]$.

E Training Time and Compute

The progressive design of HetSkills allows different skills to be trained independently on top of the shared latent space, instead of jointly optimizing all skills in a single monolithic model. This design keeps each training stage relatively lightweight. After the tracking skill learns the shared motion decoder, later stages only need to learn skill-specific latent mappings or task-guidance modules while reusing the frozen low-level controller.

In our implementation, the tracking skill is trained on two RTX 5090 GPUs for approximately 2 days, and the text-to-motion skill is trained on two RTX 5090 GPUs for approximately 5 days. For motion completion, the three instantiations are trained independently on a single RTX 5090 GPU. Human-scene interaction takes approximately 2 days, VR tracking takes approximately 1 day, and motion in-betweening takes approximately 2 days.

For downstream task adaptation, each body part may be conditioned on multiple language instructions. These instruction branches are evaluated in parallel by batching them together, so increasing the number of instructions does not lead to a proportional increase in wall-clock training time. In our experiments, each downstream task can be trained within 5 hours on a single RTX 5090 GPU.