

TeamHOI: Learning a Unified Policy for Cooperative Human-Object Interactions with Any Team Size

Supplementary Material

1. Training with Various Team Sizes

1.1. Team-Size Advantage Normalization

PPO [4] algorithm computes the advantage term A_t , which measures how much better an action performs relative to the expected return of the policy’s actions in the same state, as estimated by the critic network. The advantages are computed from trajectories collected over a finite time horizon, which determines how far into the future rewards are accumulated. Because their scale can vary across trajectories, the advantages are typically normalized across a batch of trajectories to stabilize training, given as:

$$A_t \leftarrow \frac{A_t - \mu(A)}{\sigma(A) + \epsilon},$$

where $\mu(A)$ and $\sigma(A)$ are the mean and standard deviation computed over the batch.

In our framework, training batches can include data from teams of different sizes, each producing rewards with distinct scales and variances. Normalizing all advantages together across such heterogeneous data can distort their relative magnitudes and influence the accuracy of the policy update signal. Thus, we normalize advantages separately for each team size n :

$$A_t^{(n)} \leftarrow \frac{A_t^{(n)} - \mu_n(A)}{\sigma_n(A) + \epsilon}.$$

As seen in Figure 1, the team-size advantage normalization results in higher task reward.

1.2. Environment Instantiation

We use IsaacGym [3] simulator to train our model. A current limitation of IsaacGym is that each environment in the parallel training must contain the same number of actors, including the humanoid agents and objects. To address this limitation and enable the any team-size unified policy training, we add a dummy ceiling plane in each environment and instantiate a fixed number of agents N . For any environment that requires a smaller team size n , we place the remaining $N - n$ agents on the dummy ceiling. These agents are ignored for the reward calculation, observation states, and gradient computation. Additionally, their PD controllers are disabled. See Figure 2 for an illustration.

2. Reward Functions

Here, we detail all reward components used in the cooperative carrying task, excluding the formation reward r_{form} ,

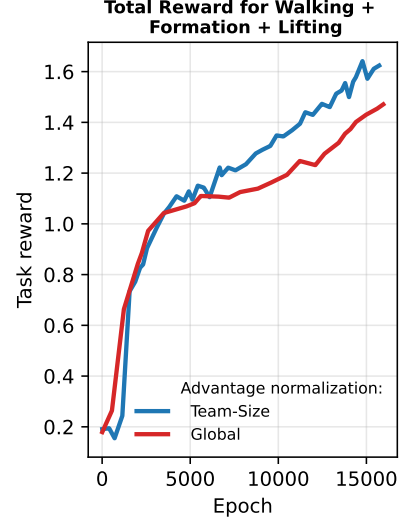


Figure 1. Comparison of task reward curves for models trained with team-size and global advantage normalizations.

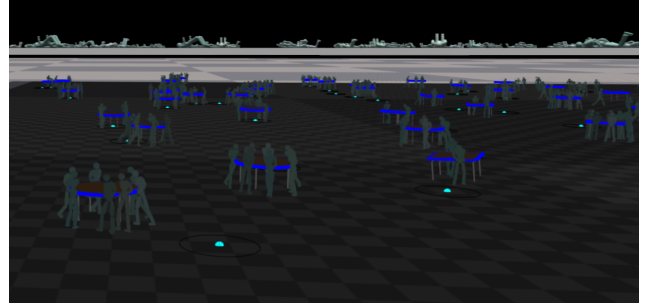


Figure 2. Environment setup in IsaacGym using a dummy ceiling to support flexible team-size training. Extra agents are moved to the ceiling and excluded from observations, rewards, and gradient updates.

angular spread reward r_{ang} , and principal-axes coverage reward r_{cov} already described in the main paper.

2.1. Walking Toward Object

After initialization, each humanoid starts at some distance from the table and is encouraged to walk to the object before the lifting phase. The walking reward is decomposed into three terms that shape the position, velocity, and facing of each agent.

Position: For each agent, we locate the nearest sampled point along the table perimeter, denoted as \mathbf{p}^* , and compute the distance to the agent’s root \mathbf{x}_{root} in x - y plane,

$d = \|\mathbf{x}_{\text{root}} - \mathbf{p}^*\|_2$. The agent is encouraged to stand at a target gap $d_{\text{gap}} = 0.3$ m from \mathbf{p}^* by penalizing the squared deviation $\Delta_{\text{gap}} = (d - d_{\text{gap}})^2$. The position reward is:

$$r_{\text{walk}}^{\text{pos}} = \begin{cases} \exp(-2.0\Delta_{\text{gap}}), & \Delta_{\text{gap}} > 0.04 \text{ m}, \\ 1, & \Delta_{\text{gap}} \leq 0.04 \text{ m}. \end{cases} \quad (1)$$

This term acts as an attractive potential that pulls each agent toward the table. It must be balanced by the formation reward r_{form} to ensure that agents spread out while still converging to their appropriate standing regions before lifting.

Velocity: Let $\mathbf{v} \in \mathbb{R}^2$ be the x, y root velocity. We define a desired walking direction $\mathbf{u}^* \in \mathbb{R}^2$ as the inward unit normal in x - y plane associated with the nearest perimeter point \mathbf{p}^* . The agent’s directional speed s is computed by projecting \mathbf{v} onto \mathbf{u}^* , $s = \mathbf{u}^{*\top} \mathbf{v}$. We then encourage the agent to move toward the table within a preferred speed range from $s_{\text{low}}^* = 1.5$ m/s to $s_{\text{high}}^* = 2.5$ m/s. The deviation from this range is expressed using ReLU functions, $\delta_{\text{vel}} = \max(0, s_{\text{low}}^* - s) + \max(0, s - s_{\text{high}}^*)$. The velocity reward is:

$$r_{\text{walk}}^{\text{vel}} = \begin{cases} 0, & s \leq 0, \\ 1, & \Delta_{\text{gap}} \leq 0.04 \text{ m}, \\ \exp(-2.0\delta_{\text{vel}}^2), & \text{otherwise.} \end{cases} \quad (2)$$

Facing: We compute the agent’s facing direction by extracting the heading component of its root orientation. Let $\mathbf{f} \in \mathbb{R}^2$ be the facing direction in the x - y plane. We define two target facing directions: the inward normal \mathbf{u}^* at \mathbf{p}^* for near-view alignment, and the direction from the agent toward the table center, $\mathbf{c}^* = \frac{\mathbf{p}_{\text{center}} - \mathbf{x}_{\text{root}}}{\|\mathbf{p}_{\text{center}} - \mathbf{x}_{\text{root}}\|_2}$. The facing reward is computed as:

$$r_{\text{walk}}^{\text{face}} = \begin{cases} \max(0, \mathbf{u}^{*\top} \mathbf{f}), & d \leq 1.0 \text{ m}, \\ \max(0, \mathbf{c}^{*\top} \mathbf{f}), & d > 1.0 \text{ m}. \end{cases} \quad (3)$$

2.2. Hand Contact Preparation

After the agents are close to the table with $\|\mathbf{x}_{\text{root}} - \mathbf{p}^*\|_2 \leq 1.0$ m, agents are encouraged to reach the hands towards to the table contact points and maintain a reasonable hand configuration for lifting.

Hand reaching: For each agent, let $\mathbf{h}_j \in \mathbb{R}^3$, $j \in \{\text{L}, \text{R}\}$, be the left and right hand positions, and $\{\mathbf{q}_k\}_{k=1}^{64}$ the 64 candidate contact points. We first find the nearest contact point for each hand and its distance, $d_j^{\text{hand}} = \min_k \|\mathbf{h}_j - \mathbf{q}_k\|_2$. A proximity term encourages both hands to approach the contact point:

$$r_{\text{prox}} = \frac{1}{2} \sum_j \exp(-5.0d_j^{\text{hand}}). \quad (4)$$

In addition, we encourage the hands to reach the lower edge of the table rather than drifting onto the tabletop surface. For each hand $j \in \{\text{L}, \text{R}\}$ with position $\mathbf{h}_j \in \mathbb{R}^3$, let $\mathbf{p}_j^* \in \mathbb{R}^3$ be its nearest sampled perimeter point on the table. We define a contact direction $\hat{\mathbf{v}}_j = \frac{\mathbf{h}_j - \mathbf{p}_j^*}{\|\mathbf{h}_j - \mathbf{p}_j^*\|_2}$. Let $\mathbf{e}_z = (0, 0, 1)^\top$ be the world-up direction. We compute $\cos \theta_j = \hat{\mathbf{v}}_j^\top \mathbf{e}_z$, which measures how much the hand moves upward relative to its associated contact point. The per-hand vertical alignment score is then defined as:

$$r_{\text{above},j} = \begin{cases} \exp(-3.0 \cos \theta_j), & \cos \theta_j > 0, \\ 1, & \cos \theta_j \leq 0. \end{cases} \quad (5)$$

The combined term over both hands is:

$$r_{\text{above}} = \frac{1}{2} (r_{\text{above,L}} + r_{\text{above,R}}). \quad (6)$$

Hand separation: We also encourage a target horizontal separation between the two hands. First, we compute the horizontal separation in the x - y plane, $d_{\text{hand}} = \|(\mathbf{h}_L - \mathbf{h}_R)_{xy}\|_2$. We encourage the hands to remain within a preferred separation interval $d_{\text{low}}^* = 0.4$ m and $d_{\text{high}}^* = 0.6$ m. Deviations from this interval are expressed using ReLU functions, $\delta_{\text{sep}} = \max(0, d_{\text{low}}^* - d_{\text{hand}}) + \max(0, d_{\text{hand}} - d_{\text{high}}^*)$. We then obtain a hand separation reward:

$$r_{\text{sep}} = \exp(-5.0\delta_{\text{sep}}^2). \quad (7)$$

To encourage consistent lifting, we penalize vertical mismatch between the two hands. Let z_L and z_R be their heights, and the reward is defined as:

$$r_{\text{same-z}} = \exp(-20.0(z_L - z_R)^2). \quad (8)$$

Combined reward: The combined hand preparation reward is:

$$r_{\text{hand}} = r_{\text{prox}} \times r_{\text{above}} \times r_{\text{sep}} \times r_{\text{same-z}}, \quad (9)$$

which requires all four terms to be satisfied simultaneously.

2.3. Contact and Lifting

Once the hands are placed near the table edge, additional rewards are activated so that the agents establish contact and lift the table to a desired height.

Contact activation: Let d_j^{hand} be the nearest hand-to-contact distance defined earlier. A per-hand contact score is computed as $\gamma_j = \max\left(0, 1 - \frac{d_j^{\text{hand}}}{0.06 \text{ m}}\right)$. We then define a contact reward as the minimum of the per-hand contact scores across the two hands:

$$r_{\text{contact}} = \min(\gamma_L, \gamma_R), \quad (10)$$

and contact indicator for each hand:

$$m_j = \begin{cases} 1, & d_j^{\text{hand}} < 0.04 \text{ m}, \\ 0, & \text{otherwise,} \end{cases}$$

which is used to gate the subsequent lifting and transport rewards.

Lifting height: After contact is established, the hands should lift the table to a target height. Let \hat{z}_j be the height of the contact point associated with hand j , and the target lifting height $z_{\text{lift}}^* = 0.94 \text{ m}$. We obtain a lifting reward for each hand:

$$\rho_j = \exp(-5.0 |\hat{z}_j - z_{\text{lift}}^*|). \quad (11)$$

Only hands with valid contact contribute. Therefore, the combined lifting reward is given as:

$$r_{\text{lift}} = \frac{1}{2} (m_L \rho_L + m_R \rho_R). \quad (12)$$

2.4. Collective Transport

Transport: Once all agents establish contact with the table using both hands, they are encouraged to move the object toward a target location collectively. Let $\mathbf{x}_{\text{obj}} \in \mathbb{R}^2$ be the x, y table position and $\mathbf{x}_{\text{tar}} \in \mathbb{R}^2$ the target location. We define the transport reward as:

$$r_{\text{transport}} = \begin{cases} \exp(-0.15 \|\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{obj}}\|_2^2), & m_L = m_R = 1 \\ 0, & \text{for all agents,} \\ & \text{otherwise.} \end{cases} \quad (13)$$

Carrying alignment: While not strictly required for transport, we include a carrying alignment reward that encourages at least one agent to face toward the target direction while carrying. We identify this agent as the agent farthest from the target. Let $\mathbf{f} \in \mathbb{R}^2$ be this agent's facing direction in the $x-y$ plane, and the desired transport direction:

$$\mathbf{u}_{\text{tar}} = \frac{\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{obj}}}{\|\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{obj}}\|_2}.$$

We compute the alignment reward:

$$r_{\text{align}} = \begin{cases} \max(0, \mathbf{u}_{\text{tar}}^\top \mathbf{f}), & m_L = m_R = 1 \text{ for all agents} \\ & \text{and } \|\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{obj}}\|_2 \geq 0.5 \text{ m}, \\ 1, & m_L = m_R = 1 \text{ for all agents} \\ & \text{and } \|\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{obj}}\|_2 < 0.5 \text{ m}, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Both $r_{\text{transport}}$ and r_{align} are shared across all agents. During transport, when $m_L = m_R = 1$ for all agents, we set $r_{\text{walk}}^{\text{face}} = 1.0$ so that agents can adjust flexible heading directions while carrying the object collectively.

2.5. Putdown

Once the object reaches the target location, agents must put-down the table and release their hands from the table. Thus, we introduce putdown reward once the object reaches target: $\|\mathbf{x}_{\text{tar}} - \mathbf{x}_{\text{obj}}\|_2 < 0.03 \text{ m}$.

Hand release: Let z_j be the height of hand $j \in \{L, R\}$ and $z_{\text{put}}^* = 0.65 \text{ m}$ the target hand height during putdown. Let d_j^{hand} be the nearest hand-table distance defined earlier. We compute the hand-release reward as:

$$r_{\text{put}}^{\text{release}} = \begin{cases} 1, & d_L^{\text{hand}} > 0.07 \text{ m} \\ & \text{and } d_R^{\text{hand}} > 0.07 \text{ m}, \\ \min_{j \in \{L, R\}} \exp(-5.0 |z_j - z_{\text{put}}^*|), & \text{otherwise.} \end{cases} \quad (15)$$

Zero velocity: During putdown, we also encourage agents to stop moving by applying the following reward:

$$r_{\text{put}}^{\text{vel}} = \exp(-2 \|\mathbf{v}\|_2), \quad (16)$$

where \mathbf{v} is the agent's x, y root velocity.

Combined reward: The final putdown reward is a weighted combination of the hand-release and zero-velocity terms:

$$r_{\text{put}} = 0.8 r_{\text{put}}^{\text{release}} + 0.2 r_{\text{put}}^{\text{vel}}. \quad (17)$$

2.6. Total Task Reward

The task reward for the cooperative carrying task is aggregated as follows:

$$\begin{aligned} r^{\text{task}} = & 0.2 r_{\text{walk}}^{\text{pos}} + 0.4 r_{\text{walk}}^{\text{vel}} + 0.2 \sqrt{r_{\text{walk}}^{\text{face}} r_{\text{ang}}} + 0.6 r_{\text{form}} \\ & + 0.7 (r_{\text{hand}} r_{\text{cov}}) + 0.7 r_{\text{contact}} + 0.7 (r_{\text{lift}} r_{\text{cov}}) \\ & + 1.0 r_{\text{transport}} + 0.4 r_{\text{align}} + 1.0 r_{\text{put}}. \end{aligned} \quad (18)$$

3. Generalized Principal-Axes Coverage Reward

We elaborate the components to obtain the generalized principal-axes coverage reward r_{cov} that supports irregular geometries (including concave shapes such as L-shape) and non-uniform mass distributions.

Center of mass: Let $\mathcal{X} = \{\mathbf{x}_k \in \mathbb{R}^2\}_{k=1}^N$ denote a set of 2D points sampled from the object's $x-y$ plane (e.g., the tabletop surface). Each point may optionally carry a mass weight $w_k > 0$ representing local density. Uniform density corresponds to $w_k = 1$. The planar center of mass is obtained as $\mathbf{c} = \frac{\sum_{k=1}^N w_k \mathbf{x}_k}{\sum_{k=1}^N w_k}$.

Principal-axes: Next, we obtain the principal-axes \mathbf{u}_1 and \mathbf{u}_2 from the eigenvectors of the real and symmetric object's

planar inertia matrix $\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$. Let $\tilde{\mathbf{x}}_k = \mathbf{x}_k - \mathbf{c}$ denote the centered coordinates with components $(\tilde{x}_k, \tilde{y}_k)$. The inertia components are computed as $I_{xx} = \sum_k w_k \tilde{y}_k^2$, $I_{yy} = \sum_k w_k \tilde{x}_k^2$, and $I_{xy} = -\sum_k w_k \tilde{x}_k \tilde{y}_k$. We then compute the eigen decomposition of \mathbf{I} and define \mathbf{u}_1 as the eigenvector associated with the smallest eigenvalue, and \mathbf{u}_2 as the remaining orthonormal eigenvector.

Boundary extents: We compute the boundary extents ℓ_i^+ and ℓ_i^- along each principal axis \mathbf{u}_i in a manner that remains well-defined for irregular and concave geometries. To this end, we compute the convex hull \mathcal{H} of the object boundary in the planar domain. The boundary extents are the maximum distances from the center of mass \mathbf{c} to the convex hull \mathcal{H} along the positive and negative directions of \mathbf{u}_i .

4. Additional Implementation Details

4.1. Training Strategy

Training the unified policy directly with up to eight agents is computationally inefficient due to the long-horizon nature of the cooperative carrying task. We therefore adopt a sequential training with different stages that progressively approaches task completion and increases team size, with early termination triggered whenever any agent falls or table topples. All stages below are trained using a single NVIDIA A100 GPU.

Stage 1: We first train environments instantiated with 1-4 agents to acquire core navigation, contact, and lifting behaviors. At this stage, the transport, alignment, and put-down rewards are disabled (i.e., $r_{\text{transport}} = r_{\text{align}} = r_{\text{put}} = 0$). The full-body discriminator D_{full} is supervised using only forward and sideways walking reference motions, while the masked discriminator D_{mask} additionally receives pickup motions. The target-location token is also masked out during this stage. Training runs for approximately 1.5 days with an episode length of 400 timesteps.

Stage 1+2: Next, we continue training with 2-4 agents to proceed with the coordinated transport and putdown. We enable the remaining task components, including $r_{\text{transport}}$, r_{align} and r_{put} , as well as unmasking the target-location token. Backward walking reference motions are added to supervise D_{mask} to improve locomotion diversity while carrying the object. This stage converges in roughly 5 days with an episode length of 600 timesteps.

Fine-tuning with up to 8 agents: Finally, we fine-tune the unified policy in environments instantiated with 2-8 agents to refine coordination patterns and stabilize collective transport for larger teams. This fine-tuning stage takes about 3 days.

4.2. Training hyperparameters

We train all stages using 1024 parallel environments. For PPO, the minibatch size is set to 16384 when training with

up to four agents, and reduced to 8192 for training with up to eight agents. For both PPO and AMP updates, observations belonging to deactivated agents (i.e., those placed above the ceiling) are excluded from the minibatches. For AMP, the reference-motion minibatch size is 4096, and the policy-observation minibatch is set to 1.5×4096 , but excluding the deactivated agents.

Unless otherwise noted, all remaining hyperparameters follow CooHOI [1]. We list the key values in Table 1 for completeness.

Table 1. Key training hyperparameters in our experiment.

Hyperparameter	Value
Horizon length	32
Optimizer	Adam [2]
Learning rate	2×10^{-5}
Task reward weight	0.5
Style reward weight	0.5
PPO clip threshold (ϵ)	0.2
Discount factor (γ)	0.99
GAE parameter (λ)	0.95

5. CooHOI* Baseline

Architecture: CooHOI* follows the same Transformer-based backbone as our method for both the policy and the critic, but replaces the cross-attention with self-attention layer without incorporating teammate tokens. This design mimics the original CooHOI formulation where cooperation emerges solely from the shared dynamics of the object.

Approach-angle reward: We design an approach-angle reward to guide each agent toward its designated contact point while avoiding collision with the table. Let $\mathbf{p}_{\text{des}} \in \mathbb{R}^2$ be the x, y coordinate of the designated point. We first calculate the normalized x, y direction from the object to agent’s root: $\hat{\mathbf{a}}_o = \frac{\mathbf{x}_{\text{root}} - \mathbf{x}_{\text{obj}}}{\|\mathbf{x}_{\text{root}} - \mathbf{x}_{\text{obj}}\|_2}$, as well as the normalized x, y direction the object to the designated point: $\hat{\mathbf{p}}_o = \frac{\mathbf{p}_{\text{des}} - \mathbf{x}_{\text{obj}}}{\|\mathbf{p}_{\text{des}} - \mathbf{x}_{\text{obj}}\|_2}$. We then calculate the approach-angle reward based on the cosine similarity between the two directions:

$$r_{\text{approach}} = \frac{\hat{\mathbf{a}}_o^\top \hat{\mathbf{p}}_o + 1}{2}. \quad (19)$$

This yields $r_{\text{approach}} = 1$ when the agent is perfectly aligned toward its target contact point ($\theta = 0^\circ$) and $r_{\text{approach}} = 0$ when it faces the opposite direction ($\theta = 180^\circ$), effectively avoiding collision with the table when agents are initialized in random positions.

The aggregated task reward follows the same structure as Equation 18, except that r_{form} , r_{ang} , and r_{cov} are replaced by the approach-angle reward r_{approach} .

Training strategy: We follow a two-stage training procedure as in CooHOI. In the first stage, a single agent is trained to acquire foundational locomotion and manipulation skills, including approaching the table, maneuvering toward the designated contact point, establishing contact, lifting (or tilting) the table to the target height, and subsequently pushing or dragging it toward the goal. To simplify learning, the friction between the table legs and the ground is set to zero and the table mass is reduced by half during this stage. Training runs for approximately 3 days.

Multi-agent cooperation is then introduced in the second stage by resuming from the single-agent checkpoint. Separate models are trained for team sizes of 2, 4, and 8 agents, denoted as CooHOI-2, CooHOI-4, and CooHOI*-8, respectively. CooHOI*-2 converges in roughly 2 days. CooHOI*-4 requires about 6 days, and CooHOI*-8 continues from the 4-agent checkpoint and trains for an additional 5 days. All models are trained with an episode length of 600 timesteps.

Contact point assignment: To reduce inter-agent collisions when spawning large teams, we enforce a consistent geometric mapping between agents and their designated contact points. All contact points are sorted counter-clockwise, starting from the bottom-left corner. After agents are initialized, they are indexed in the same counter-clockwise order, also starting from the bottom-left position. A one-to-one assignment is then performed between agent indices and contact points following this order.

6. More Experimental Results

Unified policy across all team sizes: To complement the results presented in the main paper, Table 2 reports the full performance of our unified policy across all team sizes from 2 to 8 agents under both normal ($1\times$) and heavy ($5\times$) table weights. Beyond the configurations shown in the main paper (2A, 4A, 8A), we additionally include intermediate team sizes (3A, 5A, 6A, 7A), demonstrating that the same decentralized policy generalizes smoothly across all team sizes without retraining. Under the normal-weight setting, our model consistently achieves near-perfect success rates across all team sizes with consistent cooperation.

Under the heavy-weight setting ($5\times$ table mass), the increased load amplifies the need for coordinated force generation. As team size grows, our unified policy facilitates effective cooperation that leverages the additional mechanical advantage provided by larger groups, resulting in steadily improving success rates with more agents.

Zero-shot generalization: We further evaluate our unified policy under unseen table geometries and team sizes, testing whether the coordinated formation and carrying skills acquired during training transfer to new scenarios. We consider both smaller tables (round with 1.40 m diameter, square 1.30 m \times 1.30 m, rectangular 1.60 m \times 0.90 m) and

Table 2. Performance of our unified policy across team sizes under normal ($1\times$) and heavy ($5\times$) table weights.

Normal weight ($1\times$)				
Team size	SR (%) \uparrow	d (m) \downarrow	t_{coop} (%) \uparrow	$ J $ (m/s ³) \downarrow
2	99.1	0.06	95.2	51.0
3	99.4	0.06	98.3	50.5
4	99.2	0.08	96.1	44.7
5	99.5	0.06	97.3	40.6
6	99.3	0.07	95.9	38.0
7	98.6	0.11	93.7	35.7
8	97.5	0.18	90.1	34.2
Heavy weight ($5\times$)				
4	3.5	4.77	90.9	23.4
5	18.2	2.48	79.0	28.3
6	50.1	1.04	79.0	32.0
7	71.6	0.59	81.2	31.8
8	81.1	0.49	81.5	31.7

Table 3. Performance of our unified policy across team sizes for small and large tables. All results are averaged over 10,000 simulation episodes.

Small tables				
Team size	SR (%) \uparrow	d (m) \downarrow	t_{coop} (%) \uparrow	$ J $ (m/s ³) \downarrow
2	93.1	0.37	94.8	63.2
3	97.5	0.14	97.0	64.7
4	98.4	0.12	97.1	55.5
8	96.4	0.23	85.4	45.0
Large tables				
2	71.0	0.85	91.7	53.3
3	82.7	0.46	94.3	52.1
4	85.3	0.51	94.6	48.8
8	84.2	0.93	86.1	45.4
12	80.6	1.14	58.2	45.2
16	74.5	1.41	15.1	46.6

larger tables (round with 2.40 m diameter, square 2.20 m \times 2.20 m, rectangular 3.0 m \times 1.40 m), all with the same mass density as in the main experiment.

As shown in Table 3, our policy maintains coherent cooperation across all configurations despite this distribution shift. For smaller tables, agents occasionally display slightly stronger lift initiation, resulting in modestly higher jerk, but the transport phase remains stable and success rates stay consistently high. For larger tables, agents still maintain synchronized cooperative behaviors. However, the increased mass and longer moment arms make lifting and stabilizing harder. Consequently, agents can sometimes lose balance, fall, and trigger early termination. The large-table setting is particularly more challenging for two-agent teams, which have less mechanical leverage to stabilize and lift the heavier tables, resulting in slower transport.

We also evaluate zero-shot generalization to 12-agent and 16-agent teams carrying the large tables, pushing the policy far beyond the team sizes encountered during training. The unified policy continues to produce synchronized and coherent motion, achieving relatively high success rates and low jerk, in contrast to the baseline which becomes highly unstable. However, when teams become very large, the tabletop perimeter becomes crowded, and agents have not fully learned to position themselves within tight support gaps, resulting in lower cooperative-time ratios. Nonetheless, our policy exhibits overall robust generalization to object sizes and larger team sizes unseen during training. We provide several qualitative results in Figure 3.

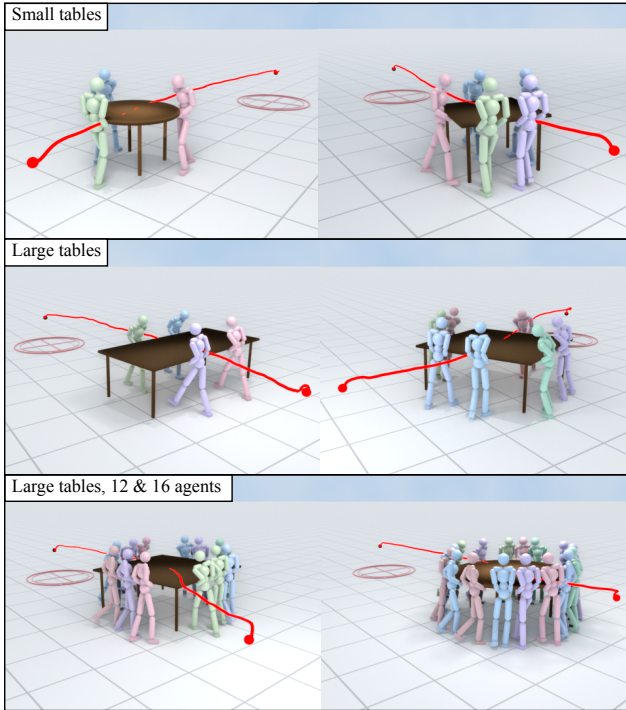


Figure 3. Qualitative visualization of the zero-shot generalization under unseen table geometries and team sizes. Red line indicates the table’s movement trajectory, and the black dot marks its final position at the end of each episode.

7. Multiple Affordance Behaviors

While our main experiments focus on a single affordance behavior (edge-lifting), our framework can support multiple affordance behaviors by adapting the task reward. Fig. 4 demonstrates this capability, where agents adapt to either side-holding or edge-lifting depending on their proximity to regions where the corresponding affordances are feasible.

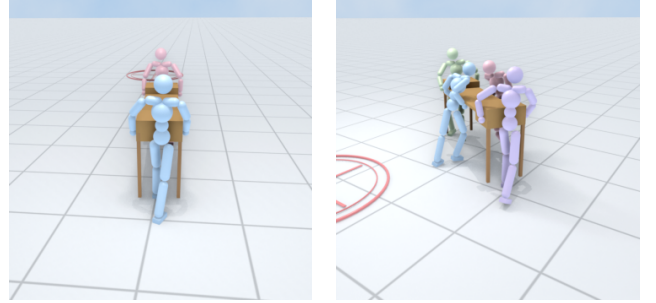


Figure 4. Examples of multiple affordance behaviors learned by adapting the task reward. The agents are able to adapt to side-holding or edge-lifting while being able to walk toward diverse directions. The policy is trained using the same single-human reference motions as our main experiments.

References

- [1] 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. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 4
- [2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 4
- [3] 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. 1
- [4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1