Unleashing Humanoid Reaching Potential via Real-world-Ready Skill Space

Zhikai Zhang[1,3]* **Chao Chen**[3,6]* **Han Xue**[1,3]* **Jilong Wang**[2,3] **Sikai Liang**[3,7]
Yun Liu[1,3] **Zongzhang Zhang**[6] **He Wang**[2,3] **Li Yi**[1,4,5]
Tsinghua University[1] Peking University[2] Galbot[3] Shanghai AI Laboratory[4]
Shanghai Qi Zhi Institute [5] Nanjing University [6] Tongji University[7]

R^{2}S^2.GITHUB.IO

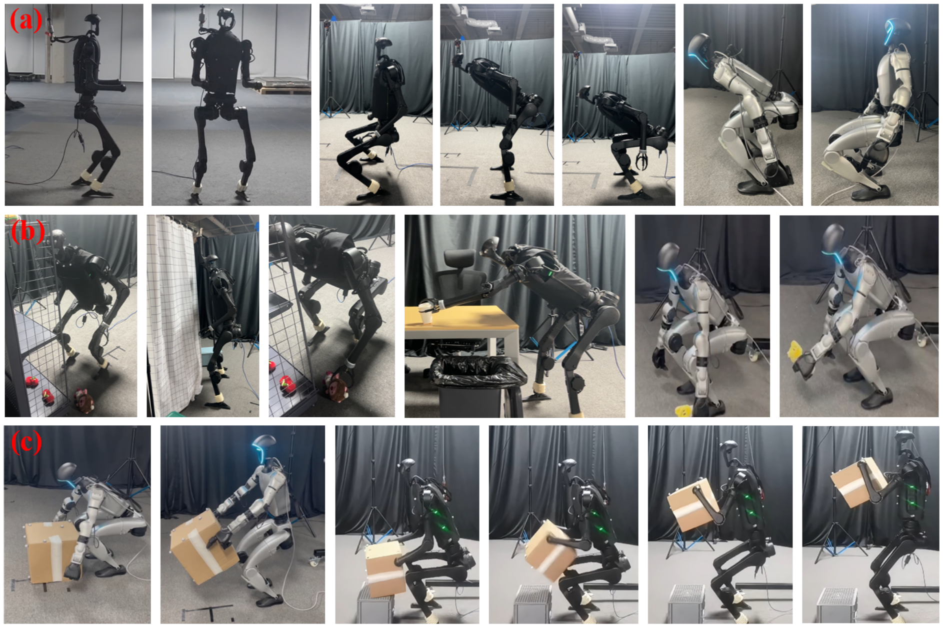

Figure 1: **(a)** The humanoid showcases multiple real-world-ready primitive skills, including locomotion and body-pose-adjustment. **(b)** The humanoid touches points at varying heights and distances. **(c)** The humanoid picks up a box with its hands.

Abstract: Humans possess a large reachable space in the 3D world, enabling interaction with objects at varying heights and distances. However, realizing such large-space reaching on humanoids is a complex whole-body control problem and requires the robot to master diverse skills simultaneously—including base positioning and reorientation, height and body posture adjustments, and end-effector pose control. Learning from scratch often leads to optimization difficulty and poor sim2real transferability. To address this challenge, we propose Real-world-Ready Skill Space (R^{2}S^2). Our approach begins with a carefully designed skill library consisting of real-world-ready primitive skills. We ensure optimal performance and robust sim2real transfer through individual skill tuning and sim2real evaluation. These skills are then ensembled into a unified latent space, serving as a structured prior that helps task execution in an efficient and sim2real transferable manner. A high-level planner, trained to sample skills from this space, enables the robot to accomplish real-world goal-reaching tasks. We demonstrate zero-shot sim2real transfer and validate R^{2}S^2 in multiple challenging goal-reaching scenarios, including point touch and box pickup as shown in Figure 1.

Keywords: Humanoid Whole-Body Control, Representations for Robotic Control, Sim-to-Real Transfer

1 Introduction

The bipedal structure of humans provides a significantly larger reachable workspace than quadrupedal systems, enabling interactions with objects at varying heights and distances—from overhead shelves to floor-level items. For humanoid robots to effectively assist humans in daily tasks, they should achieve similar workspace [1]. However, this presents a complex whole-body control challenge and requires mastering diverse skills (base positioning and reorientation, height and body posture adjustments, and hand pose control) within a dynamically unstable system [2].

Traditional model-based control methods [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] struggle with the inherent imperfections in system modeling and environmental disturbances. Recent end-to-end reinforcement learning has achieved great progress in humanoid whole-body control tasks [14, 15, 16, 17, 18, 19, 20, 21, 22]. Can we utilize it to endow humanoids the capability to accomplish tasks requiring a human-level large reachable space? We found that optimization difficulty and sim2real instability for such a complex whole-body control (WBC) problem are major concerns. As mentioned before, multiple skills are required for unleashing humanoid reaching potential. Learning from scratch is difficult. Also, a stable real-world performance usually requires imposing skill-specific constraints during training. However, it is very challenging to impose sufficient behavior constraints in an end-to-end RL manner under this multi-skill setting, since different skills may desire different constraints and conflict each other.

In this work, we propose Real-world-Ready Skill Space (R^2S^2), aiming at constructing a skill space that encompasses and encodes various real-world-ready motor skills. The learned space can serve as a structured skill prior and helps task execution in an efficient and sim2real transferable manner.

Concretely, we first construct a library of shared and easy-to-define primitive skills. In this work, there are locomotion, body-pose-adjustment (changing body height, bending over) and hand-reaching for goal-reaching tasks. We ensure optimal performance and robust sim2real transfer through individual skill tuning and sim2real evaluation. Though separated primitive skills are real-world-ready, they are insufficient to serve as a practical skill space for two reasons: 1) due to separated training of primitive skills, the coordination (e.g., upper-body reaching an object while lower-body squatting) and transition (e.g., lower-body from locomotion to body-pose-adjustment) between different skills are out-of-distribution, making the skill space incomplete for practical applications; 2) separated skills often have mismatched task spaces, lacking a unified representation for high-level planners. Therefore, we introduce an additional stage to ensemble different skills (not only learn skills themselves, but also their coordination and transition) and encode skill representations. We achieve this by combining Imitation Learning (IL) and Reinforcement Learning (RL) to train a student policy with a variational information bottleneck, with the skills in the pre-constructed library as teacher policies. With imitation learning, the student policy can inherit real-world-ready skill priors from teacher policies. Reinforcement learning further enhances the student policy the coordination and transition skills that are incapable for teacher policies. The student network is designed as an encoder-decoder architecture with a variational information bottleneck to model the motor skill distribution conditioned on proprioception. Finally, we train task-specific high-level planning policies to sample latent codes from this real-world-ready skill space to stably achieve real-world goal-reaching task completion.

We achieve sim2real transfer on Unitree G1 (1.27 meters tall) and a more challenging full-sized humanoid Unitree H1 (1.8 meters tall). With our R^2S^2, the robot can stably unleash its reaching potential in two practical daily applications: point touch and box pickup. Extensive experiments are conducted to evaluate the effectiveness of our major designs.

2 Related Works

Humanoid Robot Learning Reinforcement learning (RL) policies have achieved great progress in recent humanoid robot learning. Researches on locomotion [23, 24, 25, 26, 27, 28, 29, 30, 31] aim at provide bipedal humanoids with the ability to traverse different terrains in a stable and agile

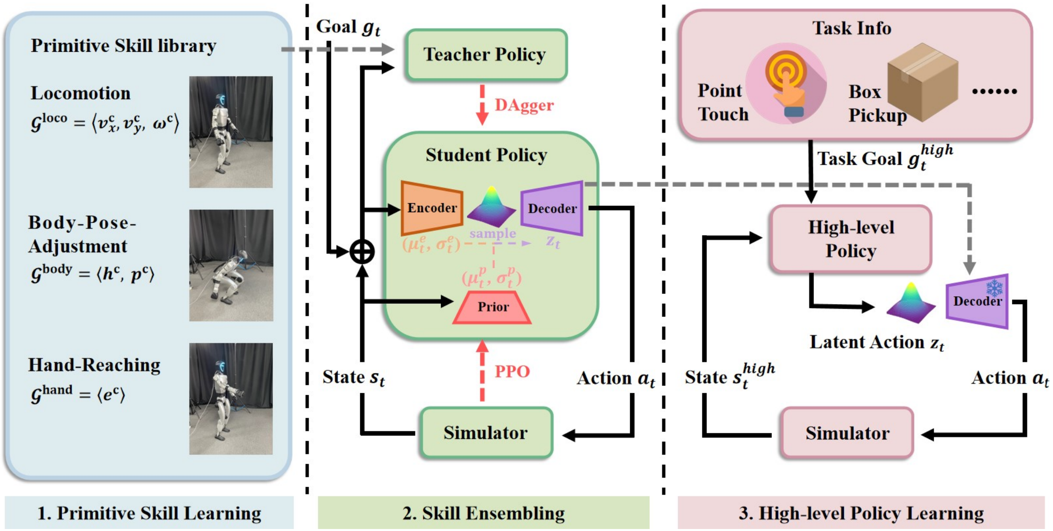

Figure 2: We present Real-world-Ready Skill Space (R^{2}S^2), a skill space that encompasses and encodes various real-world-ready motor skills. 1) We decompose the complex WBC motor skills into a library of primitive skills, each separately tuned and sim2real evaluated. 2) We ensemble multiple primitive skills into a student policy with a variational information bottleneck. 3) We train high-level planning policies to sample from R^{2}S^2 to efficiently and stably accomplish real-world goal-reaching tasks.

manner. But these works often focus only on the lower body of humanoids and ignore their whole-body reaching and interaction potential. Learning-based humanoid whole-body-control [14, 15, 16, 17, 19, 20, 32, 33, 34, 35] recently demonstrate new capabilities and push the boundaries of humanoid robots. Data-driven motion tracking methods [14, 15, 32, 19, 20, 33, 34] mimic human motion expressively, allowing for human-to-humanoid teleoperation. Zhang et al. [16] formulate humanoid whole-body-control as sequential contacts and propose a contact-based WBC framework. However, existing works either act in a relatively neutral body pose [16, 17] or lack a planning module for practical task accomplishment [35]. How to equip humanoid robots with a human-level goal-reaching capability remains underexplored.

Skill Space Learning In physics-based character animation, skill spaces [36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46] are often learned to reuse motion priors from motion capture datasets. Motion imitation [37, 39, 40, 41, 42, 46, 47] or adversarial learning [36, 43, 44, 45] is used to form a skill latent space, and then the sampled latent variable can be translated into actions through a decoder. For high-level tasks, task-specific planners are learned to reuse skills from the pre-constructed latent space for more efficient and natural task accomplishment. Though skill space learning has achieved great success in physics-based character animation [38, 48, 49], transferring such a paradigm to real-world humanoid is challenging because of lack of high-quality humanoid motion datasets and sim2real difficulty. Contrast to these works, we learn a skill space from sim2real-evaluated (real-world-ready) primitive skills rather than motion capture data for real-world stability.

3 Unleashing Humanoid Reaching Potential via Real-world-Ready Skill Space

In this section, we describe how to learn a Real-world-Ready Skill Space (R^{2}S^2) and utilize it to support practical goal-reaching tasks in a stable and sim2real transferable manner. First, we construct a library of n shared and easy-to-define RL-based primitive skills $\{\pi^{\text{prim}}_1, \ldots, \pi^{\text{prim}}_n\}$, with each skill π^{prim}_i separately tuned and sim2real evaluated (real-world ready). Then we will ensemble and encode these skills via combining IL and RL into an ensemble student policy π^{ensem} with a latent skill z space. The learned skill space encompasses various real-world-ready motor skills, serving as a skill prior and helping task execution in a sim2real transferable manner. With the learned skill z space, high-level planners π^{plan} are trained to sample latent skills for practical task accomplishment. The pipeline is shown in Figure 2. We use PPO [50] for all of our policy training, domain randomization for sim2real transfer and Isaac Gym [51] for simulation. For more details of our training, please refer to the appendix.

3.1 Primitive Skill Repository

Aiming at unleashing humanoid reaching potential, we design the primitive skill library $\{\pi_1^{\text{prim}}, \dots, \pi_n^{\text{prim}}\}$ as locomotion, body-pose-adjustment (changing body height, bending over) and hand-reaching. Each is separately tuned and sim2real evaluated to maximize their capability and real-world stability.

Our primitive skills can be seen as goal-conditioned RL policies $\pi^{\text{prim}} : \mathcal{G}^{\text{prim}} \times \mathcal{S}^{\text{prim}} \mapsto \mathcal{A}^{\text{prim}}$, where $\mathcal{G}^{\text{prim}}$ includes goal commands g_t specifying skill target. $\mathcal{S}^{\text{prim}}$ includes the robot's proprioceptive observation and history action information $s_t = [\omega_t, gr_t, q_t, \dot{q}_t, a_{t-1}]$ at each timestep t, where $\omega_t, gr_t, q_t, \dot{q}_t, a_{t-1}$ are angular velocity in the base frame, projected gravity, body-part dof positions, body-part dof velocities, and last-frame low-level action, respectively. It is worth noting that for $q_t, \dot{q}_t, a_{t-1}$, each policy only takes relevant body part information as observation for different skills. $\mathcal{A}^{\text{prim}}$ includes the robot body-part action (PD targets) a^{prim}, which is fed into a PD controller for torque computation. a^{prim} only controls corresponding body part for each skill, and other joints are fixed. Their training reward can be written as:

$$r_{\text{prim}} = r_{\text{command}} + r_{\text{behavior}} + r_{\text{regularization}}, \tag{1}$$

where r_{task} represents skill command tracking objectives, r_{behavior} depicts skill-specific behavior constraints for sim2real stability, and $r_{\text{regularization}}$ is skill-agnostic regularization. In the following sections, we mainly introduce r_{behavior} of each skill since they are most important for sim2real transfer of each skill. For more details of our reward design, please refer to our appendix.

Locomotion For locomotion, $\mathcal{G}^{\text{loco}} = \langle v_x^{\text{c}}, v_y^{\text{c}}, \omega^{\text{c}} \rangle$ actuates the humanoid to track desired linear and angular velocities of the robot base in the robot base frame. To constrain locomotion behavior and replicate human-like bipedal gaits, we model each foot's motion as an alternating sequence of swing and stance phases and introduce a periodic reward framework inspired by [31, 52]:

$$r_{\text{behavior}}^{\text{loco}} = r_{\text{gait\_velocity}} + r_{\text{gait\_force}}, \tag{2}$$

$$r_{\text{gait\_velocity}} = \sum_{\text{foot}} [1 - \text{C}_{\text{foot}}(t)] |v_{\text{foot}}|^2, r_{\text{gait\_force}} = \sum_{\text{foot}} [\text{C}_{\text{foot}}(t)] |\text{f}_{\text{foot}}|^2, \tag{3}$$

where $C_{\text{foot}}(t)$ follows Von Mises distributions and $t \in [0, 1)$ is a time-dependent phase variable cycling periodically through normalized time.

Body-Pose-Adjustment For body-pose-adjustment, $\mathcal{G}^{\text{body}} = \langle h^{\text{c}}, p^{\text{c}} \rangle$ tracks the base height and pitch angle in the global frame. We found that for such a skill, kinematic and dynamic symmetry is important for real-world stability, so we introduce:

$$r_{\text{behavior}}^{\text{body}} = r_{\text{base\_roll}} + r_{\text{leg\_pos}} + r_{\text{leg\_torque}} + r_{\text{touch\_ground}}, \tag{4}$$

where $r_{\text{base\_roll}}$ and $r_{\text{leg\_pos}}$ are designed for kinematic symmetry. $r_{\text{base\_roll}}$ punishes robot base roll rotation in prevent of humanoid tip over. $r_{\text{leg\_pos}}$ punishes asymmetric dof pos of paired joints in legs. $r_{\text{leg\_torque}}$ and $r_{\text{touch\_ground}}$ are designed for dynamic balance. $r_{\text{leg\_torque}}$ encourages paired lower-body joint to output the same torques. $r_{\text{touch\_ground}}$ encourages both feet to be in contact with ground.

Hand-Reaching For hand reaching, $\mathcal{G}^{\text{hand}} = \langle e^{\text{c}} \rangle$ tracks the target end-effector 6D pose in the robot local frame. Arms are relatively easy for sim2real deployment, so we do not specifically design any r_{behavior} for this skill.

3.2 Real-world-Ready Skill Space

Given real-world-ready primitive skills $\{\pi_1^{\text{prim}}, \dots, \pi_n^{\text{prim}}\}$, a straight attempt to reuse these primitive skills for different tasks is directly planning in their primary task spaces. But they are actually insufficient to serve as a practical skill space. Because of separated training, primitive skills are unseen to each other. The coordination (e.g., upper-body reaching an object while lower-body squatting) and transition (e.g., lower-body from locomotion to body-pose-adjustment) between different skills are out-of-distribution problems. Naïvely concatenating actions of different body parts or switching from locomotion to body-pose-adjustment skill will lead to instability or even cause robot to fall. Without seamless coordination and transition, the skill space is incomplete for practical task accomplishment. In addition, the mismatched task spaces ($v_x^{\text{c}}, v_y^{\text{c}}, \omega^{\text{c}}$ for locomotion, $h^{\text{c}}, p^{\text{c}}$ for body-pose-adjustment, and e^{c} for hand-reaching in our setting) of primitive skills are inefficient for high-level planning. We will demonstrate this in Section 4.2.

To solve these problems, we propose to train a ensemble student policy $\pi^{\text{ensem}}(a_t|s_t, g_t)$ with a variational information bottleneck to ensemble different skills. "Ensemble" means not only imitating different primitive skills, but also learning their coordination and transition. During skill ensembling, different skills are encoded into a latent skill z space, and then decoded into per-joint actions.

Skill Ensembling via Imitation Learning and Reinforcement Learning Online imitation learning methods (e.g., DAgger [53]) are commonly used for skill distillation from teacher policies to student policies. However, relying only on imitation learning can not provide student policies with new capabilities (e.g., coordination and transition between different skills) beyond teacher policies. Thus, we propose to combine imitation learning and reinforcement learning by adding IL loss and RL loss together. The IL, which is DAgger in our setting, distills real-world-ready skill priors from multiple teacher policies. Based on this, RL, which is PPO in our setting, further encourages the policies to learn new behaviors for seamless transition and coordination. Unlike training separated primitive skills, we modify our training environments in two aspects: 1) we simultaneously send goal commands for different body parts (e.g., the policy needs to track target hand 6D pose while walking at the same time) to model skill coordination; 2) we allow the skill of a certain body part to transition from one to another in an episode to model skill transition. Formally, at each timestep t, two primitive skills $\{\pi_t^{\text{lower}}, \pi_t^{\text{upper}}\}, \pi_t^{\text{lower}} \in \{\pi^{\text{loco}}, \pi^{\text{body}}\}$ and $\pi_t^{\text{upper}} \in \{\pi^{\text{hand}}\}$, serve as teacher policies for different body parts, one for lower-body and the other one for upper-body. A skill indicator is included in student policy goal g_t to indicate which teacher policy is activated. When transition happens, we let $\pi_{t+1}^{\text{lower}} \neq \pi_t^{\text{lower}}$. By doing so, all possible coordination and transition between different skills are included in the student policy. The reward function can be written as:

$$\mathcal{L}_{\text{Ensem}} = \lambda_1 \mathcal{L}_{\text{DAgger}} + \lambda_2 \mathcal{L}_{\text{PPO}}, \tag{5}$$

where λ_1 decreases from 0.95 to 0.05 gradually and λ_2 inversely adjusted. This design encourages the student policy to mimic teacher policies first and exploring new behaviors latter. We let

$$\mathcal{L}_{\text{DAgger}} = \mathbb{E}_{(s,a^*)\sim\mathcal{D}_{\text{agg}}} \left[\| a_t^{\text{ensem}} - a_t^* \|^2 \right], \tag{6}$$

where a_t^{ensem} is the output action of π^{ensem} and $a_t^* = concat(a_t^{\text{lower}}, a_t^{\text{upper}})$ is the combination of actions from lower-body and upper-body teacher policies. For $\mathcal{L}_{\text{PPO}}$, we simply combine the reward terms of π_t^{lower} and π_t^{upper} defined in the primitive skill training stage. We found that the student policy can successfully learn coordination and transition skills without any additional reward terms. Though coordination and transition are new learned at this stage, the skill prior inherited from teacher policies serves as a good warm-up and makes the new capabilities sim2real transferable.

Learning a Real-World-Ready Skill Space as Latent Representation While the student policy can ensemble multiple primitive skills, mismatched skill spaces hinders efficient high-level planning due to the absence of a unified skill representation. To mitigate this, we adopt an encoder-decoder framework with a conditional variational information bottleneck. We employ a variational encoder $\mathcal{E}(z_t|s_t, g_t) = \mathcal{N}(z_t; \mu^e(s_t, g_t), \sigma^e(s_t, g_t))$ to model latent codes conditioned on current state and goal. A corresponding decoder $\mathcal{D}(a_t|s_t, z_t)$ maps the sampled latent code to action conditioned on state. Inspired by [40], we introduce a learnable conditional prior $\mathcal{P}(z_t|s_t) = \mathcal{N}(z_t; \mu^p(s_t), \sigma^p(s_t))$ to capture state-based action distribution instead of assuming a fixed unimodal Gaussian structure over the latent space, since the robot action distribution should be significantly different given different states. So the ensemble student policy can be formulated as $\pi^{\text{ensem}} \triangleq (\mathcal{E}, \mathcal{D}, \mathcal{P})$. The total loss in training π^{ensem} can be written as:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Ensem}} + \lambda_3 \mathcal{L}_{\text{Regu}} + \lambda_4 \mathcal{L}_{\text{KL}}, \tag{7}$$

where

$$\mathcal{L}_{\text{Regu}} = \|\mu^e(s_t, g_t) - \mu^e(s_{t+1}, g_{t+1})\| \tag{8}$$

encourages temporal consistency between consecutive latent codes and makes the skill space more structured. $\mathcal{L}_{\text{KL}} = D_{\text{KL}}(\mathcal{E}(z_t|s_t, g_t) \parallel \mathcal{P}(z_t|s_t))$ encourages the distribution of the latent code to be close to the learnable prior.

3.3 High-Level Planning in Real-World-Ready Skill Space

Building on the learned latent skill space, we train task-specific high-level planners $\pi^{\text{plan}}(z_t|s_t^{\text{high}}, g_t^{\text{high}})$ to select latent skill embeddings for different goal reaching tasks. The action for π^{plan} is now in the latent z_t space. The sampled z_t is decoded into per-joint actions a_t via frozen decoder $\mathcal{D}$. The training reward can be written as:

$$r_{\text{plan}} = r_{\text{task}} + r_{\text{regularization}}, \tag{9}$$

where r_{task} is task execution objective and $r_{\text{regularization}}$ is the skill-agnostic regularization reward introduced at skill library construction stage. We found that reusing $r_{\text{regularization}}$ can enhance motion stability. For more details of our high-level planner training, please refer to our appendix.

4 Experiments

In this section, comprehensive experiments in both simulation and real-world will be conducted to answer the following questions: **Q1.** (Section 4.1) Can Real-world-Ready Skill Space (R^{2}S^2) unleash humanoid reaching potential? **Q2.** (Section 4.2) How does R^{2}S^2 help to unleash humanoid reaching potential? We conduct all of our experiments on Unitree H1.

4.1 Humanoid Reachable Space

In this part, we want to compare our method with existing works to evaluate whether R^{2}S^2 unleashes humanoid reaching potential significantly.

4.1.1 Experiment Setting

We define the humanoid reaching problem as: given a target *reaching state* $[xy\omega_{\text{root}}, xyz_{\text{hand}}]$, whether the humanoid can successfully achieve it. $xy\omega_{\text{root}}$ is the horizontal position and orientation of the robot root. xyz_{hand} is the 3D position that a hand can reach when $xy\omega_{\text{root}}$ is fixed. Since most existing works are equipped with omnidirectional locomotion capability and can already satisfy any $xy\omega_{\text{root}}$ on a plane ground, we mainly compare the capability of achieving any given xyz_{hand}.

4.1.2 Experiment Metrics

We use the covered space of reachable xyz_{hand} to measure the humanoid reaching capability. We roughly estimate the reachable space by calculating the volume swept by a sphere centered at one of the shoulder joint with the arm's length as the radius. While such computation is not perfectly precise, it enables convenient estimation of reachable xyz_{hand} space given body pose parameters.

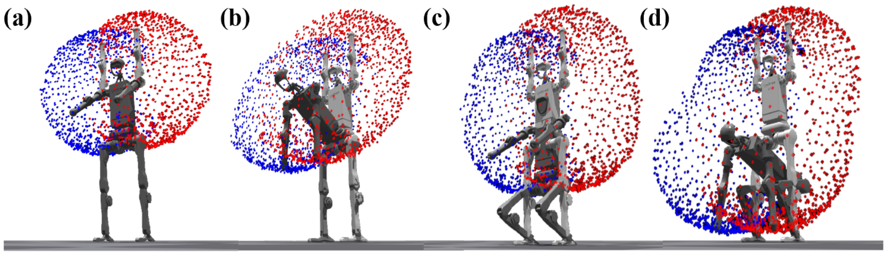

Figure 3: Red points and blue points represent the reachable space of the left and right hand respectively. **(a)** is the reachable space of neutral humanoid standing posture. **(b)** is the reachable space with only bend-over skill. **(c)** is the reachable space with only change-body-height skill. **(d)** is the reachable space with our full body-pose-adjustment skill.

4.1.3 Baselines

How to unleash humanoid reaching potential is underexplored. There are few relevant works. We mainly compare our method with two recent works focusing on whole-body-control on the same hardware (Unitree H1) as our experiments:

- **ExBody** [14]. This work decouples motion goal into movement goal and expressive goal, with the movement goal including commands for robot base such as velocity, roll, pitch, and base height. We reproduce this method in simulation and deploy it on our hardware.
- **HUGWBC** [28]. This work proposes a unified whole-body controller, generating versatile locomotion and adjusting body postures. Due to the unavailability of code, we calculate the reachable space according to the posture adjustment parameters in their original paper.
- **Our method w/o** h^c, p^c. We ablate on the task space of our primitive skills to evaluate how each part contribute to the reachable space.

4.1.4 Experiment Results

The result is shown in Table 1. Compared with existing methods, our method improves reachable space significantly. Control of body height h^c and pitch angle p^c can both contribute to the reachable space. We also found that ExBody generates physical artifacts (e.g., jerking, foot sliding) in simulation and fails to transfer to real-world, mainly because of lacking in skill-specific tuning for sim2real transfer. We also visualize the reachable space in Figure 3.

Method	*Reachable Space* $(m^3)\uparrow$	
	Sim	Real
ExBody [14]	1.707	✗
HUGWBC [28]	2.094	2.094
Ours w/o h^c, p^c	1.150	1.150
Ours w/o h^c	2.127	2.127
Ours w/o p^c	1.814	1.814
Ours	**3.173**	**3.173**

Table 1: Compared with baseline methods, our method improves reachable space significantly.

4.2 Evaluation of Real-world-Ready Skill Space

In this part, we want to figure out how our proposed real-world-ready skill space helps our goal-reaching tasks and evaluate the effectiveness of each of our major design.

4.2.1 Experiment Setting

In our experiment setting, we evaluate how each design of R^2S^2 helps the accomplishment of two goal-reaching tasks: point touch and box pickup. For point touch, we randomly set one point within a $2m \times 2m$ square in front of the robot, with height ranging from 0.1 meters to 2.0 meters. The humanoid is asked to touch the point with a specific hand. For box pickup, the box is randomly placed within a $2m \times 2m$ square in front of the robot, with height ranging from 0.2 meters to 1.2 meters. The humanoid is asked to lift the box to a height of 1.4 meters.

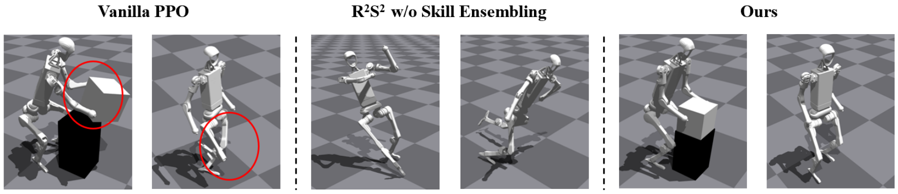

Figure 4: We show the motor behaviors under different settings. Our method significantly helps sim2real transfer with real-world-ready skill space.

4.2.2 Experiment Metrics

We use *Success Rate* and *Distance Error* as metrics. For point touch, *Success Rate* records the percentage of trials that humanoids successfully touch the target point within 5 cm. *Distance Error* is the closest distance between the humanoid's end effector and the target point in a touch. For box pickup, *Success Rate* here means the percentage of trials that humanoids successfully lift the box above 1.3 meters. For *Distance Error*, we calculate the closest distance between the box and 1.4 meters height. We conduct experiments in simulation and all metrics are averaged over 10000 trials. Beyond task accomplishment results in simulation, we also evaluate whether the motor behavior learned during task execution is sim2real transferable.

4.2.3 Baselines

We ablate on different components in our R^2S^2 and choose the following baselines:

- **Vanilla PPO**. We implement a vanilla PPO, trying to accomplish each goal-reaching task from scratch without any skill prior.
- **R^2S^2 w/o SE (Skill Ensembling)**. We use separate primitive skills to serve as the skill space. In this setting, a high-level planner policy is trained to directly output skill indicator and the command in primary task space. We adopt this baseline to mainly verify the importance of coordination and transition capability.
- **R^2S^2 w/o LS (Latent Space)**. We implement an MLP-based student policy to ensemble skills from multiple teacher policies. In this setting, though the primitive skills are ensembled (i.e., coordination and transition are learned), the high-level planning policy still needs to output skill indicator and the command in primary mismatched task space for task execution. We adopt this baseline to evaluate the effectiveness of our latent skill space.

4.2.4 Experiment Results

We report the resutls in Table 2. We found that a vanilla PPO can hardly learn to accomplish such goal-reaching tasks from scratch. The learned behavior also tends to be unstable and thus not sim2real transferable as shown in Figure 4. R^2S^2 w/o Skill Embedding performs unsafely due to the lack of coordination and transition capability. When the skill indicator generated by high-level planners changes and the robot transitions from one lower-body skill to another. It can sometimes fall as shown in Figure 4, thus it is also not sim2real transferable. For R^2S^2 w/o Latent Space, we found that generating commands from the primary task space performs significantly worse than the latent space. We suspect it is mainly because the latent skill representation is more compact and structured than raw combination of skill indicator and commands.

Method	Point touch		Box pickup		Sim2Real
	SR ↑	*DE* ↓	*SR* ↑	*DE* ↓	
Vanilla PPO	11.6	0.49	0.0	0.65	✗
R^2S^2 w/o SE	30.5	0.15	22.8	0.33	✗
R^2S^2 w/o LS	56.9	0.10	43.3	0.19	✓
Ours	**100**	**0.03**	**100**	**0.04**	✓

Table 2: We evaluate the effectiveness of our major designs. "SR" is short for *Success Rate* and "DE" is short for *Distance Error*.

5 Limitations

In this work, we propose a skill space that encompasses and encodes various real-world-ready motor skills. The space can serve as a structured skill prior and help task execution in an efficient and sim2real transferable manner. Though we believe our method can be extended to more general and complex whole-body control systems, currently its limition is obvious. 1) The number of primitive skills is now limited, incorporating more skills may bring new challenges to the skill ensembling and skill space learning. 2) Though we achieve seamless coordination and transition at skill ensembling stage, how to blend spatially overlapping skills (e.g., making the humanoid change body pose while locomotion) is still challenging. 3) For now we rely on motion capture system to understand the relationship between the robot and the interaction target. Incorporating a visual module is necessary for more general scenarios.

References

[1] Z. Gu, J. Li, W. Shen, W. Yu, Z. Xie, S. McCrory, X. Cheng, A. Shamsah, R. Griffin, C. K. Liu, et al. Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning. *arXiv preprint arXiv:2501.02116*, 2025.

[2] Q. Liao, B. Zhang, X. Huang, X. Huang, Z. Li, and K. Sreenath. Berkeley humanoid: A research platform for learning-based control. *arXiv preprint arXiv:2407.21781*, 2024.

[3] K. Hauser, T. Bretl, and J.-C. Latombe. Non-gaited humanoid locomotion planning. In *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pages 7–12. IEEE, 2005.

[4] M. Posa, C. Cantu, and R. Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.

[5] Y.-C. Lin, B. Ponton, L. Righetti, and D. Berenson. Efficient humanoid contact planning using learned centroidal dynamics prediction. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5280–5286. IEEE, 2019.

[6] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter. Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators. *IEEE Robotics and Automation Letters*, 7(2):2377–2384, 2022.

[7] J. Carpentier and N. Mansard. Multicontact locomotion of legged robots. *IEEE Transactions on Robotics*, 34(6):1441–1460, 2018.

[8] X. Xinjilefu, S. Feng, and C. G. Atkeson. Dynamic state estimation using quadratic programming. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 989–994. IEEE, 2014.

[9] G. Schultz and K. Mombaur. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on mechatronics*, 15(5):783–792, 2009.

[10] P. M. Wensing and D. E. Orin. Improved computation of the humanoid centroidal dynamics and application for whole-body control. *International Journal of Humanoid Robotics*, 13(01): 1550039, 2016.

[11] H. Dai, A. Valenzuela, and R. Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 295–302. IEEE, 2014.

[12] D. E. Orin, A. Goswami, and S.-H. Lee. Centroidal dynamics of a humanoid robot. *Autonomous robots*, 35:161–176, 2013.

[13] K. Harada, S. Kajita, H. Saito, M. Morisawa, F. Kanehiro, K. Fujiwara, K. Kaneko, and H. Hirukawa. A humanoid robot carrying a heavy object. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1712–1717. IEEE, 2005.

[14] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[15] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

[16] C. Zhang, W. Xiao, T. He, and G. Shi. Wococo: Learning whole-body humanoid control with sequential contacts. *arXiv preprint arXiv:2406.06005*, 2024.

[17] J. Dao, H. Duan, and A. Fern. Sim-to-real learning for humanoid box loco-manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16930–16936. IEEE, 2024.

[18] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. *arXiv preprint arXiv:2412.07773*, 2024.

[19] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[20] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[21] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025.

[22] M. Liu, Z. Chen, X. Cheng, Y. Ji, R.-Z. Qiu, R. Yang, and X. Wang. Visual whole-body control for legged loco-manipulation. *arXiv preprint arXiv:2403.16967*, 2024.

[23] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen. Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning. *arXiv preprint arXiv:2408.14472*, 2024.

[24] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[25] H. Duan, B. Pandit, M. S. Gadde, B. Van Marum, J. Dao, C. Kim, and A. Fern. Learning vision-based bipedal locomotion for challenging terrain. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 56–62. IEEE, 2024.

[26] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *arXiv preprint arXiv:2401.16889*, 2024.

[27] H. Wang, Z. Wang, J. Ren, Q. Ben, T. Huang, W. Zhang, and J. Pang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. *arXiv preprint arXiv:2502.10363*, 2025.

[28] Y. Xue, W. Dong, M. Liu, W. Zhang, and J. Pang. A unified and general humanoid whole-body controller for fine-grained locomotion. *arXiv preprint arXiv:2502.03206*, 2025.

[29] A. Kumar, Z. Li, J. Zeng, D. Pathak, K. Sreenath, and J. Malik. Adapting rapid motor adaptation for bipedal robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1161–1168. IEEE, 2022.

[30] Z. Xie, P. Gergondet, F. Kanehiro, et al. Learning bipedal walking for humanoids with current feedback. *IEEE Access*, 11:82013–82023, 2023.

[31] J. Siekmann, Y. Godse, A. Fern, and J. Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7309–7315. IEEE, 2021.

[32] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024.

[33] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024.

[34] Y. Liu, B. Yang, L. Zhong, H. Wang, and L. Yi. Mimicking-bench: A benchmark for generalizable humanoid-scene interaction learning via human mimicking. *arXiv preprint arXiv:2412.17730*, 2024.

[35] Z. Zhuang, S. Yao, and H. Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024.

[36] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

[37] J. Won, D. Gopinath, and J. Hodgins. Physics-based character controllers using conditional vaes. *ACM Transactions on Graphics*, 41(4):1–12, 2022.

[38] H. Yao, Z. Song, B. Chen, and L. Liu. Controlvae: Model-based learning of generative controllers for physics-based characters. *ACM Transactions on Graphics*, 41(6):1–16, 2022.

[39] Z. Zhang, Y. Li, H. Huang, M. Lin, and L. Yi. Freemotion: Mocap-free human motion synthesis with multimodal large language models. In *European Conference on Computer Vision*, pages 403–421. Springer, 2024.

[40] Z. Luo, J. Cao, J. Merel, A. Winkler, J. Huang, K. Kitani, and W. Xu. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582*, 2023.

[41] Z. Luo, J. Cao, S. Christen, A. Winkler, K. Kitani, and W. Xu. Grasping diverse objects with simulated humanoids. *arXiv preprint arXiv:2407.11385*, 2024.

[42] J. Ren, M. Zhang, C. Yu, X. Ma, L. Pan, and Z. Liu. Insactor: Instruction-driven physics-based characters. *Advances in Neural Information Processing Systems*, 36:59911–59923, 2023.

[43] J. Juravsky, Y. Guo, S. Fidler, and X. B. Peng. Padl: Language-directed physics-based character control. 2022.

[44] C. Tessler, Y. Kasten, Y. Guo, S. Mannor, G. Chechik, and X. B. Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023.

[45] J. Juravsky, Y. Guo, S. Fidler, and X. B. Peng. Superpadl: Scaling language-directed physics-based control with progressive supervised distillation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[46] H. Yao, Z. Song, Y. Zhou, T. Ao, B. Chen, and L. Liu. Moconvq: Unified physics-based motion control via scalable discrete representations. *ACM Transactions on Graphics (TOG)*, 43(4):1–21, 2024.

[47] J. Durán-Hernández, R. Q. Fuentes-Aguilar, L. Campos-Macías, and O. Carbajal-Espinosa. Control implementation in a low-cost designed biped robot to reproduce squats. In *2022 10th International Conference on Control, Mechatronics and Automation (ICCMA)*, pages 36–41. IEEE, 2022.

[48] A. Serifi, R. Grandia, E. Knoop, M. Gross, and M. Bächer. Vmp: Versatile motion priors for robustly tracking motion on physical characters. In *Computer Graphics Forum*, volume 43, pages i–ix, 2024.

[49] A. Tirinzoni, A. Touati, J. Farebrother, M. Guzek, A. Kanervisto, Y. Xu, A. Lazaric, and M. Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. *arXiv preprint arXiv:2504.11054*, 2025.

[50] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[51] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[52] G. B. Margolis and P. Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[53] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

Appendix

Please refer to our supplementary videos for real-world performance.

A Training Details

A.1 Policy Observation

During training, we utilize an asymmetric actor-critic strategy, where the critic has access to privileged observations, including additional robot states and environment extrinsics, for better value prediction. Such privileged information is invisible to the actor policy since it is unavailable in real world. The observations for different policies are shown in Table 3.

π^{prim}	π^{ensem}	π^{plan}
Proprioception	*Proprioception*	*Proprioception*
Angular velocity	Angular velocity	Angular velocity
Projected gravity	Projected gravity	Projected gravity
Body-part dof pos	Whole-body dof pos	Whole-body dof pos
Body-part dof vel	Whole-body dof vel	Whole-body dof vel
Last body-part action	Last whole-body action	Last whole-body action
		Last latent action
Privileged Info	*Privileged Info*	*Privileged Info*
Linear velocity	Linear velocity	Linear velocity
Friction coefficient	Friction coefficient	Friction coefficient
Mass parameter	Mass parameter	Mass parameter
Motor strength	Motor strength	Motor strength
Foot velocity	Foot velocity	Foot velocity
Foot force	Foot force	Foot force
Foot height	Foot height	Foot height
Command Goals	*Command Goals*	*Task Goals*
v_x^c, v_y^c, ω^c (for loco)	$v_x^c, v_y^c, \omega^c, h^c, p^c, e^c$	Point position
h^c, p^c (for body)	Skill indicator	Box pose
e^c (for hand)		

Table 3: We list observations for different policies here.

A.2 Domain Randomization

We introduce randomness to various parameters during training to make the model more robust to real-world scenarios. Table 4 lists the domain randomization used in our experiments, including randomization of observation parameters, dynamics parameters, and external disturbance.

A.3 Network Design

We use MLPs for all of our actors and critics except for π^{ensem}. The actor of π^{ensem} is an encoder-decoder architecture with a variational information bottleneck. The latent dim is 8 in our setting. The architecture parameters of our networks are shown in Table 5.

A.4 Reward Design

We list all of our reward terms and corresponding weights in Table 6.

A.5 Training Hyperparameters

We list all of our PPO training hyperparameters in Table 7.

Term	Value
Observation	
Angular velocity noise	$\mathcal{U}(-0.2, 0.2)$ rad/s
Projected gravity noise	$\mathcal{U}(-0.05, 0.05)$
Dof pos noise	$\mathcal{U}(-0.01, 0.01)$ rad
Dof vel noise	$\mathcal{U}(-1.5, 1.5)$ rad/s
Proprioception latency	$\mathcal{U}(0.005, 0.045)$ s
Dynamics	
Friction factor	$\mathcal{U}(0.4, 1.1)$
Base mass	$\mathcal{U}(-1.0, 5.0)$ kg
Base center of mass	$\mathcal{U}(-0.1, 0.1)^3$ m
Motor strength	$\mathcal{U}(-0.8, 1.2)$
Joint position biases	$\mathcal{U}(-0.08, 0.08)$ rad
External	
Push interval	10.0 s
Push velocity	$\mathcal{U}(0.0, 1.0)$ m/s

Table 4: We randomize various parameters during training to make the model more robust to real-world scenarios.

MLP	Hidden Layers	3
	Hidden Units	512
	Activation	ELU
Encoder	Hidden Layers	2
	Hidden Units	512
	Activation	ELU
	Latent Dim	8
Decoder	Hidden Layers	2
	Hidden Units	512
	Activation	ELU

Table 5: We list our network architecture parameters here.

B More Analysis

B.1 Evaluation of r_{behavior}

In our setting, r_{behavior} is utilized for skill-specific behavior constraint, which is important for sim2real transfer of each skill. In this part, we evaluate the effectiveness of each r_{behavior}.

B.1.1 Experiment Setting

For locomotion policy π^{loco}, we mainly evaluate how well the policy can track a given speed command. For body-pose-adjustment policy π^{body}, we want to evaluate how much it can change the humanoid body and provide more reachable space for the hands. So we mainly evaluate the body-pose-adjustment range. For both controllers, we conduct experiments in simulation and real world.

B.1.2 Experiment Metrics

Considering the different focuses of the two primitive skill controllers, we design distinct evaluation metrics for each. For locomotion policy, we report the v^c tracking error err_v, ω^c tracking error err_ω, where v^c tracking error can be written as $err_v = (v_x - v_x^c)^2 + (v_y - v_y^c)^2$, ω^c tracking error

r_{command}

Term	Equation	Scale
Locomotion		
Linear velocity tracking	$\exp\{-5.0\lvert v^c - v\rvert^2\}$	1.0
Angular velocity tracking	$\exp\{-7.0\lvert \omega^c - \omega\rvert^2\}$	1.0
Body-Pose-Adjustment		
Body height tracking	$\exp\{-4.0\lvert h^c - h\rvert^2\}$	1.0
Pitch angle tracking	$\exp\{-4.0\lvert p^c - p\rvert^2\}$	1.0
Hand-Reaching		
End-effector pose tracking	$\exp\{-4.0\lvert e^c - e\rvert^2\}$	1.0

r_{behavior}

Term	Equation	Scale
Locomotion		
Gait velocity tracking	$\sum_{\text{foot}}[1 - \text{C}_{\text{foot}}(t)]\lvert v_{\text{foot}}\rvert^2$	1.0
Gait force tracking	$\sum_{\text{foot}}[\text{C}_{\text{foot}}(t)]\lvert \text{f}_{\text{foot}}\rvert^2$	1.0
Body-Pose-Adjustment		
Base roll error	$\exp\{-4.0r^2\}$	1.0
Leg pos symmetry	$\lVert q_{\text{left\_leg}} - q_{\text{right\_leg}}\rVert_2$	0.5
Leg torque symmetry	$\lvert a^{\text{left\_leg}}_{\text{low}} - a^{\text{right\_leg}}_{\text{low}}\rvert$	-0.2
Contact ground	$c_{left} * c_{right}$	1.0

$r_{\text{regularization}}$

Term	Equation	Scale
Action acc	$\lVert a_t - 2a_{t-1} + a_{t-2}\rVert_2$	-0.01
Action rate	$\lVert a_t - a_{t-1}\rVert_2$	-0.01
Collision	$\mathbb{1}$(undesired collision)	-5.0
Default joint error	$\exp\{-2.0\lvert q - q_0\rvert^2\}$	0.2

r_{task}

Term	Equation	Scale
Point Touch		
Point touch	$\exp\{-dist(hand, point)\}$	1.0
Box Pickup		
Hand approach	$\exp\{-dist(hand, box\_side)\}$	1.0
Lift box	$\exp\{-dist_z(box\_height, 1.4)\}$	1.0

Table 6: We list all of our reward terms and corresponding weights here.

can be written as $err_\omega = (\omega - \omega^c)^2$. We randomly sample 1000 trajectories in simulation and 5 trajectories in real world and report the mean error. For body-pose-adjustment policy, we compare the adjustment range of body height ran_h and pitch angle ran_p when there is no significant physical artifacts (e.g., jerking, sliding).

B.1.3 Experiment Results

Table 8 shows the results in simulation. Table 9 shows the results in real world. We found that absence of r_{behavior} will lead to a minor performance degradation in simulation and a significant performance drop in real world, which proves the effectiveness of r_{behavior} for sim2real transfer.

Parameter	**Value**
Environments	4096
Steps per episode	24
Total batch size	$4096 \times 24 = 98304$
Value loss coefficient (*value_loss_coef*)	1.0
Value loss clip parameter	0.2
Entropy coefficient (*entropy_coef*)	0.005
Training episodes	30,000
Learning rate	5.0×10^{-4} (adaptive)
Discount factor (γ)	0.995
GAE factor (λ)	0.95
Target KL divergence	0.01
Max gradient norm	1.0

Table 7: We list hyperparameters for our PPO training here.

Method	Locomotion		Body-Pose-Adjustment	
	$err_v \downarrow$	$err_\omega \downarrow$	ran_h	ran_p
w/o r_{behavior}	0.106	0.115	[0.66, 0.98]	[0.0, 0.78]
Ours	**0.069**	**0.058**	**[0.55, 1.00]**	**[0.0, 1.2]**

Table 8: We evaluate the effectiveness of r_{behavior} in simulation.

B.2 Evaluation of Latent Space

As mentioned before, we found that generating commands in latent skill space is better than generating commands directly in the primary task space. We suspect it is mainly because that latent skill representation is more *compact* and *structured* than raw combination of skill indicator and commands. In this part, we will evaluate this empirically.

B.2.1 Experiment Setting

For compactness, we mainly evaluate the impact of different command dimensions. For structuredness, we found that the one-hot skill indicator makes the action space for planning policy discontinuous and brings difficulty to planning policy learning. Therefore, we design an experimental setup where the target point and the box were positioned at an ergonomic height to the humanoid. We remove the body-pose-adjustment policy π^{body} so the humanoid doesn't have to output skill indicator to select primitive skills.

B.2.2 Experiment Metrics

We still use *Success Rate* and *Distance Error* as mentioned before for evaluation.

B.2.3 Experiment Results

We show the impacts of different command dimensions in Table 10. Our results show that compact latent skill representations lead to a reduced planning policy action space, thus facilitating policy learning. We show the impacts of skill indicators in Table 11. We found that in a scenario where choosing which skill to activate is needed, the improvement brought by latent space will be more pronounced. This is likely because the presence of skill indicator introduces discontinuities in the action space, thereby hindering policy learning.

Method	Locomotion		Body-Pose-Adjustment	
	$err_v \downarrow$	$err_\omega \downarrow$	ran_h	ran_p
w/o r_{behavior}	0.164	0.178	[0.77, 0.98]	[0.0, 0.46]
Ours	**0.084**	**0.070**	**[0.55, 1.00]**	**[0.0, 1.2]**

Table 9: We evaluate the effectiveness of r_{behavior} in real world.

Method	Point touch		Box pickup	
	SR ↑	*DE* ↓	*SR* ↑	*DE* ↓
R^{2}S^2 w/o LS	56.9	0.10	43.3	0.19
R^{2}S^2 32D	67.4	0.11	49.6	0.21
R^{2}S^2 24D	86.9	0.07	81.4	0.12
R^{2}S^2 16D	99.7	0.05	99.4	0.07
R^{2}S^2 8D (Ours)	**100**	**0.03**	**100**	**0.04**

Table 10: We evaluate the impacts of different command dimensions. "SR" is short for *Success Rate* and "DE" is short for *Distance Error*.

Method	Point touch		Box pickup	
	SR ↑	*DE* ↓	*SR* ↑	*DE* ↓
Only π^{loco}				
R^{2}S^2 w/o LS	81.8	0.06	79.1	0.12
Ours	**100 (+18.2)**	**0.02 (-0.04)**	**100 (+20.9)**	**0.03 (-0.09)**
π^{loco} and π^{body}				
R^{2}S^2 w/o LS	56.9	0.10	43.3	0.19
Ours	**100 (+43.1)**	**0.03 (-0.07)**	**100 (+56.7)**	**0.04 (-0.15)**

Table 11: We evaluate the impacts of skill indicators. "SR" is short for *Success Rate* and "DE" is short for *Distance Error*.

C More Applications

We found that our work yields, as a byproduct, a tele-operation system with an extensive reachable space. Concretely, we can directly control each π^{ensem} with commands including $v_x^c, v_y^c, \omega^c, h^c, p^c, e^c$ from a joystick or motion capture data of a tele-operator. Our tele-operation system enables loco-manipulation tasks with a large reachable space.