Supplementary Material

In this section, we categorize our discussion into three main parts. Initially, we delve into the sources and processing methods for motion data used in training. Following that, we explore how observations are constructed and how reward functions are established. Finally, we describe the implementation details including physics simulation and hyperparameters in network training.

A   Sources and Processing of Motion Data

We collected a total of four types of basic reference motion data, including 9 motions related to walking, 5 related to picking up, 4 related to carrying, and 5 related to putting down. All these data are in SMPL format and recorded at 30 fps over 139 frames. They all originate from the ACCAD subset of the AMASS [4] dataset. Additionally, to ensure the stability of cooperative tasks involving multiple individuals, we included data for sidewalk and reverse carry motions. The sidewalk data comes from the CMU subset within AMASS, while reverse carry data was scarce. Therefore, we created reverse carry data by reversing the process of the carry data. In total, we used 26 motion data as references. Additionally, we performed a simple visualization of the extended objects as in Figure 1, which sampled from dataset [1].

ArmChair   HighStool   Table   Sofa
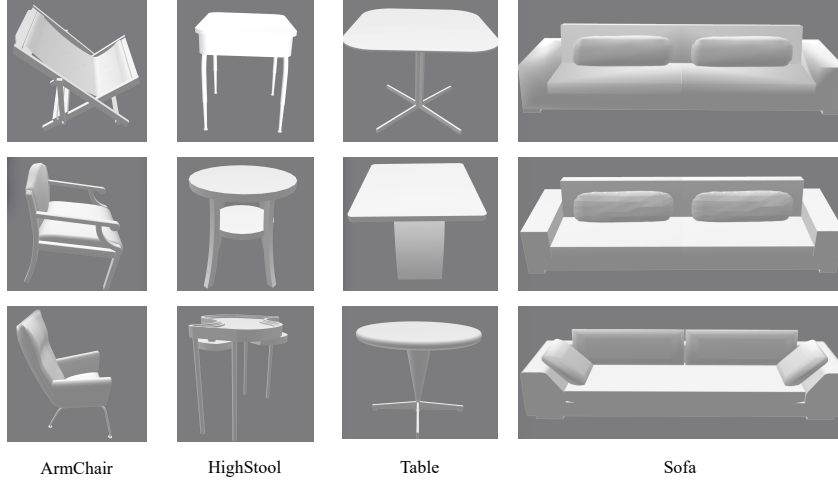
Figure 1: Some visualization of daily-life objects.

B   Task Formulation

We formulate our approach as goal-conditioned reinforcement learning. At each discrete step t, the policy $\pi\left(a_t \mid \mathbf{s}_t, \mathbf{g}_t\right)$ generates an action $\mathbf{a}_t$, based on the current state $\mathbf{s}_t$ and a goal-specific feature $\mathbf{g}_t$. Following this action, the environment transitions into a subsequent state, and the agent receives a reward r_t. An episode concludes either after reaching a predetermined length or if conditions for early termination (ET) are met. Further details are provided below.

B.1   Task Observation

The observational for the task is divided into two primary elements: the state feature $\mathbf{s}$, which encapsulates the character's bodily configuration, and the goal feature $\mathbf{g}$, which pertains to tasks involving object manipulation.

The state feature $\mathbf{s}$ is constituted by a 225-dimensional vector, encompassing:

- Height of the root: 1 dimension.

1

28 • Rotation of the root: 6 dimensions.

29 • Linear and angular velocity of the root: 6 dimensions.

30 • Position of local joints: 42 dimensions.

31 • Rotations of local joints: 84 dimensions.

32 • Linear and angular velocity of local joints: 84 dimensions.

33 While the root height is measured in the global reference frame, all other components are defined in
34 the frame local to the character. Rotations follow a 6-dimensional representation for continuity [10].
35 The simulated character aligns with [8, 7, 2, 6], featuring 12 internally movable joints and a total of
36 28 degrees of freedom.

37 The goal feature **g** comprises a 75-dimensional vector, including:

38 • Position of the object: 3 dimensions.

39 • Rotation of the object: 6 dimensions.

40 • Dynamics of the object, which cover the bounding box position, along with linear and
41 angular velocities: 33 dimensions.

42 • Target location: 3 dimensions.

43 • Target orientation: 6 dimensions.

44 • Dimensions of the target's bounding box: 24 dimensions.

45 These are measured in the frame local to the character.

B.2 Reward Functions

47 The agent's reward r_t at each time step t is defined by

$$r_t = w^G r^G \left(\mathbf{s}_t, \mathbf{g}_t, \mathbf{s}_{t+1} \right) + w^S r^S \left(\mathbf{s}_t, \mathbf{s}_{t+1} \right) \tag{1}$$

48 Follow the formulation of the AMP framework [8], the **style reward** r^S is calculated according to
49 the discriminator:

$$r^S \left(\mathbf{s}_t, \mathbf{s}_{t+1} \right) = - \log \left(1 - D \left(\mathbf{s}_t, \mathbf{s}_{t+1} \right) \right) \tag{2}$$

50 And the discriminator is trained by the following objective:

$$\begin{aligned}
\underset{D}{\arg \min} &- \mathbb{E}_{d^{\mathcal{M}}(\mathbf{s}, \mathbf{s}_{t+1})} \left[\log \left(D \left(\mathbf{s}, \mathbf{s}_{t+1} \right) \right) \right] \\
&- \mathbb{E}_{d^{\pi}(\mathbf{s}, \mathbf{s}_{t+1})} \left[\log \left(1 - D \left(\mathbf{s}, \mathbf{s}_{t+1} \right) \right) \right] \\
&+ w^{\mathrm{gp}} \mathbb{E}_{d^{\mathcal{M}}(\mathbf{s}, \mathbf{s}_{t+1})} \left[\left\| \nabla_{\phi} D(\phi) |_{\phi = (\mathbf{s}, \mathbf{s}_{t+1})} \right\|^2 \right]
\end{aligned} \tag{3}$$

51 The **task reward** function r^G is generally segmented into three components, as in Equation (4): 1)
52 r^G_{walk}, which encourages the agent to approach the object intended for manipulation. 2) r^G_{held}, which
53 encourages the agent to align the center of its hands with the center of the box. 3) r^G_{target}, which
54 encourages the agent to transport the object to the specified destination.

$$r^G = 0.2 * r^G_{\mathrm{walk}} + 0.4 * r^G_{\mathrm{held}} + 0.4 * r^G_{\mathrm{target}} \tag{4}$$

55 The walk reward r^G_{walk} is formulated as Equation (5), where x^{standing}_t denotes the position of the
56 standing point near the object, v^* denotes the target velocity, and d^* denotes the desired direction
57 from root to the object.

$$r^G_{\mathrm{walk}} = \begin{cases} 0.4 \exp \left(-0.5 \left\| x^{\mathrm{standing}}_t - x^{\mathrm{root}}_t \right\|^2 \right) + \\ 0.4 \exp \left(-2.0 \left\| v^* - d^{\mathrm{root}}_t \cdot \dot{x}^{\mathrm{root}}_t \right\|^2 \right) + \\ 0.2 \left\| d^* \cdot d^{\mathrm{root}}_t \right\|^2, & \left\| x^*_t - x^{\mathrm{root}}_t \right\| > 0.2m \\ 1.0, & \mathrm{otherwise} \end{cases} \tag{5}$$

2

58 The held reward r_{held}^G is formulated in Equation (6), where x_t^{hand} denotes the center of the agent's two
59 hands and h_t is the position of the object helding point.

$$r_{\text{held}}^G = \exp\left(-5.0\|x_t^{\text{hand}} - h_t\|^2\right) \tag{6}$$

60 The target reward r_{target}^G consist of two parts, r_{carry} and r_{face}, as described in Equation (7).

$$r_{\text{target}}^G = 0.5 * r_{\text{carry}} + 0.5 * r_{\text{face}}. \tag{7}$$

61 The face reward r_{face} guides the agent to walk either forwards or backward. As shown in Equation (8),
62 this is achieved by comparing the agent's velocity direction with its orientation relative to the
63 endpoint's location, thereby cultivating the agent's proficiency in bidirectional locomotion.

$$r_{\text{face}} = \begin{cases} x_t^{\text{face}} \cdot v_t^{\text{face}}, & x_t^{\text{face}} \cdot (d_t - x_t^{\text{root}}) \geq 0 \\ -x_t^{\text{face}} \cdot v_t^{\text{face}}, & x_t^{\text{face}} \cdot (x_t^{\text{root}} - d_t) \geq 0 \end{cases} \tag{8}$$

64 The carry reward r_{carry}, is designed to guarantee that the object is delivered to the precise location
65 at a specific angle. As outlined in Eq. 9, we constrain the agent's movement direction, alongside
66 the proximity to the end destination and the intended angle. Within this context, x_t^* signifies the
67 3D coordinates of the destination, while p_t^* represents the 2D destination coordinates. Similarly,
68 p_t^{root} indicates the 3D position of the agent's root. Furthermore, rot^* designates the object's desired
69 orientation.

$$r_{\text{carry}} = \begin{cases} 0.5 * r_t^{\text{near}} + 0.25 * r_t^{far} + 0.25 * r_t^{\text{dir}}, & \|x_t^* - x_t^{\text{root}}\| > 0.1m \\ 0.5 * r_t^{\text{near}} + 0.25 * r_t^{\text{dir}} + 0.25, & \text{otherwise}, \end{cases} \tag{9}$$

70 where

$$r_t^{\text{far}} = \exp\left(-0.5\left\|p_t^* - p_t^{\text{root}}\right\|^2\right)$$

$$r_t^{\text{near}} = \exp\left(-10.0\left\|x_t^* - x_t^{\text{root}}\right\|^2\right)$$

$$r_t^{\text{dir}} = \left\|\text{rot}^* \cdot \text{rot}_t^{\text{object}}\right\|^2$$

B.3 Reset and early termination condition

72 An episode ends either after reaching a predetermined duration or upon the activation of early
73 termination (ET) conditions. During our experiments, we observed that lower object heights could
74 lead to kicking actions, where the agent tend to kick the object to destination, significantly slowing
75 down the training process. To address this, we assess the object's velocity and height to determine the
76 presence of kicking phenomena. If the height of the object is lower than 0.3m and its velocity in x-y
77 plane is greater than 1m/s, the kicking early termination (KET) condition is triggered. Experimental
78 results show that this strategy significantly stabilize the training process.

C Implementation Details

C.1 Training Details.

81 Adopting the methodology of AMP [8], we develop a low-level controller encompassing both policy
82 and discriminator networks. The policy network is bifurcated into a critic and an actor-network, each
83 initiating with a CNN layer and proceeding to two MLP layers configured with [1024, 1024, 512]
84 units. The discriminator network is similarly structured, featuring two MLP layers with [1024, 1024,
85 512] units. We select PPO [9] as the primary reinforcement learning algorithm, coupled with the
86 Adam optimizer [3] at a learning rate of 2e-5. The only difference between the multi-agent setting and
87 the single-agent setting during training is whether a pre-trained weight is loaded. Our experiments
88 are conducted on the IsaacGym simulator [5] using a single Nvidia GTX 3090Ti GPU. We run 4096
89 parallel environments across 15,000 epochs, which takes approximately 15 hours to complete.

3

C.2 Hyperparameters

Following previous work[8, 2, 6], we use the Isaac Gym simulator [5]. The simulation runs at 60Hz and the control policy runs at 30Hz.

Besides, the hyperparameters we used in the training process is detailed below:

Table 1: Hyperparameters for CooHOI.

Parameter	Value
Number of Environments	4096
w_G Task-Reward Weight	0.5
w_S Style-Reward Weight	0.5
PPO Minibatch Size	16384
AMP Minibatch Size	4096
Horizon Length	32
Learning Rate	$2e-5$
PPO Clip Threshold ϵ	0.2
γ Discount	0.99
GAE (λ)	0.95
T Episode Length	600

D Failure case visualization.

Here, we conducted a visual analysis of the fail cases. First, for the case lacking a stand point, we can clearly see that the agent moves towards the nearest face, even though it is not the shortest edge, which leads to the agent's inability to carry the object. In the second image, in the absence of dynamic input, we observe that the agent stands still, unable even to squat. In the third image, which depicts the scenario without reverse walking, the agent is able to lift the box, but because it cannot learn the backward gait, the two agents end up pushing the box against each other, causing a deadlock.

No stand point **No Dynamic Observation** **No Reverse Walk**
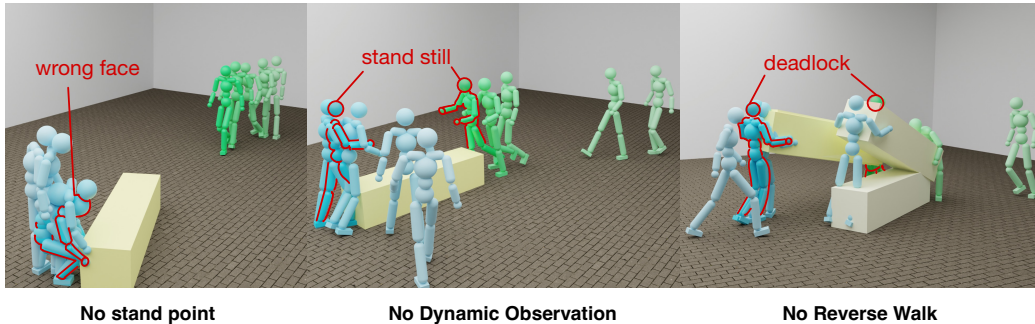
Figure 2: Some visualization on failure cases. "Stand point" means a leading point behind the object to encourage the agent to walk to the object. "Dynamic Observation" means that each agent has its unique input. "Reverse Walk" indicates whether a single agent possesses the skill to walk backward. Without any of the methods we propose, the policy cannot be successfully trained.

References

[1] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11374–11384, 2021.

[2] Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. Synthesizing physical character-scene interactions. *arXiv preprint arXiv:2302.00883*, 2023.

4

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

[4] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[5] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[6] Liang Pan, Jingbo Wang, Buzhen Huang, Junyu Zhang, Haofan Wang, Xu Tang, and Yangang Wang. Synthesizing physically plausible human motions in 3d scenes. *arXiv preprint arXiv:2308.09036*, 2023.

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

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

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

[10] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

5