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

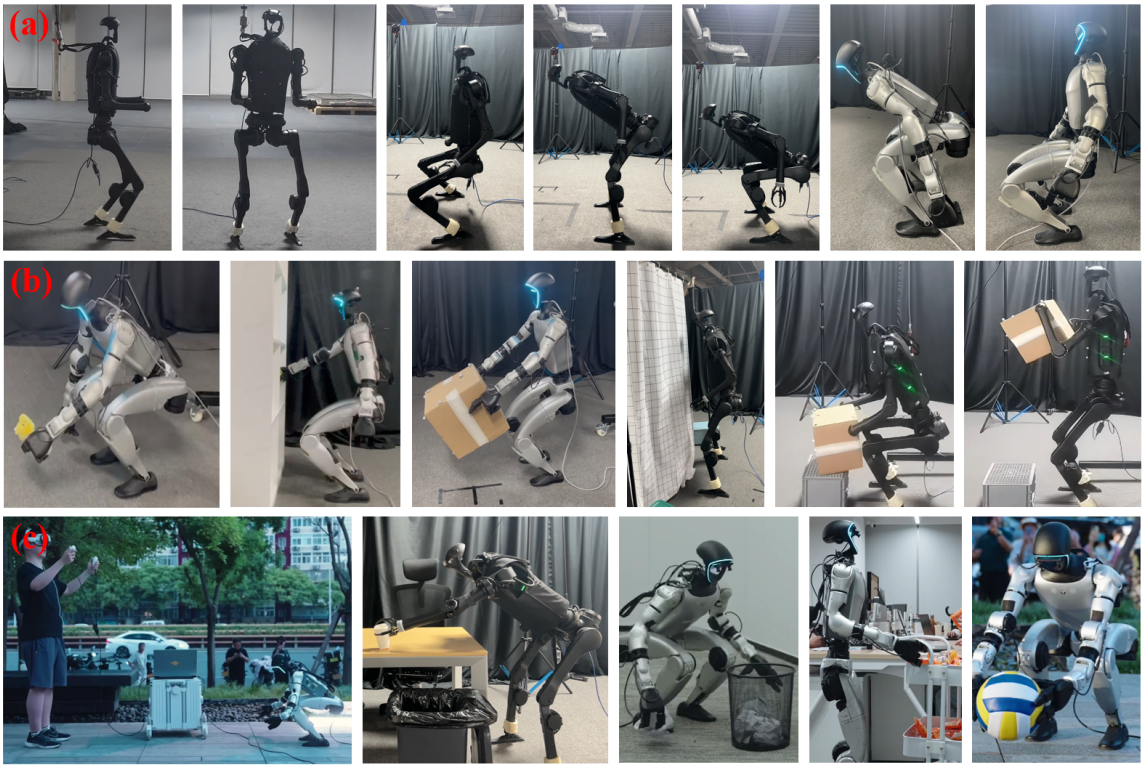

Fig. 1: **(a)** The humanoid showcases multiple real-world-ready primitive skills, including locomotion and body-pose-adjustment. **(b)** The humanoid autonomously accomplishes various WBC tasks. **(c)** The humanoid performs various tasks under our proposed teleoperation system.

***Abstract*— Humans possess a large reachable space in the 3D world, enabling interactions with objects at varying heights and distances. However, realizing such large-space reaching on humanoids is a complex whole-body control (WBC) problem and requires the robot to master and intelligently utilize diverse skills simultaneously. Learning from scratch often leads to optimization difficulty and poor sim2real transferability. To address these challenges, we present Real-world-Ready Skill Space (R^2S^2), a structural skill prior that helps autonomous whole-body-control task execution in an efficient manner while maintaining sim2real transferability. Inheriting knowledge from a set of real-world-ready primitive skills to ease multi-skill learning, R^2S^2 further expands the capability of primitive skills and learns a unified structural skill representation. By sampling from R^2S^2, we unleash humanoid reaching potential in many real-world tasks. As a beneficial side effect, R^2S^2 can also support humanoid whole-body teleoperation with a large reachable space. We validate the generalizability of R^2S^2 in various challenging goal-reaching tasks across different robot platforms, simulation and real world. We show some examples in Figure 1.**

I. INTRODUCTION

Many human daily tasks can be viewed as reaching a series of points under certain conditions. A large reachable space enables 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 a similar workspace [1]. However, this presents a complex whole-body control challenge and requires mastering and intelligently utilize diverse skills—including base positioning and reorientation, height and body posture adjustments, and end-effector pose control within a dynamically unstable system [2].

Traditional model-based control methods [3]–[6] 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 [7]–[11]. Can we utilize it to endow humanoids the capability to accomplish tasks requiring a human-level large reachable space? We found that optimiza-

tion and sim2real difficulty for such a complex whole-body control (WBC) problem are major concerns. As mentioned before, multiple skills are required to be mastered and intelligently utilized for unleashing humanoid reaching potential. Learning all skills together from scratch is difficult. Existing works often rely on highly intricate reward engineering and curriculum design to balance the rewards [12] or trajectory optimization to provide guidance [13]. Additionally, a stable real-world performance usually requires iterative sim2real deployment to diagnose the sim2real gap and design corresponding constraints to mitigate the behavior discrepancy. For an end-to-end WBC policy, the coupling between different skills as well as between planning and control makes both sim2real diagnosis and constraint design much more challenging. Can we design prior knowledge to assist WBC tasks for less optimization and sim2real difficulty?

Toward this end, we propose **Real-world-Ready Skill Space ($\mathbf{R^2S^2}$)**, aiming at constructing a skill space that encompasses and encodes various real-world-ready motor skills. Sampling from it, *the learned space can serve as a structural skill prior and helps autonomous WBC task execution with minimal reward engineering efforts in a sim2real transferable manner.*

To be specific, we first construct a library of pre-trained primitive skills to ease the optimization and sim2real transfer of multi-skill learning. These skills are task-agnostic and generalizable across different scenarios. Each skill can be individually tuned and sim2real evaluated for optimal real-world performance with minimal engineering efforts (because of decoupled training). Though separated primitive skills can provide real-world-ready prior, they are insufficient to serve as a practical skill space for two reasons: 1) separated training makes the coordination and transition between different skills out-of-distribution; 2) different skills often have mismatched command spaces, lacking a unified representation for multi-skill planning.

Therefore, we introduce a core stage called **heterogeneous skill ensembling**. At this stage, we inherit knowledge from pre-trained skills and expand it into a unified neural skill representation. We achieve this by first constructing a heterogeneous skill training environment, then dynamically combining Imitation Learning (IL) and Reinforcement Learning (RL) to train a CVAE-based student policy, which inherits real-world-ready skill prior from the pre-trained teacher policies and explores new coordination and transition skills. Different from existing hierarchical humanoid control frameworks [12], [13] where the planning is often conducted in the primary command space of the MLP-based low-level controller, the student network in our framework is designed as a CVAE to neurally model the motor skill distribution conditioned on proprioception, which proves to be a more efficient representation for multi-skill planning.

With task-specific planning policies trained to sample from $\mathrm{R^2S^2}$, our method enables the robot to autonomously accomplish various complex whole-body-control tasks in a sim2real transferable manner with minimal reward engineering efforts. In this work, we mainly focus on solving WBC tasks requiring a large reachable space. As a beneficial side effect, $\mathrm{R^2S^2}$ also enables us to build a humanoid whole-body teleoperation system capable of reaching its full workspace, not just for table-top pick-and-place tasks.

We validate the generalizability of $\mathrm{R^2S^2}$ across high-dof Unitree G1 (29 dofs) and full-sized Unitree H1 (1.8 meters tall) on various autonomous goal-reaching tasks. Extensive experiments are conducted in both simulation and real world to evaluate the effectiveness of our major designs.

In summary, our main contributions are fourfold:

- We propose $\mathbf{R^2S^2}$, a structural skill prior that helps complex WBC task execution with minimal reward engineering efforts in a sim2real transferable manner.
- We propose a framework to construct $\mathrm{R^2S^2}$. As the core, we propose **heterogeneous skill ensembling**, which can inherit knowledge from pre-trained skills and expand it into a unified skill representation.
- We implement $\mathrm{R^2S^2}$ to unleash humanoid reaching potential in real world. As a beneficial side effect, we also utilize $\mathrm{R^2S^2}$ to build a humanoid whole-body teleoperation system with a large reachable space.
- We validate the generalizability of $\mathrm{R^2S^2}$ across different humanoid platforms, tasks, sim and real.

II. Related Works

Humanoid Robot Learning. Reinforcement learning (RL) has achieved great progress in recent humanoid robot learning. Researches on locomotion [11], [14]–[18] aim at provide bipedal humanoids with the ability to traverse different terrains in a stable and agile manner. But these works often focus only on the lower body of humanoids and ignore their whole-body reaching and interaction abilities. Learning-based humanoid whole-body-control [7]–[13], [19]–[25] recently demonstrate new capabilities and push the boundaries of humanoid robots. In [12], [13], humanoid reaching potential is unleashed by combining the capability of locomotion and body posture adjustment. However, these works either rely on highly intricate reward engineering and curriculum design to balance the rewards [12] or trajectory optimization to provide guidance [13], making it hard to extend to new skills. In this work, we design a novel framework to ensemble multiple pre-trained skills without affecting the acquisition of each individual skill, having the potential to scale up efficiently to more complex scenarios. Compared with existing hierarchical humanoid control frameworks [12], [13] where the planning is often conducted in the primary command space, we focus more on acquiring an efficient skill representation to connect humanoid multi-skill control and planning in this work.

Skill Space Learning. In physics-based character animation, skill spaces [26]–[31] are often learned to reuse motion prior from motion capture datasets. Motion imitation [27], [29]–[31] or adversarial learning [26], [32], [33] 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 trained 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 [28], [34], [35], transferring such a paradigm to real-world humanoid is challenging because of lack of high-quality motion datasets and cross-embodiment gap. Rather than utilizing motion prior from motion datasets, we inherit real-world-ready skill prior from a set of pre-trained skills and expand it into a unified skill space.

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

In this section, we describe how to learn a **Real-world-Ready Skill Space ($\mathbf{R^2S^2}$)** and utilize it to support practical WBC tasks, with a focus on unleashing humanoid reaching potential. We first introduce the construction of primitive skill library in Section III-A. We then present the core **heterogeneous skill ensembling** stage in Section III-B. Finally, we show how to sample from $\mathrm{R^2S^2}$ to efficiently solve various real-world goal-reaching tasks in Section III-C. The pipeline is shown in Figure 3. We use PPO [36] for all of our policy training, domain randomization for sim2real transfer, and Isaac Gym [37] for simulation.

A. Primitive Skill Library

Aiming at unleashing humanoid reaching potential, we design the primitive skill library $\{\pi_i^{\text{prim}}\}_{i=1}^n$ as locomotion, body-pose-adjustment (changing body height, bending over) and hand-reaching, which can handle with many goal-reaching scenarios. Compared with training a generalist controller from scratch, such a decomposition avoids performance degradation brought by multi-skill learning.

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 skill policy only takes relevant body part information as observation, lower-body for locomotion and body-pose-adjustment and two arms for hand-reaching. $\mathcal{A}^{\text{prim}}$ 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_{command} 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. For detailed rewards, please refer to Table I.

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 [38], [39]:

$$r_{\text{behavior}}^{\text{loco}} = r_{\text{gait_velocity}} + r_{\text{gait_force}}, \tag{2}$$

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

$$r_{\text{gait_force}} = \sum_{\text{foot}} [\mathrm{C}_{\text{foot}}(t)] |\mathrm{f}_{\text{foot}}|^2, \tag{4}$$

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, $\mathcal{G}^{\text{body}} = \langle h^{\text{c}}, b^{\text{c}} \rangle$ tracks the base height and torso bending angle in the global frame. We found that for such a skill,

TABLE I: We list all of our reward terms and corresponding weights here.

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	$\exp\{-4.0\lvert b^c - b\rvert^2\}$	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	$\sum_{\text{foot}}[1 - \mathrm{C}_{\text{foot}}(t)]\lvert v_{\text{foot}}\rvert^2$	1.0
Gait force tracking	$\sum_{\text{foot}}[\mathrm{C}_{\text{foot}}(t)]\lvert \mathrm{f}_{\text{foot}}\rvert^2$	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	$\lvert a_{\text{low}}^{\text{left_leg}} - a_{\text{low}}^{\text{right_leg}}\rvert$	-0.2
Contact ground	$c_{\text{left}} * c_{\text{right}}$	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	undesired collision	-5.0
Default joint error	$\exp\{-2.0\lvert q - q_0\rvert^2\}$	0.2
r_{task}		
Term	**Equation**	**Scale**
Single-point Touch		
Single-point touch	$\exp\{-dist(hand, point)\}$	1.0
Dual-point Touch		
Dual-point touch	$\exp\{-dist(hands, points)\}$	1.0
Shelf Touch		
Shelf 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

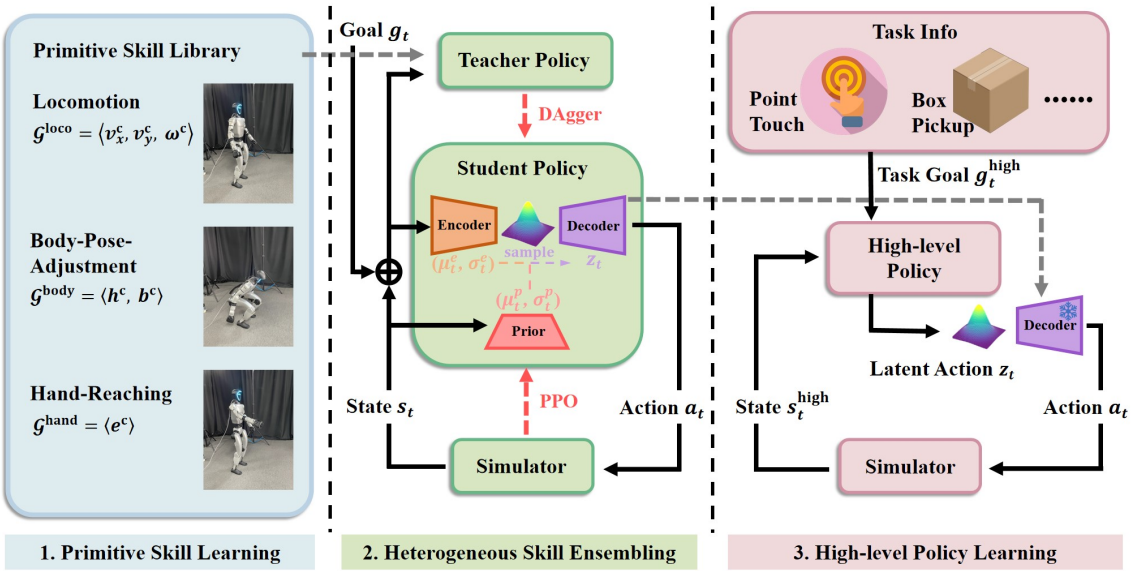

Fig. 3: We present R^2S^2, a structural skill prior that helps autonomous WBC task execution in an efficient and sim2real transferable manner. 1) We ease multi-skill learning by first constructing a primitive skill library, each individually tuned and sim2real evaluated. 2) We heterogeneously ensemble primitive skills to expand their capability and learn a unified neural skill representation. 3) Sampling from R^2S^2, we unleash humanoid reaching potential in many real-world tasks.

kinematic and dynamic symmetry is important for real-world stability, so we introduce:

$$r_{\text{behavior}}^{\text{body}} = r_{\text{base_roll}} + r_{\text{leg_pos}} + r_{\text{leg_torque}} + r_{\text{touch_ground}}, \quad (5)$$

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, $\mathcal{G}^{\text{hand}} = \langle e^c \rangle$ 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.

B. Heterogeneous Skill Ensembling

Given real-world-ready primitive skills $\{\pi_i^{\text{prim}}\}_{i=1}^n$, a straight attempt to reuse these primitive skills for WBC tasks is directly planning in their primary command spaces, for example, training a planner policy to predict which primitive skill to be activated at each timestep and output corresponding primary commands (e.g., v_x^c, v_y^c, ω^c for locomotion). But individual primitive skills are actually insufficient for practical WBC tasks. Because of separated training, isolated 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 command space of primitive skills (v_x^c, v_y^c, ω^c for locomotion, h^c, b^c for body-pose-adjustment, and e^c for hand-reaching in our setting) are mismatched. Direct combination of skill indicators and corresponding skill commands is not a structural action space for planner policy and poses challenge for multi-skill planning. These two drawbacks make the primitive skills inefficient for planning as shown in Section IV-B.

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

Expanding Coordination and Transition Capability. The ensemble student policy $\pi^{\text{ensem}}(a_t|s_t, g_t)$ should not only retain the real-world transferability of primitive skills, but also expand new coordination and transition skills. To retain the real-world transferability of primitive skills, an effective choice is to leverage online imitation learning methods (e.g., DAgger [40]). In particular, we first construct a heterogeneous skill training environment to model skill coordination and transition: 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 randomly set the skill of a certain body part to transition from one to another in an episode to model skill transition. 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 situations are covered in the student policy training process.

However, relying only on imitation learning can not expand 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 dynamically adding IL loss and RL loss together. The IL, which is DAgger in our setting, distills real-world-ready skill prior from multiple teacher policies. Based on this, the RL, which is PPO in our setting, further encourages the policies to learn new behaviors for seamless transition and coordination. The reward function can be written as:

$$\mathcal{L}_{\text{Ensem}} = \lambda_1 \mathcal{L}_{\text{DAgger}} + \lambda_2 \mathcal{L}_{\text{PPO}}, \tag{6}$$

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. Instead of utilizing a sequential training strategy that first employs IL for pretraining followed by exclusive RL fine-tuning, we maintain the supervision signal from the teacher policies throughout the entire training process. Our strategy can prevent catastrophic forgetting of the skill prior provided by teacher policies. We let

$$\mathcal{L}_{\text{DAgger}} = \mathbb{E}_{(s,a^*)\sim\mathcal{D}_{\text{agg}}} \left[\| a_t^{\text{ensem}} - a_t^* \|^2 \right], \tag{7}$$

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 newly 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 Unified Neural Skill Representation. While the student policy can ensemble multiple primitive skills, mismatched command 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 [30], 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. The total loss in training π^{ensem} can be written as:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Ensem}} + \lambda_3 \mathcal{L}_{\text{Regu}} + \lambda_4 \mathcal{L}_{\text{KL}}, \tag{8}$$

where

$$\mathcal{L}_{\text{Regu}} = \| \mu^e(s_t, g_t) - \mu^e(s_{t+1}, g_{t+1}) \| \tag{9}$$

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

C. High-Level Planning in Real-World-Ready Skill Space

In this work, we mainly focus on utilizing R^2S^2 to solve WBC tasks requiring a large reachable space. Without loss of generality, we formulate many human daily activities as reaching a series of points $\{p_i\}_{i=1}^{m_1}$ under certain conditions $\{c_i\}_{i=1}^{m_2}$, aiming at unleashing humanoid reaching potential. The skill prior encoded in R^2S^2 helps autonomous task execution in a sim2real transferable manner with only points and conditions input. We achieve this by training task-specific high-level planners $\pi^{\text{plan}}(z_t|s_t^{\text{plan}}, g_t^{\text{plan}})$ with RL to sample from the learned latent skill space. 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 the frozen decoder $\mathcal{D}$. The training reward can be written as:

$$r_{\text{plan}} = r_{\text{task}} + r_{\text{regularization}}, \tag{10}$$

where r_{task} is task execution objective describing desired reaching points and task conditions. For example, in the box pickup task, the reaching points can be the midpoints of the two side surfaces of the box, and the condition can be lifting the box above a certain height. $r_{\text{regularization}}$ is the skill-agnostic regularization reward reused from the skill library construction stage to enhance motion stability. It is worth noting that with R^2S^2, we only need to define r_{task} at this stage. The humanoid can autonomously accomplish various tasks in a sim2real transferable manner without any additional designs. For detailed r_{task}, please refer to Table I.

IV. Experiments

In this section, comprehensive experiments in both simulation and real-world will be conducted to answer the following questions: **Q1.** (Section IV-A) Compared with baseline methods, can Real-world-Ready Skill Space (R^2S^2) better assist with various WBC tasks in an efficient and sim2real transferable manner? **Q2.** (Section IV-B) How does each part of R^2S^2 contribute to the final results? The quantitative results are reported on Unitree H1.

A. Unleashing Humanoid Reaching Potential

In this part, we want to compare our method with baseline methods to evaluate whether R^2S^2 assists with humanoid whole-body control tasks, with a focus on unleashing humanoid reaching potential.

TABLE II: We compare R^2S^2 with baseline methods. "SR" is short for **Success Rate** and "DE" is short for **Distance Error**. Our method exceeds baselines in task accomplishment performance and sim2real transferability.

Method	Single-point Touch		Dual-point Touch		Shelf Touch		Box Pickup		Sim2Real
	SR(%) ↑	*DE(m)* ↓	*SR(%)* ↑	*DE(m)* ↓	*SR(%)* ↑	*DE(m)* ↓	*SR(%)* ↑	*DE(m)* ↓	
Sim									
Vanilla PPO [36]	11.63	0.49	5.71	0.52	13.20	0.37	0.00	0.65	✗
Vanilla DreamerV3 [41]	30.51	0.15	22.83	0.33	8.93	0.28	0.00	0.59	✗
HumanoidBench [42]	**100**	0.04	**100**	0.04	47.69	0.13	0.00	0.44	✗
Ours	**100**	**0.03**	**100**	**0.03**	**100**	**0.02**	**100**	**0.04**	✓
Real									
Ours	**100**	**0.03**	**100**	**0.03**	**100**	**0.03**	**90**	**0.08**	N/A

TABLE III: We evaluate the effectiveness of our major designs. "SR" is short for **Success Rate** and "DE" is short for **Distance Error**. Each of our major designs contributes to the final results.

Method	Single-point Touch		Dual-point Touch		Shelf Touch		Box Pickup		Sim2Real
	SR(%) ↑	*DE(m)* ↓	*SR(%)* ↑	*DE(m)* ↓	*SR(%)* ↑	*DE(m)* ↓	*SR(%)* ↑	*DE(m)* ↓	
R^2S^2 w/o PS	49.58	0.14	43.15	0.12	47.81	0.22	30.74	0.26	✗
R^2S^2 w/o SE	30.51	0.15	26.39	0.16	28.34	0.29	22.82	0.33	✗
R^2S^2 w/ Seq-ILRL	47.16	0.10	45.98	0.09	54.72	0.09	29.53	0.24	✗
R^2S^2 w/o LS	56.87	0.10	52.38	0.07	44.86	0.17	43.29	0.19	✓
Ours	**100**	**0.03**	**100**	**0.03**	**100**	**0.02**	**100**	**0.04**	✓

1) Experiment Setting

In our experiment setting, we select four representative goal-reaching tasks, including scenarios involving a single hand, both hands, obstacle avoidance, and humanoid-object interaction:

- `Single-point Touch`: We randomly set one point within a $2m \times 2m$ square in front of the robot, with a height ranging from 0.1 meters to 2.0 meters. The humanoid is asked to touch the point with one hand.
- `Dual-point Touch`: We randomly set two points within a $2m \times 2m$ square in front of the robot, with each height ranging from 0.1 meters to 2.0 meters. The distance between the points is less than 1 meter. The humanoid needs to touch each point with one hand.
- `Shelf Touch`: We randomly set a point inside a multi-layer shelf, with height ranging from 0.1 meters to 2.0 meters. The humanoid is asked to touch the point without causing a collision with the shelf.
- `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.

2) Experiment Metrics

We use two metrics:

- **Success Rate**: For point-touch tasks, **Success Rate** records the percentage of trials that humanoids successfully touch *each* target point within 5 cm. For Box Pickup, **Success Rate** means the percentage of trials that humanoids successfully lift the box above 1.3 meters.
- **Distance Error**: For point-touch tasks, **Distance Error** is the averaged closest distance between the humanoid's end effector and the corresponding target point in a touch. For Box Pickup, **Distance Error** is the closest distance between the box and 1.4 meters height.

In simulation, all metrics are averaged over 10000 trials. Beyond task accomplishment results in simulation, we also measure the sim2real transferability. For sim2real transferable methods, we also report real-world results averaged over 10 trials.

3) Baselines

We choose three baseline methods, including model-free, model-based, and hierarchical RL:

- **Vanilla PPO** [36]: We implement a vanilla PPO without any manually designed prior.
- **Vanilla DreamerV3** [41]: We implement a vanilla DreamerV3 without any manually designed prior.
- **HumanoidBench** [42]: We adopt the hierarchical RL framework in HumanoidBench [42], whose low-level policy is a two-hand reaching policy.

4) Experiment Results

The results are shown in Table II. R^2S^2 exceeds baseline methods in task accomplishment performance and sim2real transferability. Vanilla RL methods without any prior, whether model-based or model-free, struggle to learn to accomplish even simple tasks and lack sim2real transferability due to unstable motor behaviors. For the hierarchical method in HumanoidBench [42], although it introduces auxiliary rewards that enable successful completion of point-touch tasks, the heavy vibration significantly degrades its performance in other more challenging tasks and prevents its transfer to the real world. The comparison of motor behaviors is shown in Figure 4.

B. Evaluation of Real-world-Ready Skill Space

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

1) Experiment Setting and Metrics

We reuse the experiment setting and metrics in Section IV-A. In this part, we run all experiments in simulation.

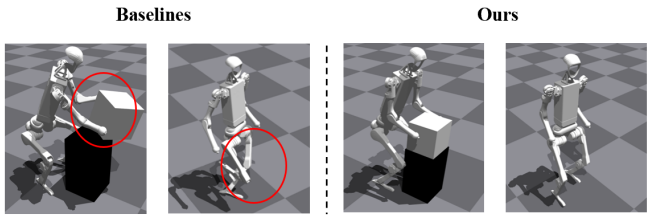

Fig. 4: We compare the motor behaviors with baseline methods. Our method significantly helps sim2real transfer.

2) Baselines

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

- **R^2S^2 w/o PS (Primitive Skills)**: We do not use pre-trained individual primitive skills. Instead, we train a multi-skill controller from scratch to track different commands. We adopt this baseline to verify the necessity of decoupling the multi-skill training to avoid performance degradation.
- **R^2S^2 w/o SE (Skill Expansion)**: During heterogeneous skill ensembling, we only use IL to train the student policy. In this setting, the student policy is not encouraged to explore new skills with RL. We adopt this baseline to mainly verify the importance of expanding coordination and transition capability.
- **R^2S^2 w/ Seq-ILRL (Sequential IL and RL)**: We implement a sequential strategy to combine IL and RL in the student policy training process. We first use only IL for pretraining followed by exclusive RL fine-tuning. We adopt this baseline to validate the necessity of maintaining supervision signals throughout the entire training process in prevent of catastrophic forgetting.
- **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 the primary mismatched command space. We adopt this baseline to evaluate the effectiveness of our latent skill space.

3) Experiment Results

We report the results in Table III. For R^2S^2 w/o PS, R^2S^2 w/o SE, R^2S^2 w/ Seq-ILRL, the poor task performance is mainly attributed to insufficient learning of low-level skills, making it difficult for the planning policy to select suitable skills. We visualize the normalized skill command tracking accuracy in Figure 5. For R^2S^2 w/o PS, the low-level controller cannot learn to track different skill commands very well from scratch. When sim2real transferred, the motor behavior is unstable. For R^2S^2 w/o SE, the controller is trained with only IL. It 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, thus it is also not sim2real transferable. R^2S^2 w/ Seq-ILRL suffers from catastrophic forgetting. Once RL fine-tuning begins, the policy's performance rapidly deteriorates, and it forgets the knowledge acquired during the IL pre-training stage, ultimately performing even worse than training from scratch.

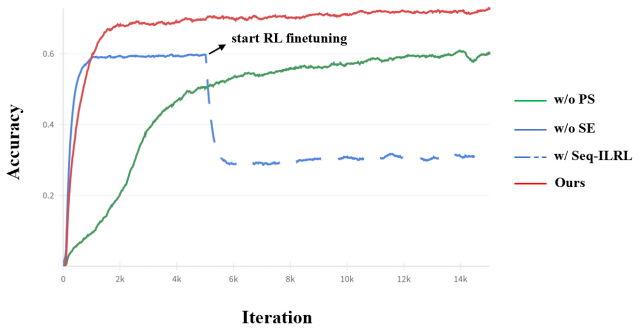

Fig. 5: We compare the normalized skill command tracking accuracy.

For R^2S^2 w/o LS, the skill command tracking accuracy is not affected. The performance degradation is mainly attributed to the difficulty of high-level planner learning. Sampling from raw combination of skill indicators and mismatched command spaces leads to significantly less efficient RL exploration for high-level planners. Compared with R^2S^2 w/o LS, our latent space provides a more structural action space for multi-skill planning.

V. Teleoperation System

As a beneficial side effect, the ensemble student policy $\pi^{\text{ensem}}(a_t|s_t, g_t)$ can also be utilized to build a humanoid whole-body teleoperation system. Inspired by [43], we mount a single stereo RGB camera on the robot head. During teleoperation, the VR device worn by the user receives streaming real-time, ego-centric robot observations. The hand pose is captured by the VR device and retargeted to humanoid hand via Inverse Kinematics (IK). We design a pair of split joysticks. The user can hold one joystick in each hand and press the buttons on each joystick to control the lower-body movement of the robot and hand opening/closing without affecting the hand pose.

VI. Conclusion and Limitation

In this work, we propose R^2S^2, a structural skill prior to help the execution of autonomous WBC tasks in an efficient and sim2real transferable manner. We validate the generalizability of R^2S^2 across different humanoid platforms, tasks, simulation and real world. Although we believe our method can be extended to more general and complex whole-body control systems, currently its limitation 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) Although we achieve seamless coordination and transition at skill ensembling stage, how to blend spatially overlapping skills is still challenging. 3) For now, we rely on the 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.* IEEE, 2005, pp. 7–12.

[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*, vol. 33, no. 1, pp. 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)*. IEEE, 2019, pp. 5280–5286.

[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*, vol. 7, no. 2, pp. 2377–2384, 2022.

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

[8] 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.

[9] 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.

[10] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbab, C. Pan, Z. Yi, G. Qu, K. Kitani, J. Hodgins, L. J. Fan, Y. Zhu, C. Liu, and G. Shi, "Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," 2025. [Online]. Available: https://arxiv.org/abs/2502.01143

[11] 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.

[12] 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.

[13] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang, "Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control," 2025. [Online]. Available: https://arxiv.org/abs/2505.03738

[14] 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.

[15] 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)*. IEEE, 2021, pp. 2811–2817.

[16] 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)*. IEEE, 2024, pp. 56–62.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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)*. IEEE, 2024, pp. 16 930–16 936.

[23] 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.

[24] 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.

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

[26] 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)*, vol. 41, no. 4, pp. 1–17, 2022.

[27] J. Won, D. Gopinath, and J. Hodgins, "Physics-based character controllers using conditional vaes," *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–12, 2022.

[28] H. Yao, Z. Song, B. Chen, and L. Liu, "Controlvae: Model-based learning of generative controllers for physics-based characters," *ACM Transactions on Graphics*, vol. 41, no. 6, pp. 1–16, 2022.

[29] 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*. Springer, 2024, pp. 403–421.

[30] 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.

[31] 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.

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

[33] 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*, 2023, pp. 1–9.

[34] 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*, vol. 43, no. 8, 2024, pp. i–ix.

[35] 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.

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

[37] 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.

[38] 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)*. IEEE, 2021, pp. 7309–7315.

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

[40] 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*, 2011, pp. 627–635.

[41] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[42] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel, "Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation," *arXiv preprint arXiv:2403.10506*, 2024.

[43] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," *arXiv preprint arXiv:2407.01512*, 2024.