MASKILLBLENDER: DECENTRALIZED WHOLE-BODY COORDINATION FOR MULTI-HUMANOID LOCO-MANIPULATION VIA SKILL BLENDING

Anonymous authors
Paper under double-blind review

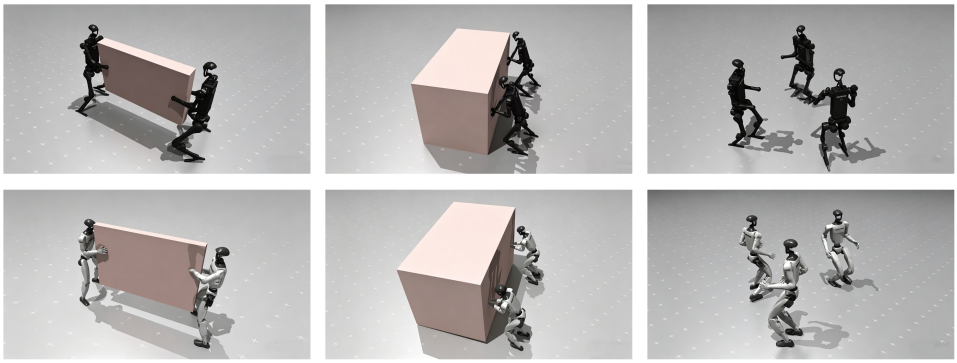


Figure 1: MASkillBlender enables decentralized multi-humanoid whole-body coordination across diverse tasks and humanoid embodiments without requiring task-specific motion references.

ABSTRACT

Coordinated multi-humanoid loco-manipulation is promising yet challenging due to high-dimensional whole-body control, decentralized decision making, and scalability. While recent reinforcement learning methods have improved single-humanoid whole-body control, extending them to the multi-humanoid setting remains nontrivial and often requires substantial reward engineering or task-specific design. We propose MASkillBlender, a general multi-agent reinforcement learning framework to achieve decentralized multi-humanoid whole-body coordination. By learning a shared decentralized high-level policy over reusable pre-trained single-humanoid skills, MASkillBlender enables coordinated behaviors using only task-level rewards, without requiring task-specific motion references. To improve learning efficiency, we further introduce a permutation-based data augmentation strategy for homogeneous multi-humanoid systems, and theoretically show that the permuted samples preserve the policy-gradient direction of the original samples under the homogeneous Markov game formulation. We evaluate MASkillBlender on multiple multi-humanoid coordination tasks across two humanoid embodiments. Simulation results demonstrate that the proposed framework consistently achieves strong task performance and enables coordinated behaviors across different tasks and humanoid embodiments. Project page: https://maskillblender.github.io/.

1 INTRODUCTION

Humanoid robots have substantial potential to replace or assist humans in completing daily tasks across diverse environments due to their human-like morphology (Gu et al., 2025). With advancements in reinforcement learning (RL) and robot hardware, the loco-manipulation capabilities of humanoids have been significantly enhanced (He et al., 2025). Nevertheless, many real-world tasks require multiple humanoids to coordinate their actions (Zhang et al., 2023). Compared with single-humanoid control, multi-humanoid settings introduce additional challenges due to complex inter-agent dependencies. Developing general learning methods for coordinated multi-humanoid control therefore represents an important step toward scalable and capable humanoid teams.

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

Recent studies have demonstrated increasingly sophisticated multi-humanoid whole-body interactions, showing the potential of learning-based approaches for coordinated humanoid control (Feng et al., 2026). However, existing methods typically rely on human motion references to guide policy learning through motion imitation or tracking objectives. While such motion priors provide effective guidance for acquiring natural whole-body behaviors, they also introduce additional data and supervision requirements, and the resulting motion-based objectives may compete with task rewards during optimization, potentially constraining the policy from discovering coordination strategies that are most effective for the task (Mo et al., 2026). Beyond the reliance on motion supervision, incorporating observations of other humanoids into each agent's policy has been widely adopted to facilitate coordination. While such inter-agent information can improve learning and coordination performance, it may hinder scalability to larger teams and complicate robust real-world deployment.

For single-humanoid loco-manipulation, skill-based learning has emerged as an effective paradigm that does not rely on task-specific motion references (Sferrazza et al., 2024; Kuang et al., 2025). By blending pre-trained primitive skills into the control policy, humanoids can reuse existing motor capabilities and reduce the burden of low-level control learning during downstream task training. However, extending this paradigm to multi-humanoid control is not straightforward, since skills pre-trained for individual humanoids do not encode the inter-agent coordination required by downstream tasks. Learning how to compose these skills across agents under coupled task dynamics therefore remains challenging.

In this work, we propose MASkillBlender, a general skill-based multi-agent RL (MARL) framework for decentralized coordination in homogeneous multi-humanoid systems. Under centralized training with decentralized execution (CTDE), a shared decentralized high-level policy learns to coordinate the composition of reusable pre-trained primitive skills across humanoids using local observations. The primitive skills remain fixed during downstream training, allowing the policy to focus on high-level skill composition and inter-agent coordination using only task-level rewards. To improve learning efficiency, we introduce a permutation-based data augmentation strategy that exploits the permutation symmetry of homogeneous multi-humanoid systems. To evaluate the proposed framework, we build three representative multi-humanoid coordination tasks spanning cooperative object manipulation and coordinated locomotion across two distinct humanoid embodiments, using only task-level rewards and limited local observations. To summarize, the main contributions of this work are as follows:

- We propose MASkillBlender, a general skill-based MARL framework that learns decentralized multi-humanoid coordination over reusable pre-trained single-humanoid skills using only task-level rewards, without requiring task-specific motion references.
- We incorporate a permutation-based data augmentation strategy into MASkillBlender and theoretically show that the augmented samples preserve the policy-gradient direction of the original samples.
- We evaluate the proposed MARL framework on diverse multi-humanoid coordination tasks across different embodiments, and extensive simulation results show that the learned policies produce versatile coordinated behaviors and effectively handle all tasks in a fully decentralized manner.

2 Related Works

Single-Humanoid Whole-Body Control with RL. RL has become a key paradigm for humanoid whole-body control, facilitating complex locomotion and manipulation (Radosavovic et al., 2024; Gu et al., 2024; Chen et al., 2024; 2026b; Lin et al., 2025). Existing RL-based methods can be broadly categorized according to whether the control policy relies on task-specific motion references. Reference-conditioned methods learn versatile whole-body behaviors by tracking motions obtained from MoCap datasets (Cheng et al., 2024; Ji et al., 2024; Yang et al., 2025) or human teleoperation (Fu et al., 2024; He et al., 2024b;a; Sun et al., 2025; Ze et al., 2025; Ben et al., 2025). While effective for single-humanoid loco-manipulation, extending this paradigm to multi-humanoid coordination requires coordinated motion references that are substantially more difficult to acquire. In contrast, task-driven methods learn loco-manipulation behaviors without conditioning on task-specific reference motions (Sferrazza et al., 2024; Kuang et al., 2025; Dao et al., 2024; Zhang et al., 2024). To address the difficulty of learning in high-dimensional state-action spaces, these methods typically employ task decomposition (Dao et al., 2024; Zhang et al., 2024) or skill-based hierarchical control architectures (Sferrazza et al., 2024; Kuang et al., 2025). For the latter, pre-trained low-level skills

are reused and composed by a high-level policy to reduce the burden of learning whole-body control from scratch, which empirically improves learning efficiency and enables diverse loco-manipulation behaviors across downstream tasks.

Multi-Humanoid Whole-Body Coordination with RL. Learning coordinated whole-body behaviors among multiple humanoids has recently received growing attention. SMPLOlympics (Luo et al., 2024) explores competitive interactions between physically simulated humanoids in diverse sports environments. For cooperative coordination, reference-based methods retarget multi-person motions to humanoids and learn interaction policies through motion tracking and interaction-specific objectives (Liu et al., 2025; Chen et al., 2026a), enabling complex whole-body interactions such as contact-rich social behaviors and synchronized motions. To alleviate the scarcity of multi-person motion data, CooHOI (Gao et al., 2024) and TeamHOI (Lionar & Lee, 2026) instead leverage single-human motion priors to facilitate cooperative object manipulation. CooHOI transfers single-agent object-interaction skills to cooperative settings, while TeamHOI employs masked motion priors and a teammate-aware policy to support cooperative carrying with varying team sizes. Note that these methods are typically task-specific and require high-quality motion retargeting to obtain suitable humanoid references. CDA-PPO (Baltes et al., 2025) follows a conventional MARL formulation and learns cooperative object transportation from scratch without motion supervision. However, it is specifically designed for a two-humanoid setting and relies on extensively engineered locomotion and cooperation rewards as well as explicit teammate observations. As a result, it remains a huge challenge to develop general reference-free RL methods for diverse multi-humanoid whole-body coordination with lightweight reward engineering.

3 PRELIMINARIES

3.1 HOMOGENEOUS MARKOV GAMES

Markov Games (MGs) provide a general framework for formulating multi-agent policy learning problems (Zhang et al., 2021). An MG is defined as $\langle N, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \{R_i\}_{i=1}^N, p, \gamma\rangle$, where N is the number of agents, $\mathcal{S}$ is the state space, $\mathcal{A}_i$ is the action space of agent i with $\mathcal{A} = \prod_{i=1}^N \mathcal{A}_i$ being the joint action space, and $R_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the corresponding reward function. The transition function $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, +\infty)$ specifies the environment dynamics, and $\gamma \in [0, 1)$ is the discount factor. For homogeneous agents, a subclass of MGs, termed H-MGs (Chen et al., 2022; Hu et al., 2024), can be defined as follows:

Definition 1. An MG $\langle N, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \{R_i\}_{i=1}^N, p, \gamma\rangle$ is an H-MG if it satisfies the following conditions:

- The state space is the product space of N homogeneous local state spaces, i.e., $s = (s_1, \dots, s_N) \in \mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_N$ with $\mathcal{S}_i = \mathcal{S}_j, \forall i, j \in \mathcal{I}$, where $s_i \in \mathcal{S}_i, \forall i \in \mathcal{I}$. The local action spaces of all agents are identical, i.e., $\mathcal{A}_i = \mathcal{A}_j, \forall i, j \in \mathcal{I}$[1].
- The joint reward function $R(s_t, a_t) = (R_1(s_t, a_t), \dots, R_N(s_t, a_t))$ is permutation preserving, and the transition function is permutation invariant, i.e., for any $M \in \mathcal{M}$, $s_t = (s_{1,t}, \dots, s_{N,t}) \in \mathcal{S}$, and $a_t = (a_{1,t}, \dots, a_{N,t}) \in \mathcal{A}$, it holds
$$R(Ms_t, Ma_t) = MR(s_t, a_t), \quad p(s_{t+1}|s_t, a_t) = p(Ms_{t+1}|Ms_t, Ma_t).$$
- Each agent $i \in \mathcal{I}$ obtains its local observation through a surjective function $o_i : \mathcal{S} \to \mathcal{O}$, where the observation space $\mathcal{O}$ is shared by all agents. Furthermore, the joint observation $o(s) = (o_1(s), \dots, o_N(s))$ is permutation preserving, i.e., for any $s \in \mathcal{S}$ and $M \in \mathcal{M}$, it holds
$$o(Ms) = Mo(s).$$

3.2 PROBLEM FORMULATION

We formulate the multi-humanoid whole-body control policy learning problem as a MARL problem under the H-MG framework. Notice that H-MGs are capable of modeling diverse multi-humanoid

[1] All notations are summarized in Appendix A.

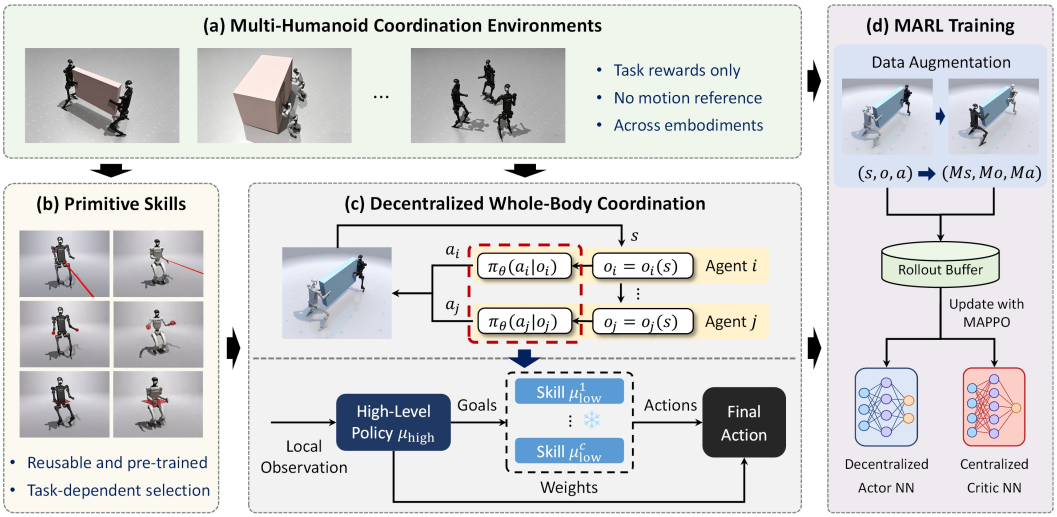


Figure 2: Overview of MASkillBlender. (a) MASkillBlender enables diverse multi-humanoid coordination across different humanoid embodiments without task-specific motion references or extensive reward engineering. (b)-(c) For each task, relevant pre-trained primitive skills are selected and blended by a shared decentralized high-level policy. (d) The high-level policy is trained with MAPPO using permutation-based data augmentation.

coordination tasks, where one example is provided in Appendix B to give more intuition about this formulation. Following prior work on single-humanoid RL (Zhang et al., 2024; Kuang et al., 2025), the local observation of each humanoid consists of its proprioceptive state and a local task goal. The local action corresponds to target joint positions of the humanoid's whole body, which are mapped to motor torques through PD control. Moreover, all humanoids share a team reward function, denoted by R_{team}, such that $R_{\text{team}} = R_i, \forall i \in \mathcal{I}$. Let $\pi^{\text{tot}} : \mathcal{S} \times \mathcal{A} \to [0, \infty)$ be the joint policy, based on which the value function is defined as $V_{\pi^{\text{tot}}}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{\text{team}}(s_t, a_t) | s_0 = s\right]$. The target is to find an optimal policy π_*^{tot} that maximizes the objective function $J(\pi^{\text{tot}}) = \mathbb{E}_{s_0 \sim \rho}\left[V_{\pi^{\text{tot}}}(s_0)\right]$, where ρ denotes the initial state distribution over $\mathcal{S}$.

4 Methodology

In practice, it is challenging to learn coordinated humanoid whole-body control policies with vanilla MARL. Without task-specific motion references, learning from scratch requires the policy to simultaneously acquire low-level motor control and inter-agent coordination. Moreover, the data-collection cost increases substantially as more humanoids are included into the environment, making iterative reward engineering particularly costly. To deal with these challenges, we propose MASkillBlender (see Fig. 2), which combines MARL with reusable pre-trained skills for decentralized multi-humanoid coordination and permutation-based data augmentation for improved data efficiency. We first introduce the decentralized skill-blended control policy in Sec. 4.1, then the permutation-based data augmentation in Sec. 4.2, and finally the overall MARL framework in Sec. 4.3.

4.1 Decentralized Skill-Blended Control Policy

As shown in Fig. 2(c), we adopt decentralized control policies for scalable multi-humanoid coordination. We employ policy sharing (Gupta et al., 2017) due to its empirical effectiveness in MARL (Hu et al., 2023) and theoretical support for optimality in homogeneous settings (Chen et al., 2022). Accordingly, the joint policy factorizes as $\pi^{\text{tot}}(a_t|s_t) = \prod_{i=1}^{N} \pi(a_{i,t}|o_i(s_t))$, where π denotes an observation-based local policy shared by all humanoids. Let $s_{i,t}^{\text{prop}}$ and $g_{i,t}$ denote the proprioceptive state and the local task goal of humanoid i, respectively. Then, the local observation is given by $o_i(s_t) = (s_{i,t}^{\text{prop}}, g_{i,t})$. Note that the task goal $g_{i,t}$ may vary across humanoids but has the same dimensionality for all agents.

Multi-humanoid loco-manipulation can be viewed as coordinating the individual loco-manipulation capabilities of multiple humanoids toward a shared task. Recent work on single-humanoid whole-body control has shown that complex loco-manipulation behaviors can be composed from reusable motor skills rather than learned entirely from scratch (Kuang et al., 2025). Following this intuition, we build the decentralized local policy π upon a set of pre-trained primitive skills. For each humanoid embodiment, we construct a reusable skill library consisting of task-agnostic primitive skills (see Fig. 2(b)), including `Walking`, `Reaching`, and `Squatting`, which provide basic motor capabilities for synthesizing diverse whole-body behaviors. These skills are pre-trained using dense rewards to capture low-level control objectives such as whole-body stability and energy efficiency, thereby reducing the requirement to explicitly encode these objectives in the reward design for multi-humanoid coordination. Brief descriptions of these primitive skills are provided in Appendix C. Note that the skill library can be extended with additional primitive skills to support more complex coordination behaviors.

Following Kuang et al. (2025), we adopt a hierarchical policy architecture to blend primitive skills. For each coordination task, we first select C relevant primitive skills from the skill library according to the task requirements. For instance, in a box-pushing scenario, the coordinated behavior mainly relies on the `Walking` and `Reaching` skills, whereas `Squatting` contributes little. Accordingly, we select only `Walking` and `Reaching` as the low-level skills for our hierarchical policy. Let μ^c_{low} denote the c-th primitive skill. At each time instant t, given the proprioceptive state $s^{\text{prop}}_{i,t}$ and the skill-dependent goal $g^c_{i,t} \in \mathbb{R}^{d_c}$ as inputs, μ^c_{low} outputs a deterministic local action for humanoid i, given by $u^c_{i,t} = \mu^c_{\text{low}}(s^{\text{prop}}_{i,t}, g^c_{i,t}) \in \mathbb{R}^d$, where $u^c_{i,t}$ denotes the target joint positions and d denotes the number of DoFs of the humanoid. Note that the dimension of $g^c_{i,t}$ varies across skills.

After the primitive skills are selected, a decentralized high-level policy is built to blend these skills, which is illustrated in Fig. 2(c). Specifically, at each time instant t, humanoid i's high-level policy μ_{high} takes the local observation $o_i(s_t)$ as its input, and outputs a raw joint goal vector $\tilde{g}_{i,t}$ and a raw joint weight vector $\tilde{w}_{i,t}$, i.e., $[\tilde{g}^T_{i,t}, \tilde{w}^T_{i,t}]^T = \mu_{\text{high}}(o_i(s_t))$, where $\tilde{g}_{i,t} = [(\tilde{g}^1_{i,t})^T, \dots, (\tilde{g}^C_{i,t})^T]^T$ with $\tilde{g}^c_{i,t} \in \mathbb{R}^{d_c}$ and $\tilde{w}_{i,t} = [(\tilde{w}^1_{i,t})^T, \dots, (\tilde{w}^C_{i,t})^T]^T$ with $\tilde{w}^c_{i,t} = [\tilde{w}^c_{i,t,1}, \dots, \tilde{w}^c_{i,t,d}]^T \in \mathbb{R}^d$ for any c. To provide reasonable goal information, a skill-specific clipping function is applied to $\tilde{g}_{i,t}$ to produce the final goal vector $g^c_{i,t}$, which is then used by each primitive skill μ^c_{low} to generate its action $u^c_{i,t}$. To avoid exaggerated values when blending these actions, each raw weight vector $\tilde{w}^c_{i,t}$ is processed through a pointwise Softmax function. Let $w^c_{i,t} = [w^c_{i,t,1}, \dots, w^c_{i,t,d}]^T$ denote the processed weight vector of the c-th primitive skill, where each $w^c_{i,t,k}$ satisfies $w^c_{i,t,k} = \text{softmax}_c([\tilde{w}^1_{i,t,k}, \dots, \tilde{w}^C_{i,t,k}]^T)$. Based on the weight vectors $\{w^c_{i,t}\}^C_{c=1}$, all actions $\{u^c_{i,t}\}^C_{c=1}$ are blended to generate the final local action $u_{i,t} = [u_{i,t,1}, \dots, u_{i,t,d}]^T$, where each $u_{i,t,k}$ satisfies

$$u_{i,t,k} = \sum_{c=1}^{C} w^c_{i,t,k} \cdot u^c_{i,t,k}, \qquad \forall k = 1, \dots, d. \tag{1}$$

Note that both the primitive skills and the high-level policy require only local information. Therefore, the resulting hierarchical whole-body control policy is decentralized.

To facilitate RL-based policy training, we parameterize the high-level policy as $\mu_{\text{high},\xi}$ and model the resulting decentralized policy as a d-dimensional diagonal Gaussian distribution: $\pi_\theta(\cdot|o_i(s_t)) = \mathcal{N}(u_{i,t}, \text{diag}(\sigma^2))$, where $\theta = [\xi^T, \sigma^T]^T$ denotes the learnable parameters and $\sigma \in \mathbb{R}^d$ is the standard deviation vector. Denote the corresponding parameterized joint policy by π^{tot}_θ. Note that the primitive skills remain fixed during training, and their parameters are therefore excluded from θ. This shifts the learning problem from acquiring low-level whole-body motor skills from scratch to learning high-level skill composition and inter-agent coordination over a fixed set of reusable skills, thereby structuring policy learning and making the MARL problem more tractable.

4.2 Permutation-Based Data Augmentation

RL-based humanoid control typically relies on large-scale parallel simulation (Makoviychuk et al., 2021). Nevertheless, the simulation cost increases substantially in the multi-humanoid setting due to the larger number of agents and more complex inter-agent interactions. To improve data efficiency, we exploit the permutation properties of H-MGs (Chen et al., 2022) and incorporate a permutation-based data augmentation strategy into policy learning.

Denote $o_t = o(s_t)$. For any permutation $M \in \mathcal{M}$, we can apply a permutation transformation to the collected data sample (s_t, o_t, a_t) to obtain the permuted counterpart, i.e.,

$$(s_t, o_t, a_t) \xrightarrow{M} (Ms_t, Mo_t, Ma_t). \tag{2}$$

For notational convenience, let $V_\theta(s) = V_{\pi_\theta^{\text{tot}}}(s)$, based on which the action-value function is defined as $Q_\theta(s,a) = R_{\text{team}}(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[V_\theta(s')]$ (Sutton & Barto, 2018). Under the shared decentralized policy, we consider a joint-policy surrogate objective $L_{\theta_k}(\theta)$ following the TRPO/PPO formulation (Schulman et al., 2015; 2017). Its gradient can be expressed in the following importance-sampling form:

$$\nabla_\theta L_{\theta_k}(\theta) = \mathbb{E}_{s_t \sim \rho_{\pi_{\theta_k}^{\text{tot}}}, a_t \sim \pi_{\theta_k}^{\text{tot}}} \left[\frac{\pi_\theta^{\text{tot}}(a_t|s_t)}{\pi_{\theta_k}^{\text{tot}}(a_t|s_t)} A_{\theta_k}(s_t, a_t) \nabla_\theta \log \pi_\theta^{\text{tot}}(a_t|s_t) \right], \tag{3}$$

where $\pi_{\theta_k}^{\text{tot}}$ is a rollout policy, and $\rho_{\pi_\theta^{\text{tot}}}(s) = \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\theta(s_t = s)$ is the (unnormalized) visitation measure of s, where $\mathbb{P}_\theta(s_t = s)$ denotes the marginal state distribution w.r.t. the policy π_θ^{tot}, and $A_\theta(s_t, a_t) = Q_\theta(s_t, a_t) - V_\theta(s_t)$ is the advantage function. Denote the policy gradient induced by the data sample (s_t, o_t, a_t) by

$$f(s_t, a_t, o_t; \theta, \theta_k) = \frac{\pi_\theta^{\text{tot}}(a_t|s_t)}{\pi_{\theta_k}^{\text{tot}}(a_t|s_t)} A_{\theta_k}(s_t, a_t) \nabla_\theta \log \pi_\theta^{\text{tot}}(a_t|s_t). \tag{4}$$

Lemma 1. *For an H-MG in which all agents share a team reward R_{team} and an observation-based local policy π_θ, both the value function V_θ and the action-value function Q_θ are permutation invariant, i.e., $V_\theta(s) = V_\theta(Ms)$ and $Q_\theta(s,a) = Q_\theta(Ms, Ma)$, $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\forall M \in \mathcal{M}$.*

Theorem 2. *Under the setting of H-MGs in Lemma 1, for any rollout joint policy $\pi_{\theta_k}^{\text{tot}}$, any collected sample (s_t, o_t, a_t) and its permuted counterpart (Ms_t, Mo_t, Ma_t) yield the same policy gradient, i.e.,*

$$f(s_t, a_t, o_t; \theta, \theta_k) = f(Ms_t, Ma_t, Mo_t; \theta, \theta_k), \quad \forall M \in \mathcal{M}. \tag{5}$$

The proofs of Lemma 1 and Theorem 2 can be found in Appendix D. Theorem 2 clearly indicates that augmenting a batch of collected samples with their permuted counterparts leaves the policy-gradient direction unchanged, thereby providing theoretical justification for incorporating permutation-based data augmentation into our MARL framework.

Note that prior work has provided a preliminary investigation of permutation-based experience augmentation in off-policy MARL (Ye et al., 2022), where the theoretical analysis only focuses on the validity of permuted transitions and Q-function estimation, while empirical evaluation is limited to low-dimensional multi-agent benchmarks. In contrast, we extend permutation-based augmentation to on-policy MARL for high-dimensional multi-humanoid coordination and establish its theoretical validity at the policy-gradient level under the H-MG formulation. In addition, our method is orthogonal to other data augmentation techniques, such as symmetry-based augmentation (Lin et al., 2020; Mittal et al., 2024). We leave the integration of these approaches for future work.

4.3 MASkillBlender

We finally integrate the decentralized skill-blended control policy and permutation-based data augmentation into the overall MASkillBlender framework. Specifically, we adopt MAPPO (Yu et al., 2022) to train the shared decentralized high-level policy. During the training process, a centralized critic $V_\phi(s_t)$, parameterized by ϕ, takes the global state as input to approximate the value function $V_\theta(s_t)$ associated with the current joint policy π_θ^{tot}. The centralized critic is used only for policy training, while each humanoid executes the shared policy independently using its local observation.

During policy optimization, for each minibatch $\mathcal{B}$ sampled from the rollout buffer, we select one or more permutations from $\mathcal{M}$. For each permutation M, we augment $\mathcal{B}$ with the permuted counterpart (Ms_t, Mo_t, Ma_t) of each sample (s_t, o_t, a_t). Since $V_\theta(s)$ is permutation invariant by Lemma 1, the centralized critic V_ϕ is trained using the augmented minibatch, such that $V_\phi(s_t)$ and $V_\phi(Ms_t)$ are optimized to approximate the same return. The overall training procedure of MASkillBlender is summarized in Appendix E.



Figure 3: Training curves of all methods on three Unitree H1 tasks.

5 EXPERIMENTS

We conduct experiments to evaluate the effectiveness, robustness, and generality of MASkillBlender across different multi-humanoid coordination tasks. The experimental setup is introduced in Sec. 5.1. Main results are presented in Sec. 5.2, followed by ablation studies in Sec. 5.3, policy robustness evaluations in Sec. 5.4, and cross-simulator transfer in Sec. 5.5. Additional experiments are provided in Sec. 5.6.

5.1 EXPERIMENTAL SETUP

Tasks. Inspired by classic multi-robot coordination problems, we consider three multi-humanoid coordination tasks, namely `Carry` (Gasbarrone et al., 2024), `Push` (Tang et al., 2024), and `Move` (Hu et al., 2022). All tasks are implemented in SkillBench (Kuang et al., 2025), which is implemented in NVIDIA Isaac Gym (Makoviychuk et al., 2021). In `Carry`, two humanoids collaboratively transport a large box to a target position in the air. In `Push`, two humanoids collaboratively push a heavy box toward a target position on the ground plane. Both tasks involve contact-rich interactions, where the humanoids are coupled through the motion of the shared object. In `Move`, three humanoids move toward their assigned target positions while avoiding inter-humanoid collisions. Detailed task specifications, including observations, actions and reward functions, are provided in Appendices F.1 and F.2. For all three tasks, the local policy observation is deliberately kept compact, consisting only of proprioceptive information and the task-relevant information necessary for local decision making. In addition, each task is specified with only a few reward terms, all of which are directly associated with task completion. This reduces the need for extensive reward engineering. As shown in Fig. 1, each task supports two humanoid models, namely the 19-DoF Unitree H1 and the 21-DoF Unitree G1. Detailed specifications of these two humanoid models are provided in Appendix F.3.

Baselines and Implementation Details. We compare our method with SkillBlender (Kuang et al., 2025) and TeamHOI (Lionar & Lee, 2026). SkillBlender is originally designed for single-humanoid whole-body loco-manipulation. We adapt it to the multi-humanoid setting by using a centralized high-level policy that takes the joint observations of all humanoids as input and jointly predicts their skill goals and blending weights. Accordingly, the adapted SkillBlender can be regarded as a centralized counterpart to our method. Note that SkillBlender employs the same pre-trained primitive skills as our method in all tasks. TeamHOI is a reference-based method for cooperative multi-humanoid object transport. It trains a decentralized Transformer policy with PPO and employs a masked Adversarial Motion Prior (AMP) constructed from single-human reference motions to regularize whole-body behaviors during coordination. For fair comparison, we keep the task environments and task-level information consistent across methods whenever possible, while retaining the method-specific inputs required by each baseline. We provide the implementation details of our method and two baselines in Appendix F.4.

Evaluation Metrics. To comprehensively evaluate the learned policies, our metrics include the average episodic return, denoted by Return (↑), and a set of task-specific evaluation metrics. For all methods, Return is computed using only the task rewards defined in our benchmark environments, excluding auxiliary rewards used during training. For both `Carry` and `Push`, the task metric is the

Task	Metric	MASkillBlender	SkillBlender	TeamHOI
Carry	Return (↑)	**67.928 ± 1.563**	65.855 ± 8.343	67.870 ± 9.935
	$\text{Error}_{\text{box}}$ (↓)	**0.016 ± 0.008**	0.025 ± 0.050	0.029 ± 0.102
	Success (↑)	**100.0%**	98.0%	96.0%
Push	Return (↑)	**61.137 ± 5.400**	57.550 ± 5.011	57.991 ± 5.024
	$\text{Error}_{\text{box}}$ (↓)	**0.116 ± 0.063**	0.138 ± 0.069	0.136 ± 0.075
	Success (↑)	**86.0%**	84.0%	78.0%
Move	Return (↑)	**90.622 ± 1.022**	71.190 ± 3.211	16.723 ± 0.256
	$\text{Error}_{\text{pos}}$ (↓)	**0.010 ± 0.003**	0.043 ± 0.012	2.755 ± 0.329
	$\text{Error}_{\text{head}}$ (↓)	22.109 ± 6.552	223.139 ± 84.019	**3.822 ± 3.142**
	Success (↑)	**100.0%**	10.0%	0.0%

Table 1: Quantitative comparison of MASkillBlender with SkillBlender and TeamHOI on the Unitree H1 tasks.

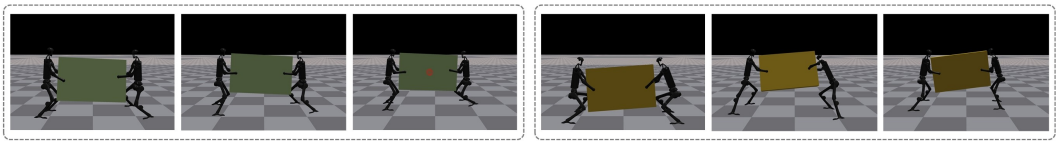


Figure 4: Qualitative comparison of MASkillBlender and TeamHOI on the Unitree H1 Carry task. Representative key frames from both methods are shown to illustrate the whole-body coordination behaviors during box lifting and transportation.

distance between the final box position and the target position, denoted by $\text{Error}_{\text{box}}$ (↓). For Move, we report the maximum distance between the humanoids' final root positions and their respective targets, as well as the cumulative heading error over the episode, denoted by $\text{Error}_{\text{pos}}$ (↓) and $\text{Error}_{\text{head}}$ (↓), respectively. Based on these metrics, we further define task-specific success criteria and report the corresponding success rate, denoted by Success (↑). For Carry, an episode is considered successful if $\text{Error}_{\text{box}} \leq 0.1$. For Push, an episode is considered successful if $\text{Error}_{\text{box}} \leq 0.2$. For Move, an episode is considered successful if both $\text{Error}_{\text{pos}} \leq 0.03$ and no inter-humanoid collision occurs during the episode.

5.2 Main Results

We evaluate all methods on the Unitree H1 using two independent trials per task, with the smoothed learning curves shown in Fig. 3. MASkillBlender achieves the highest final returns across all three tasks. TeamHOI shows large variation on Carry and Push and fails to learn Move. SkillBlender remains competitive on Carry, but shows reduced performance on Push and a substantial degradation on Move. This task-dependent gap suggests that directly extending single-humanoid skill blending to joint multi-humanoid control does not consistently provide effective coordination across tasks. For quantitative evaluation, we perform 50 rollouts with the best policy learned by each method, and the results are reported in Table 1. MASkillBlender achieves the highest success rate on all three tasks. The lower success rate on Push is mainly due to the heavy box and complex contact interactions among the humanoids, the box, and the ground. TeamHOI obtains a lower $\text{Error}_{\text{head}}$ on Move because it learns to orient toward the target but fails to reach the target position. Fig. 4 further provides a qualitative comparison on the Carry task. MASkillBlender achieves coordinated box transportation, whereas TeamHOI tends to move the box toward the target before establishing coordinated locomotion.

5.3 Ablation Studies

To assess the contribution of the key components in MASkillBlender, we conduct ablation studies on permutation-based data augmentation and primitive skill selection. We first remove permutation-based data augmentation while keeping all other settings unchanged. As shown in Fig. 3, removing the augmentation generally slows down learning and leads to lower final returns. The quantitative results in Table 4 of Appendix F.5 show consistent performance drops across all three tasks. These results demonstrate the effectiveness of our data augmentation strategy. We further examine the

contribution of the primitive skills by removing `Squatting` from `Carry` and `Walking` from `Push`. As shown in Fig. 5 of Appendix F.5, both variants exhibit degraded learning performance, highlighting the importance of task-relevant primitive skills for coordinated whole-body control.

5.4 Policy Robustness under Deployment Variations

To assess deployment robustness, we further evaluate MASkillBlender under several practical variations. Detailed experimental settings are provided in Appendix F.6. First, we introduce localization errors, asynchronous control delays, and inter-robot communication delays into the task environments. Without additional training or fine-tuning, the learned policies maintain effective coordination across all three tasks. Then, we evaluate long-horizon execution from asymmetric initial configurations on `Carry` and `Push`. By decomposing the tasks and reusing pre-trained primitive skills, the learned coordination policies can be directly extended to these more complex initial configurations without additional policy training. Note that a brief discussion of the deployment advantages of our task-decomposition strategy is also provided in Appendix F.6, where its reduced reliance on inter-robot communication helps mitigate the impact of communication delays.

5.5 Sim2Sim Transfer

To evaluate Sim2Sim transfer, we deploy MASkillBlender from Isaac Gym to MuJoCo (Todorov et al., 2012) on `Carry` and `Move`. For `Carry`, the policy is briefly trained under broader domain randomization before deployment, while `Move` is transferred directly without additional training. MASkillBlender preserves strong task performance on both tasks after transfer to MuJoCo, demonstrating effective cross-simulator transfer. Detailed settings and results are provided in Appendix F.7.

5.6 Additional Multi-Humanoid Coordination Experiments

To evaluate the versatility of MASkillBlender, we conduct additional experiments on different object-manipulation scenarios and humanoid embodiments. We introduce a two-humanoid box-exchange task on Unitree H1 to extend the evaluation beyond the manipulation of large shared objects. The results show that MASkillBlender can support coordinated manipulation involving smaller objects. We also evaluate MASkillBlender on the 21-DoF Unitree G1 across `Carry`, `Push`, and `Move`. The results suggest that MASkillBlender generalizes across different humanoid embodiments without substantial modifications. Detailed settings and results are provided in Appendices F.8 and F.9.

6 Conclusion and Discussion

In this paper, we presented MASkillBlender, a general MARL framework for decentralized multi-humanoid coordination. By building a shared decentralized high-level policy over reusable pre-trained primitive skills, MASkillBlender focuses downstream learning on high-level skill composition and inter-agent coordination using local observations and task-level rewards. We further incorporated permutation-based data augmentation for homogeneous humanoid teams and analyzed its effect on policy optimization under the H-MG formulation. Extensive simulated experiments demonstrate the versatility of MASkillBlender across diverse multi-humanoid coordination tasks and humanoid embodiments, as well as its robustness under deployment variations and cross-simulator transfer.

Despite these encouraging results, several limitations remain. First, the current formulation assumes homogeneous humanoid teams and relies on a shared local policy. Extending the framework to heterogeneous teams with different morphologies, capabilities, or functional roles remains an important direction for future work. Second, the high-level policies are still trained and evaluated primarily in simulation with state-based observations. Although we evaluate deployment variations and Sim2Sim transfer, real-world multi-humanoid deployment still requires addressing onboard perception, state estimation, contact-rich sim-to-real transfer, and hardware safety. Third, the performance of MASkillBlender depends on the quality and coverage of the primitive skill library. Incorporating richer or automatically discovered skills may further broaden the range of multi-humanoid tasks that can be solved. Overall, we hope that this work can serve as a useful step toward general learning frameworks for coordinated humanoid robot teams.

AI USE STATEMENT

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments, implement methods, assist with translation, and interpret results. We have not used generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, or clean and reformat datasets. Generating synthetic datasets, proposing or refining hypotheses, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the manuscript to improve readability, summarize and analyze existing literature, source and search for information, brainstorm ideas, create or edit software code, create or modify scientific figures and images, and format references. We have reviewed all AI-assisted work. AI-assisted methodological and experimental suggestions were reviewed by the authors before adoption, and AI-assisted code was manually inspected and validated through execution and experimental testing. Literature summaries, sourced information, and references were checked against the original sources. All AI-assisted text, translations, result interpretations, and scientific figures or images were reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C. Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Jacky Baltes, Ilham Akbar, and Saeed Saeedvand. Cooperative dual-actor proximal policy optimization algorithm for multi-robot complex control task. *Advanced Engineering Informatics*, 63: 102960, 2025. doi: 10.1016/j.aei.2024.102960.

Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025.

Dingyang Chen, Yile Li, and Qi Zhang. Communication-efficient actor-critic methods for homogeneous markov games. In *International Conference on Learning Representations*, 2022.

Hongjin Chen, Wei Zhang, Pengfei Li, Shihao Ma, Ke Ma, Yujie Jin, Zijun Xu, Xiaohui Wang, Yupeng Zheng, Zining Wang, Jieru Zhao, Yilun Chen, and Wenchao Ding. Rhythm: Learning interactive whole-body control for dual humanoids. *arXiv preprint arXiv:2603.02856*, 2026a.

Xuanyu Chen, Mohan Liu, Dengchen Mei, Zhihao Gu, Haitian Zhang, Kaimin Mao, Haiyue Zhu, Shijun Yan, and Lin Wang. Physics-guided biomechanical gait adaptation for humanoid locomotion on extreme sloped terrains. *arXiv preprint arXiv:2607.07830*, 2026b.

Zixuan Chen, Xialin He, Yen-Jen Wang, Qiayuan Liao, Yanjie Ze, Zhongyu Li, S Shankar Sastry, Jiajun Wu, Koushil Sreenath, Saurabh Gupta, et al. Learning smooth humanoid locomotion through lipschitz-constrained policies. *arXiv preprint arXiv:2410.11825*, 2024.

Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

Jeremy Dao, Helei Duan, and Alan Fern. Sim-to-real learning for humanoid box loco-manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16930–16936. IEEE, 2024.

Zhaohan Feng, Ruiqi Xue, Lei Yuan, Yang Yu, Ning Ding, Meiqin Liu, Bingzhao Gao, Jian Sun, Xinhu Zheng, and Gang Wang. Multi-agent embodied AI: Advances and future directions. *Science China Information Sciences*, 69(5):151202, 2026. doi: 10.1007/s11432-025-4820-4.

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

Jiawei Gao, Ziqin Wang, Zeqi Xiao, Jingbo Wang, Tai Wang, Jinkun Cao, Xiaolin Hu, Si Liu, Jifeng Dai, and Jiangmiao Pang. Coohoi: Learning cooperative human-object interaction with manipulated object dynamics. In *Advances in Neural Information Processing Systems*, volume 37, pp. 79741–79763, 2024.

Greta Gasbarrone, Nicola Scianca, Leonardo Lanari, and Giuseppe Oriolo. A decentralized cooperative transportation scheme for humanoid robots. In *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, pp. 381–387. IEEE, 2024.

Xinyang Gu, Yen-Jen Wang, Xiang Zhu, Chengming Shi, Yanjiang Guo, Yichen Liu, and Jianyu Chen. Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning. *arXiv preprint arXiv:2408.14472*, 2024.

Zhaoyuan Gu, Junheng Li, Wenlan Shen, Wenhao Yu, Zhaoming Xie, Stephen McCrory, Xianyi Cheng, Abdulaziz Shamsah, Robert Griffin, C Karen Liu, et al. Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning. *arXiv preprint arXiv:2501.02116*, 2025.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International conference on autonomous agents and multiagent systems*, pp. 66–83. Springer, 2017.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024a.

Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024b.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

Yifan Hu, Junjie Fu, and Guanghui Wen. Decentralized robust collision-avoidance for cooperative multirobot systems: A gaussian process-based control barrier function approach. *IEEE Transactions on Control of Network Systems*, 10(2):706–717, 2022.

Yifan Hu, Junjie Fu, and Guanghui Wen. Graph soft actor–critic reinforcement learning for large-scale distributed multirobot coordination. *IEEE transactions on neural networks and learning systems*, 2023.

Yifan Hu, Junjie Fu, Guanghui Wen, Yuezu Lv, and Wei Ren. Distributed entropy-regularized multi-agent reinforcement learning with policy consensus. *Automatica*, 164:111652, 2024.

Humanoid-SkillBlender. Skillblender. `https://github.com/Humanoid-SkillBlender/SkillBlender`, 2025. GitHub repository.

Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

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

Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv preprint arXiv:2502.20396*, 2025.

Yijiong Lin, Jiancong Huang, Matthieu Zimmer, Yisheng Guan, Juan Rojas, and Paul Weng. Invariant transform experience replay: Data augmentation for deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4):6615–6622, 2020.

Stefan Lionar and Gim Hee Lee. Teamhoi: Learning a unified policy for cooperative human-object interactions with any team size. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37121–37132, 2026.

Zuhong Liu, Junhao Ge, Minhao Xiong, Jiahao Gu, Bowei Tang, Wei Jing, and Siheng Chen. It takes two: Learning interactive whole-body control between humanoid robots. *arXiv preprint arXiv:2510.10206*, 2025.

Zhengyi Luo, Jiashun Wang, Kangni Liu, Haotian Zhang, Chen Tessler, Jingbo Wang, Ye Yuan, Jinkun Cao, Zihui Lin, Fengyi Wang, Jessica Hodgins, and Kris Kitani. Smplolympics: Sports environments for physically simulated humanoids. *arXiv preprint arXiv:2407.00187*, 2024.

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

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

Mayank Mittal, Nikita Rudin, Victor Klemm, Arthur Allshire, and Marco Hutter. Symmetry considerations for learning task symmetric robot policies. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7433–7439. IEEE, 2024.

Yunyang Mo, Yi Gu, Yangchen Zhou, Hanyang Cao, and Renjing Xu. Learning context-aware motion priors for humanoid control. *arXiv preprint arXiv:2608.03234*, 2026.

Ilija Radosavovic, Bike Zhang, Baifeng Shi, Jathushan Rajasegaran, Sarthak Kamat, Trevor Darrell, Koushil Sreenath, and Jitendra Malik. Humanoid locomotion as next token prediction. *arXiv preprint arXiv:2402.19469*, 2024.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

Wandong Sun, Luying Feng, Baoshi Cao, Yang Liu, Yaochu Jin, and Zongwu Xie. Ulc: A unified and fine-grained controller for humanoid loco-manipulation. *arXiv preprint arXiv:2507.06905*, 2025.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Zili Tang, Yuming Feng, and Meng Guo. Collaborative planar pushing of polytopic objects with multiple robots in complex scenes. *arXiv preprint arXiv:2405.07908*, 2024.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Unitree Robotics. unitree_ros. `https://github.com/unitreerobotics/unitree_ros`. GitHub repository.

Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C Karen Liu, Rocky Duan, and Guanya Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. *arXiv preprint arXiv:2509.26633*, 2025.

Zhenhui Ye, Yining Chen, Xiaohong Jiang, Guanghua Song, Bowei Yang, and Sheng Fan. Improving sample efficiency in multi-agent actor-critic methods. *Applied Intelligence*, 52:3691–3704, 2022. doi: 10.1007/s10489-021-02554-5.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Yanjie Ze, Zixuan Chen, Joao Pedro Araújo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. Twist: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025.

Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. Wococo: Learning whole-body humanoid control with sequential contacts. *arXiv preprint arXiv:2406.06005*, 2024.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. Simulation and retargeting of complex multi-character interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

Appendix

A Notations

Denote $\mathcal{I} = \{1, \ldots, N\}$. Denote $\mathbb{R}$ and $\mathbb{R}^m$ as the spaces of real numbers and m-dimensional real vectors, respectively. Let $x = (x_1, \ldots, x_N)$ be an ordered list. For any ordered list with N elements, we define a permutation $M = [m_1, \ldots, m_N]$ such that $m_i \neq m_j$ for all $i \neq j$ and $m_k \in \mathcal{I}$ for all $k \in \mathcal{I}$. Let $\mathcal{M}$ denote the set of all such permutations. The permuted list of x under M is given by $Mx = (x_{m_1}, \ldots, x_{m_N})$. For any vector $z = [z_1, \ldots, z_m]^T \in \mathbb{R}^m$, we define the pointwise Softmax mapping $\mathrm{softmax}_c : \mathbb{R}^m \to \mathbb{R}$ by $\mathrm{softmax}_c(z) = \frac{e^{z_c}}{\sum_{i=1}^{m} e^{z_i}}$, where $c \in \{1, \ldots, m\}$.

B An Example of H-MGs

As shown in Fig. 1, two humanoid robots collaboratively push a box to a target position. The target position is randomly initialized at the start of each episode and remains fixed for the duration of the episode. This task can be cast as an H-MG. Specifically, the local state s_i consists of humanoid i's proprioceptive state (e.g., its root and joint states), the root state of the box, and the task goal (i.e., the target box position). Since the humanoids are homogeneous, it holds that $\mathcal{S}_1 = \mathcal{S}_2$. Furthermore, the global state s can be reconstructed by concatenating s_1 and s_2. The local action is the target joint position of each humanoid, and thus $\mathcal{A}_1 = \mathcal{A}_2$. This task contains a single team reward, which is composed of two sub-rewards. The first sub-reward depends on the distance between the box position and the target position. For the second sub-reward, we define two contact points on the box and assign each humanoid a unique contact point based on distance. The second sub-reward then depends on the distances between the humanoid's wrists and its assigned contact point. Since both sub-rewards are invariant to the ordering of the humanoids, the overall reward function is permutation invariant. The transition function is also permutation invariant because the underlying dynamics depend only on physics. Finally, the observation of each humanoid i consists of its proprioceptive state, the relative position between its target and the box, and the relative positions between its wrists and its assigned contact point. Therefore, the joint observation is permutation-preserving.

C Primitive Skill Specifications

In our experiments, we consider three primitive skills, which are `Walking`, `Reaching`, and `Squatting`. Note that the primitive skills for the Unitree H1 are directly obtained from the project Humanoid-SkillBlender (2025), whereas those for the Unitree G1 are trained in-house.

- `Walking`: This skill enables the humanoid to track commanded base motions for locomotion. Its goal is specified by the target planar linear velocity together with the target yaw angular velocity.

- `Reaching`: This skill supports upper-body manipulation by driving the humanoid's two wrists toward designated 3D targets while keeping the body grounded. Its goal is defined as the relative offset from each wrist to its corresponding target.

- `Squatting`: This skill modulates the humanoid's vertical posture by lowering or raising the body to a desired base height. Its goal is given by the relative height error between the current root and the target root height.

More detailed specifications of these primitive skills can be found in Appendix C of Kuang et al. (2025).

D Proofs

Proof of Lemma 1. Recall that the joint reward function $R(s, a) = (R_1(s, a), \ldots, R_N(s, a))$ in H-MGs is permutation preserving as stated in Definition 1. Since all agents share a team reward R_{team}, it follows that $R_{\text{team}}(Ms, Ma) = R_{\text{team}}(s, a)$, which implies that R_{team} is permutation invariant. In addition, due to the permutation-preserving property of the joint observation, we have $o_j(Ms) = o_{m_j}(s)$ for all $j \in \mathcal{I}$ and any permutation $M = [m_1, \ldots, m_N] \in \mathcal{M}$. Therefore, for any

Algorithm 1 MASkillBlender

for iteration $p = 1, 2, \ldots, P$ **do**
  for environment $l = 1, 2, \ldots, L$ **in parallel do**
    All agents $i \in \mathcal{I}$ execute the decentralized policy $\pi_{\theta_{\text{old}}}$ in environment l for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$ from collected trajectories
  end for
  for epoch $k = 1, 2, \ldots, K$ **do**
    Sample a minibatch $\mathcal{B}$ from the rollout buffer
    Apply permutation-based data augmentation to $\mathcal{B}$
    Update policy parameters θ and value parameters ϕ with MAPPO
  end for
  $\theta_{\text{old}} \leftarrow \theta$
end for

parameter θ, it holds

$$\pi_\theta^{\text{tot}}(a|s) = \prod_{i=1}^{N} \pi_\theta(a_i|o_i(s)) = \prod_{j=1}^{N} \pi_\theta(a_{m_j}|o_{m_j}(s)) = \pi_\theta^{\text{tot}}(Ma|Ms), \tag{6}$$

where the second equality follows from $\pi_\theta(a_i|o_i(s)) = \pi_\theta(a_{m_j}|o_{m_j}(s))$ for $i = m_j$ and all $i \in \mathcal{I}$. Equation (6) indicates that the joint policy π_θ^{tot} is permutation invariant. As a result, given the permutation invariance of the transition function, the reward function, and the joint policy, we have $V_\theta(s) = V_\theta(Ms)$ for all $M \in \mathcal{M}$. Furthermore, $Q_\theta(s, a) = Q_\theta(Ms, Ma)$ follows from the permutation invariance of the reward function and the value function.

Proof of Theorem 2. Recall that π_θ^{tot} is permutation invariant for any parameter θ and permutation M, which implies

$$\frac{\pi_\theta^{\text{tot}}(a_t|s_t)}{\pi_{\theta_k}^{\text{tot}}(a_t|s_t)} = \frac{\pi_\theta^{\text{tot}}(Ma_t|Ms_t)}{\pi_{\theta_k}^{\text{tot}}(Ma_t|Ms_t)}, \tag{7}$$

$$\log \pi_\theta^{\text{tot}}(a_t|s_t) = \log \pi_\theta^{\text{tot}}(Ma_t|Ms_t). \tag{8}$$

Denote $\theta = [\theta_1, \ldots, \theta_p]^T \in \mathbb{R}^p$, and let e_i be the i-th standard basis vector in $\mathbb{R}^p$. Then, for any $i \in \{1, \ldots, p\}$, it holds that

$$\begin{aligned}
&\frac{\partial}{\partial \theta_i} \log \pi_\theta^{\text{tot}}(a_t|s_t) \\
=&\lim_{h \to 0} \frac{\log \pi_{\theta + he_i}^{\text{tot}}(a_t|s_t) - \log \pi_\theta^{\text{tot}}(a_t|s_t)}{h} \\
=&\lim_{h \to 0} \frac{\log \pi_{\theta + he_i}^{\text{tot}}(Ma_t|Ms_t) - \log \pi_\theta^{\text{tot}}(Ma_t|Ms_t)}{h} \\
=&\frac{\partial}{\partial \theta_i} \log \pi_\theta^{\text{tot}}(Ma_t|Ms_t),
\end{aligned} \tag{9}$$

where $h \in \mathbb{R}$. Since (9) holds for all $i \in \{1, \ldots, p\}$, we obtain

$$\nabla_\theta \log \pi_\theta^{\text{tot}}(a_t|s_t) = \nabla_\theta \log \pi_\theta^{\text{tot}}(Ma_t|Ms_t). \tag{10}$$

In addition, since both $V_\theta(s)$ and $Q_\theta(s, a)$ are permutation invariant, it follows that

$$A_\theta(s_t, a_t) = A_\theta(Ms_t, Ma_t). \tag{11}$$

Combining (7), (10) and (11), we obtain (5), which completes the proof.

E TRAINING PROCEDURE OF MASKILLBLENDER

The detailed training procedure of MASkillBlender is summarized in Algorithm 1.

F Experiment Details

F.1 Observations and Actions

The local observation of each humanoid i consists of the proprioceptive state $s_{i,t}^{\text{prop}}$ and the local goal $g_{i,t}$. Across all tasks, $s_{i,t}^{\text{prop}}$ includes the joint angles, joint velocities, previous actions, base angular velocity, and base Euler angles. Let d_{task} denote the dimensionality of $g_{i,t}$, and recall that d is the number of DoFs of the humanoid. The resulting local observation dimension for each humanoid is $3d + 6 + d_{\text{task}}$.

Recall that the dimensionality of the skill-dependent goal vector for primitive skill c is d_c. Therefore, the output dimensionality of the decentralized high-level policy is $\sum_{c=1}^{C}(d + d_c)$. Following existing works (Kuang et al., 2025; He et al., 2024b;a; Zhang et al., 2024), both the hierarchical policy and the low-level skills output whole-body target joint positions for the humanoid, which are then converted into torques via a PD controller.

F.2 Task Specifications

• `Carry`: The objective is to coordinate two humanoids to collaboratively carry a large box to a target position. The goal of each humanoid consists of the relative displacement between the box position and the target position, as well as the relative displacements between the humanoid's wrists and its assigned box-side reference point. The selected primitive skills are `Walking`, `Reaching`, and `Squatting`. The reward is defined as

$$R_{\text{team}}(s, a) = \omega_1 \cdot e^{-4\|p_{\text{box}} - p_{\text{goal}}\|_1} + \omega_2 \cdot e^{-4\|q_{\text{wb}}\|_\infty}, \tag{12}$$

where $p_{\text{box}} \in \mathbb{R}^3$ and $p_{\text{goal}} \in \mathbb{R}^3$ are the current box position and the target box position, respectively, $q_{\text{wb}} \in \mathbb{R}^{12}$ is the concatenated wrist-to-box-reference relative offsets of the two humanoids, and $\{\omega_i\}_{i=1}^2$ are weighting coefficients.

• `Push`: The objective is to coordinate two humanoids to collaboratively push a large and heavy box to a target position on the plane. The goal of each humanoid consists of the relative displacement between the box position and the target position, as well as the relative displacements between the humanoid's wrists and its assigned box-side reference point. The selected primitive skills are `Walking` and `Reaching`. The reward is defined as

$$R_{\text{team}}(s, a) = \omega_1 \cdot e^{-4\|p_{\text{box}} - p_{\text{goal}}\|_1} + \omega_2 \cdot e^{-4\|q_{\text{wb}}\|_\infty}, \tag{13}$$

where $p_{\text{box}} \in \mathbb{R}^3$ and $p_{\text{goal}} \in \mathbb{R}^3$ are the current box position and the target box position, respectively, $q_{\text{wb}} \in \mathbb{R}^{12}$ is the concatenated relative offsets from the wrists of the two humanoids to their assigned box-side reference points, and $\{\omega_i\}_{i=1}^2$ are weighting coefficients.

• `Move`: The objective is to coordinate three humanoids to move to their assigned target positions while avoiding collisions with the other humanoids. The goal of each humanoid consists of the relative displacement between its current root position and its target position, as well as the relative planar position of its closest neighboring humanoid. The selected primitive skill is `Walking`. The reward is defined as

$$R_{\text{team}}(s, a) = \omega_1 \cdot e^{-4\bar{d}_{\text{tar}}} + \omega_2 \cdot e^{-4\bar{d}_{\text{col}}} + \omega_3 \cdot e^{-4\bar{e}_{\text{head}}}, \tag{14}$$

where

$$\bar{d}_{\text{tar}} = \frac{1}{9}\sum_{i=1}^{3} \|p_i - p_i^\star\|_1, \quad \bar{d}_{\text{col}} = \sum_{i=1}^{3} \max(0, 0.8 - \|c_i\|_2), \quad \bar{e}_{\text{head}} = \frac{1}{3}\sum_{i=1}^{3} e_i^{\text{head}}. \tag{15}$$

Here, $p_i \in \mathbb{R}^3$ and $p_i^\star \in \mathbb{R}^3$ denote the current root position and the target position of humanoid i, respectively, $c_i \in \mathbb{R}^2$ denotes the relative planar position of the closest neighboring humanoid to humanoid i, e_i^{head} denotes the heading error of humanoid i w.r.t. its target direction, and $\{\omega_i\}_{i=1}^3$ are weighting coefficients.

A comparison of three tasks is summarized in Table 2. It is worth noting that each task relies on only a few reward terms, all of which are directly related to task completion. This design avoids extensive reward engineering.

Task	Number of Humanoids	Number of Reward Terms	Selected Primitive Skills		
			`Walking`	`Reaching`	`Squatting`
`Carry`	2	2	✓	✓	✓
`Push`	2	2	✓	✓	
`Move`	3	3	✓		

Table 2: Task settings.

Task	Actor hidden dimensions	Critic hidden dimensions
`Carry`	[512, 256, 128]	[768, 256, 128]
`Push`	[512, 256, 128]	[1024, 512, 128]
`Move`	[512, 256, 128]	[768, 256, 128]

Table 3: Network hyperparameters of MASkillBlender for three Unitree H1 tasks.

F.3 Humanoid Model Details

We consider two Unitree humanoid models in our experiments, namely the 19-DoF Unitree H1 and the 21-DoF Unitree G1. Note that the hand links are excluded for both models, and only the main upper-body and lower-body links relevant to locomotion and whole-body manipulation are retained. The corresponding embodiment details are summarized below:

• **19-DoF Unitree H1**: This humanoid embodiment is the same as that used in Kuang et al. (2025). It includes two 3-DoF shoulders, two 1-DoF elbows, one 1-DoF torso joint, two 3-DoF hips, two 1-DoF knees, and two 1-DoF ankle joints.

• **21-DoF Unitree G1**: This humanoid embodiment is a modified 21-DoF Unitree G1 model derived from the official 23-DoF configuration file (Unitree Robotics). It includes two 3-DoF shoulders, two 1-DoF elbows, one 1-DoF torso joint, two 3-DoF hips, two 1-DoF knees, and two 2-DoF ankle joints. In addition, we attach a virtual joint named `wrist_target_joint` to each arm to conveniently query the wrist position for task construction. These virtual joints are introduced only for kinematic querying and are not treated as actuated joints in control.

F.4 Implementation Details

MASkillBlender and SkillBlender. For all primitive skills, the policies are parameterized as end-to-end MLPs with three hidden layers of sizes [512, 256, 128]. Note that these primitive skill policies are pre-trained and kept fixed when training the high-level policy. For both methods, both the actor and critic are parameterized as MLPs with three hidden layers. Following Kuang et al. (2025), the critic network takes privileged observations of all humanoids as input, including base linear velocity, body mass, and contact masks. For MASkillBlender, the network hyperparameters for all three tasks with the Unitree H1 are summarized in Table 3. SkillBlender employs a larger actor network with hidden-layer sizes [768, 256, 128] for all tasks. In addition, a larger critic network with hidden-layer sizes [1024, 512, 128] is employed in MASkillBlender for all tasks with the Unitree G1. For the RL training hyperparameters, we set the entropy coefficient to 0.001, the learning rate to 1×10^{-5}, the number of learning epochs to 2, the discount factor to $\gamma = 0.994$, the GAE parameter to $\lambda = 0.9$, and the number of mini-batches to 4. This hyperparameter setting is used for all tasks across all embodiments. For `Carry` and `Push`, we set 4096 parallel environments during training. For `Move`, we set 2048 parallel environments. In all tasks, MASkillBlender uses data generated from all permutations for policy training.

TeamHOI. We adapt TeamHOI to the Unitree H1 while preserving its core reference-based PPO and AMP training paradigm. Since the original TeamHOI is developed with a simplified 28-DoF humanoid, its motion references cannot be directly applied to the 19-DoF H1 used in our tasks. We therefore retarget the selected human motion references to the H1 embodiment using General Motion Retargeting (GMR) (Araujo et al., 2026). The retargeting process accounts for the humanoid's kinematic structure and joint limits, and converts the reference motions into H1-compatible root

Task	Metric	MASkillBlender	MASkillBlender w/o Aug.
Carry	Return (↑)	**67.928 ± 1.563**	45.791 ± 5.684
	$\text{Error}_{\text{box}}$ (↓)	**0.016 ± 0.008**	0.179 ± 0.068
	Success (↑)	**100.0%**	6.0%
Push	Return (↑)	**61.137 ± 5.400**	57.130 ± 9.322
	$\text{Error}_{\text{box}}$ (↓)	**0.116 ± 0.063**	0.147 ± 0.098
	Success (↑)	**86.0%**	76.0%
Move	Return (↑)	**90.622 ± 1.022**	84.460 ± 1.147
	$\text{Error}_{\text{pos}}$ (↓)	**0.010 ± 0.003**	0.023 ± 0.006
	$\text{Error}_{\text{head}}$ (↓)	**22.109 ± 6.552**	73.530 ± 23.160
	Success (↑)	**100.0%**	86.0%

Table 4: Ablation study of permutation-based data augmentation on the Unitree H1 tasks.

poses and joint trajectories for subsequent AMP training. For Carry, we select three pickup motions and two sideways locomotion motions from the motion data provided by TeamHOI and CooHOI, covering the main behaviors required for lifting and transporting the box. For Push, since no suitable pushing motion is available in the original TeamHOI motion set, we select two heavy-object pushing sequences from the CMU Motion Capture Database through AMASS (Mahmood et al., 2019). We further adjust their upper-body posture to obtain higher and more suitable hand-object interaction configurations before retargeting them to H1. For Move, we use locomotion references covering straight walking, stand-to-walk and walk-to-stand transitions, as well as mild left/right turns. Unlike the original TeamHOI, which switches between Full and Masked AMP according to the agent-object distance, our tasks initialize the humanoids directly near the interaction region. Therefore, we use either Full AMP or Masked AMP throughout training for each task. Based on empirical training performance, Carry and Move use Full AMP, while Push uses Masked AMP. We retain the shared decentralized Transformer policy of TeamHOI while adapting its observation interface to our benchmark tasks. For Carry and Push, each humanoid observes its proprioceptive state, task goal, and wrist-to-object interaction geometry, without explicit teammate information. For Move, the local observation additionally includes the relative position of the closest teammate for collision avoidance. Thus, the adapted TeamHOI policy uses the same task-level observation information as MASkillBlender for each task. Each observation component is encoded into a 64-D token and processed by three Transformer encoder layers with two attention heads and a feed-forward dimension of 512. The actor head outputs a 19-D action for the H1 humanoid, and the same architecture is used across all three tasks with task-dependent input dimensions. For task rewards, we replace the original TeamHOI transport-specific reward design with the reward functions defined in our benchmark environments, while retaining the AMP reward for motion regularization. The task reward and AMP reward are combined with equal weights during training. We retain the main PPO hyperparameters of the original implementation, including the learning rate, clipping ratio, value loss coefficient, discount factor, and GAE parameter, while adapting the rollout and optimization schedule to our training framework.

Training. We primarily employ NVIDIA RTX 5090 GPUs for policy training, with each training or inference run using a single GPU. For MASkillBlender on the Unitree H1, a single trial takes approximately 50 hours for Carry, 76 hours for Push, and 35 hours for Move. For the adapted TeamHOI baseline, the corresponding training times are approximately 60, 84, and 55 hours, respectively. Despite the larger number of entities involved in our task environments, all tasks remain trainable on a mid-range workstation equipped with a single NVIDIA RTX 4080 GPU.

F.5 Ablation Studies

Permutation-Based Data Augmentation. The quantitative results of the ablation on permutation-based data augmentation are provided in Table 4.

Primitive Skill Selection. The learning curves of the primitive skill selection ablation are provided in Fig. 5.

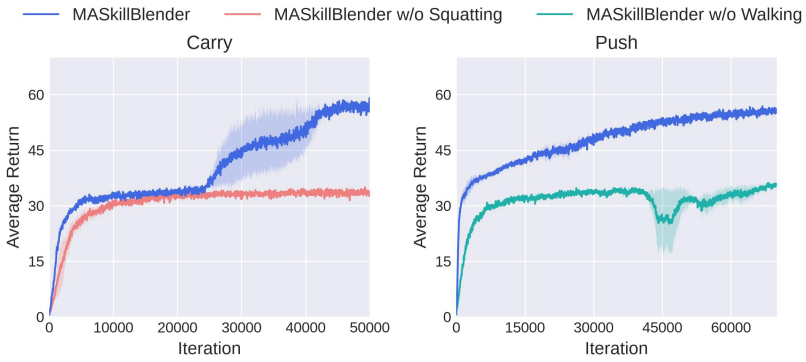


Figure 5: Primitive skill ablation on the Unitree H1. Removing `Squatting` from `Carry` or `Walking` from `Push` substantially degrades learning performance.

Task	Metric	Nominal	Constrained
`Carry`	Return (↑)	**67.928 ± 1.563**	63.462 ± 11.406
	$\text{Error}_{\text{box}}$ (↓)	**0.016 ± 0.008**	0.073 ± 0.201
	Success (↑)	**100.0%**	92.0%
`Push`	Return (↑)	**61.137 ± 5.400**	52.744 ± 12.604
	$\text{Error}_{\text{box}}$ (↓)	**0.116 ± 0.063**	0.158 ± 0.089
	Success (↑)	**86.0%**	74.0%
`Move`	Return (↑)	**90.622 ± 1.022**	90.377 ± 2.289
	$\text{Error}_{\text{pos}}$ (↓)	**0.010 ± 0.003**	0.014 ± 0.004
	$\text{Error}_{\text{head}}$ (↓)	22.109 ± 6.552	**21.834 ± 10.831**
	Success (↑)	**100.0%**	98.0%

Table 5: Robustness evaluation of MASkillBlender under deployment noise and delays on the Unitree H1 tasks.

F.6 Policy Robustness under Deployment Variations

To further evaluate MASkillBlender under more realistic deployment conditions, we consider two complementary settings beyond the nominal task environments. First, we construct constrained variants of `Carry`, `Push`, and `Move` by introducing localization errors, asynchronous control delays, and, for `Move`, inter-robot communication delays. Second, we evaluate long-horizon execution from asymmetric initial configurations on `Carry` and `Push`, where existing locomotion and whole-body skills are composed to approach the interaction region before the learned coordination policy takes over.

Robustness to Deployment Noise and Delays. For `Carry`, each humanoid independently experiences a control delay of 0, 10, or 20 ms (Zhang et al., 2024), which is sampled at the beginning of each episode and kept fixed throughout the rollout. To model localization errors in real-world perception and state estimation, we perturb the estimated box position by up to ± 0.02 m along each axis and its orientation by up to ± 0.1 rad. In addition, an independent positional error of up to ± 0.02 m is added to the positions of both wrists of each humanoid to account for inaccuracies in end-effector localization.

For `Push`, we apply the same independent control delays of 0, 10, or 20 ms. Since the box remains on the ground throughout the task, we use relatively smaller perturbations for its estimated pose, with position errors of up to ± 0.02 m in the horizontal plane and ± 0.01 m vertically, and orientation errors of up to ± 0.01 rad for roll and pitch and ± 0.05 rad for yaw. Each humanoid also receives an independent wrist-position error of up to ± 0.02 m.

For `Move`, each humanoid independently experiences a control delay of 0, 10, or 20 ms. To model localization uncertainty, its estimated base position is perturbed by up to ± 0.02 m in the horizontal

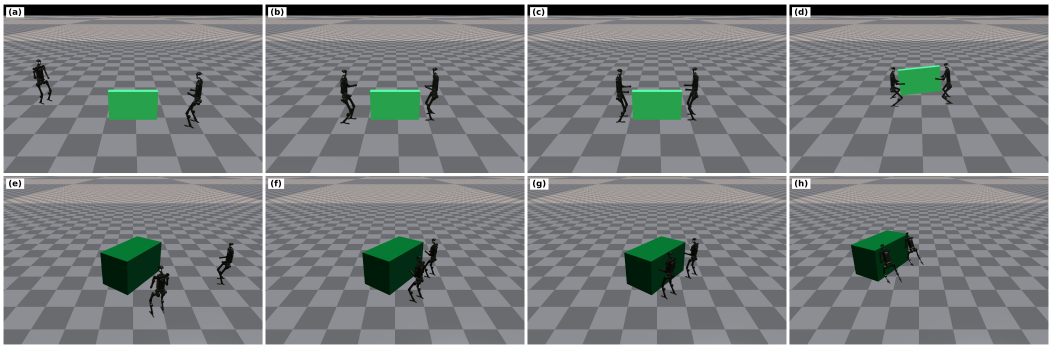


Figure 6: Long-horizon deployment from asymmetric initial configurations on the Unitree H1. The top row shows `Carry` and the bottom row shows `Push`. From left to right, the snapshots illustrate the asymmetric initial configuration, one humanoid arriving while the other is still approaching, both humanoids reaching the predefined task poses, and the final coordinated task execution.

plane and ± 0.01 m vertically. We further introduce inter-robot communication delays of 0, 10, 20, or 30 ms for the exchanged position estimates. This models communication latency in decentralized multi-robot systems (Chen et al., 2026a).

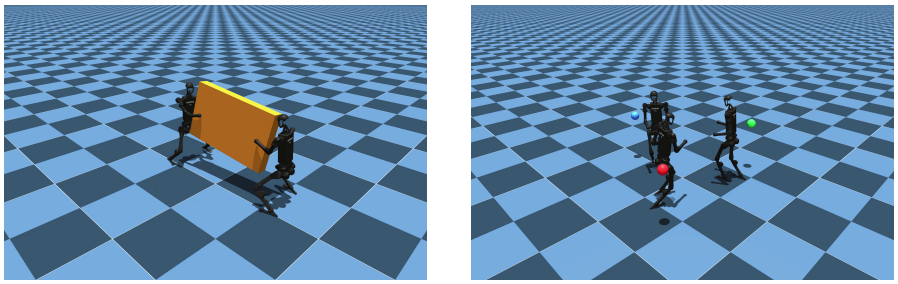
We directly evaluate the same policy checkpoint used in the nominal environment for 50 rollouts without any additional training or fine-tuning. As shown in Table 5, MASkillBlender maintains effective coordination across all three tasks under the introduced deployment noise and delays, which demonstrates its robustness to these deployment variations. This robustness may be partly attributed to the domain randomization already incorporated during training. We also observe a more noticeable drop in success rate on `Push`, which could potentially be mitigated by retraining the policy with broader domain randomization covering these deployment variations.

Long-Horizon Deployment from Asymmetric Initial Configurations. To facilitate efficient coordination learning, the humanoids in our benchmark tasks are initialized close to the object and interaction region. Nevertheless, following the idea of task decomposition (Zhang et al., 2024), the learned coordination policies can be directly extended to asymmetric initial configurations without additional training by composing them with existing primitive skills.

For both `Carry` and `Push`, the long-horizon task is divided into three steps. The two humanoids first move from their asymmetric initial positions to predefined poses near the box. If one humanoid arrives earlier, it waits until the other is also in place. Once both humanoids are ready, they switch to the learned task policy to transport or push the box toward the target. The pre-trained `walking` skill is used to approach the box, while the `squatting` skill maintains a stable posture during waiting.

As shown in Fig. 6, this task decomposition allows the learned coordination policies to be directly extended to more complex asymmetric initial configurations. By reusing the pre-trained primitive skills for navigation and maintaining a stable waiting posture, MASkillBlender can accomplish long-horizon cooperative behaviors without additional policy training.

Existing multi-humanoid methods such as TeamHOI (Lionar & Lee, 2026) learn a unified policy to cover both approaching the object and the subsequent cooperative transportation. To enable behavioral transitions within a single policy, each humanoid explicitly observes the states of the other humanoids in addition to its own state and task-related information. Such peer information is useful during the approaching stage, since each humanoid needs to reason about whether its teammates have reached the interaction region and when coordinated manipulation should begin. However, once cooperative transportation starts, coordination can be achieved primarily from the humanoid's own state and the state of the shared object, without continuously observing explicit teammate states (Gao et al., 2024). Keeping peer-state observations throughout the entire unified policy therefore introduces an additional dependency during deployment. In real multi-robot systems, these states are typically obtained through inter-robot communication and relative localization, and can consequently be affected by communication delays and asynchronous state updates (Chen et al., 2026a). In contrast, our task decomposition only requires lightweight communication to determine



(a) `Carry` (b) `Move`

Figure 7: Sim2Sim deployment of MASkillBlender in MuJoCo on (a) `Carry` and (b) `Move`.

Task	Metric	MuJoCo
`Carry`	Return (↑)	66.787 ± 2.221
	$\text{Error}_{\text{box}}$ (↓)	0.026 ± 0.015
`Move`	Return (↑)	86.323 ± 1.430
	$\text{Error}_{\text{pos}}$ (↓)	0.013 ± 0.005
	$\text{Error}_{\text{head}}$ (↓)	38.151 ± 30.906

Table 6: Sim2Sim evaluation of MASkillBlender in MuJoCo on the Unitree H1 tasks.

when both humanoids are ready to enter the cooperative stage. After the learned `Carry` or `Push` policy is activated, each humanoid executes the decentralized policy without explicit teammate-state communication. This design reduces the dependence of cooperative manipulation on continuous inter-robot communication and makes the execution less sensitive to communication latency.

F.7 Sim2Sim Transfer

To evaluate the cross-simulator transferability of MASkillBlender, we deploy the learned H1 policies from Isaac Gym to MuJoCo. We consider `Carry` and `Move`. For `Carry`, due to its contact-sensitive object manipulation, we initialize from the policy learned in Isaac Gym and continue training for a short period under a broader domain-randomization setting. For `Move`, we directly deploy the Isaac Gym policy in MuJoCo without additional training.

We evaluate both tasks over 100 episodes in MuJoCo, with the results reported in Table 6. For `Carry`, the two humanoids are able to carry the box through most episodes. In 92% of the sampled episodes, the final box-position error is no greater than 0.1 m. For `Move`, MASkillBlender achieves a success rate of 99%. Fig. 7 shows successful coordinated behaviors in MuJoCo. These results demonstrate that the learned policies can be effectively transferred across simulators.

F.8 Two-Humanoid Box Exchange

We also evaluate MASkillBlender on the manipulation of smaller objects. Specifically, we design a two-humanoid box-exchange task, where two humanoids cooperatively exchange two small boxes across a table. As shown in Fig. 8(a), two humanoids stand on opposite sides of a table, with one box initially placed on each side. The task is divided into two stages. In the first stage, each humanoid moves its assigned box to a predefined exchange position near the center of the table. Once both boxes reach their exchange positions, the task switches to the second stage. The humanoids then exchange their assigned boxes and move them to the corresponding final target positions, which are the original positions of the opposite boxes.

For each humanoid, the task-related goal consists of the relative displacement between its currently assigned box and the corresponding target position, the relative displacements between its wrists and the assigned box, the current task stage, and the relative heights of the wrists and elbows with respect to the table. The selected primitive skills are `Walking` and `Reaching`. The task reward is composed of box-goal tracking, wrist-to-box proximity, and an arm-height term:

$$R_{\text{team}}(s, a) = \omega_1 R_{\text{box}} + \omega_2 R_{\text{wrist}} + \omega_3 R_{\text{height}}, \tag{16}$$

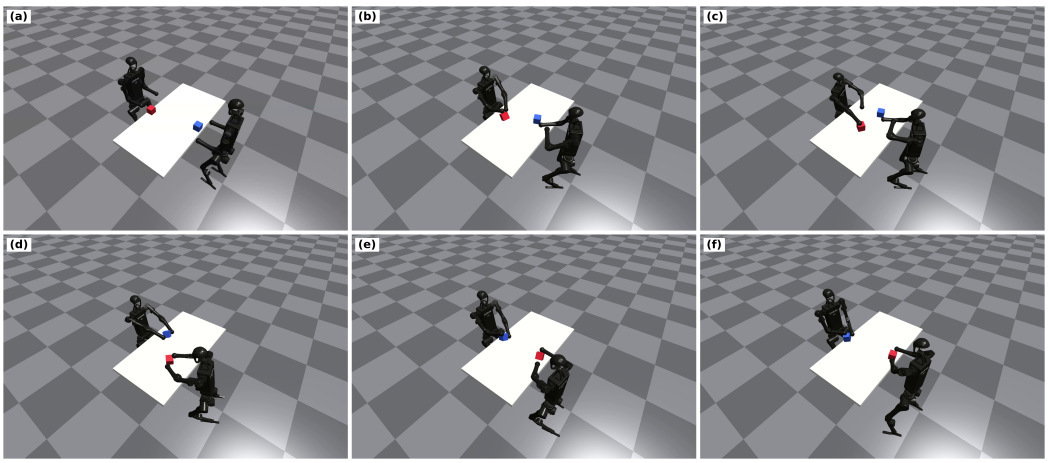


Figure 8: Two-humanoid box-exchange task on the Unitree H1. The snapshots illustrate the execution sequence from the initial configuration to the final box exchange. From (a) to (f), the two humanoids first move their assigned boxes to the exchange region and then deliver the exchanged boxes to their final target positions.

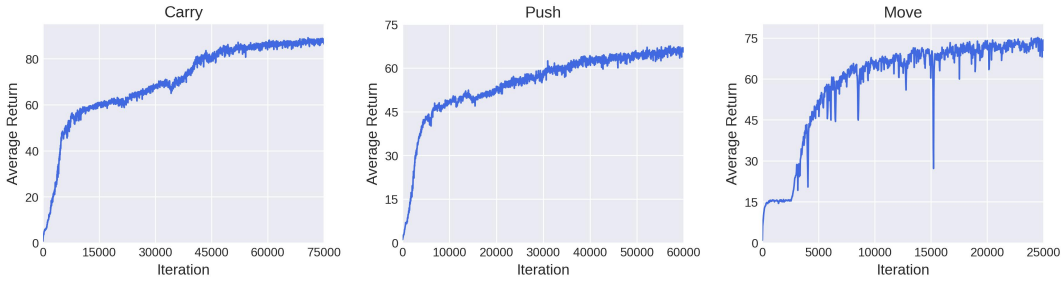


Figure 9: Training curves of MASkillBlender on three Unitree G1 tasks.

where R_{box} encourages the currently assigned boxes to approach their stage-dependent target positions, R_{wrist} encourages the humanoids' wrists to remain close to the corresponding boxes, and R_{height} encourages the wrists and elbows to remain above the table during manipulation. The box and wrist objectives are updated after the stage transition according to the exchanged box assignments. As shown in Fig. 8, the learned policy successfully coordinates the two humanoids throughout both stages of the task. The humanoids first move their assigned boxes to the exchange region and then switch responsibilities to deliver the exchanged boxes to their final target positions.

F.9 Additional Experiments on Unitree G1

We finally evaluate MASkillBlender on the 21-DoF Unitree G1 to examine its applicability across different humanoid embodiments. As illustrated in Fig. 1, we consider the same three coordination tasks. The overall task definitions and learning framework remain unchanged. We only make minor empirical adjustments to the reward coefficients and training iterations for the G1 embodiment.

We provide the training curves on the three G1 tasks in Fig. 9 and the quantitative results in Table 7. The results show that MASkillBlender successfully learns coordinated behaviors across all three tasks. On `Carry` and `Push`, the box-position errors remain comparable to those obtained on the H1 embodiment. The G1 policy also learns coordinated movement on `Move`, although larger position and heading errors are observed. This difference may be related to the performance of the G1 walking primitive. Overall, these results suggest that MASkillBlender generalizes well across different humanoid embodiments and multi-humanoid coordination tasks, while maintaining strong performance without requiring substantial modifications to the overall framework.

Task	Metric	MASkillBlender
`Carry`	Return (↑)	91.366 ± 5.339
	$\text{Error}_{\text{box}}$ (↓)	0.025 ± 0.043
`Push`	Return (↑)	68.851 ± 13.408
	$\text{Error}_{\text{box}}$ (↓)	0.113 ± 0.108
`Move`	Return (↑)	81.853 ± 2.115
	$\text{Error}_{\text{pos}}$ (↓)	0.039 ± 0.006
	$\text{Error}_{\text{head}}$ (↓)	54.420 ± 25.458

Table 7: Performance of MASkillBlender on the Unitree G1 tasks.