

# Recovering Physically Plausible Human-Object Interactions from Monocular Videos

## Supplementary Material

In this Supplementary Material, we provide additional experiments and method details that were not included in the main paper due to space constraints. The reader is encouraged to watch our supplementary video which includes temporal results of our method, comparison with baselines and ablation results.

### S.1. Kinematic to Simulation HOI States

Following prior work [2–5, 9, 16], for the human, we convert SMPL-H [10] parameters into the humanoid URDF [8] format with position states. Specifically, the humanoid URDF is instantiated using the SMPL-H body shape  $\beta$ , which determines the humanoid skeleton. The humanoid’s degree-of-freedom (DoF) states are obtained directly from the SMPL-H body pose parameters  $\theta$ . Similarly, the humanoid’s global translation and global orientation are taken from the corresponding SMPL-H global transformation. For the object, the parameters used in simulation are identical to those in the kinematic reconstruction, both represented as a 6DoF pose. We use IsaacGym [6] as our physics simulator. The hyperparameters and the physical parameters of objects are shown in Table S.1.

### S.2. Detailed Reward Functions

In this section, we discuss the detailed form of the rewards used when training our policy.

**Human Motion Reward.** The human motion reward,  $r_t^h$ , encourages accurate reproduction of the reference human pose, translation, linear velocity, and angular velocity:

$$r_t^h = \exp(-\lambda_\theta^h E_{\theta,t}^h - \lambda_p^h E_{p,t}^h), \quad (1)$$

where:

$$E_{\theta,t}^h = \|\theta_t^h - \hat{\theta}_t^h\|_2^2, \quad (2)$$

$$E_{p,t}^h = \|p_t^h - \hat{p}_t^h\|_2^2, \quad (3)$$

while  $\lambda_\theta^h, \lambda_p^h$  are hyperparameters that control the weights of the corresponding terms.

**Object Motion Reward.** Similarly, the object motion reward,  $r_t^o$ , enforces that the object orientation, position and linear velocity is consistent with the corresponding quantities of the reference object motion:

$$r_t^o = \exp(-\lambda_\theta^o E_{\theta,t}^o - \lambda_p^o E_{p,t}^o - \lambda_{\dot{p}}^o E_{\dot{p},t}^o), \quad (4)$$

Hyperparameter	Value
Sim $dt$	1/60s
Control $dt$	1/30s
Number of envs	1024
Number of substeps	2
Number of pos iterations	4
Number of vel iterations	0
Contact offset	0.02
Rest offset	0.0
Max depenetration velocity	100
Ground restitution	0.7
Object & ground friction	0.9
Object contact restitution	0.1
Object contact damping	1000
Object contact stiffness	30000
Object contact rest offset	0.015
Object density	500
Object max convex hulls	64

Table S.1. **Simulation hyperparameters used in Isaac Gym [6].**

with:

$$E_{\theta,t}^o = \|\theta_t^o - \hat{\theta}_t^o\|_2^2, \quad (5)$$

$$E_{p,t}^o = \|p_t^o - \hat{p}_t^o\|_2^2, \quad (6)$$

$$E_{\dot{p},t}^o = \|\dot{p}_t^o - \hat{\dot{p}}_t^o\|_2^2, \quad (7)$$

while  $\lambda_\theta^o, \lambda_p^o, \lambda_{\dot{p}}^o$  are hyperparameters that control the weights of the corresponding terms.

**Contact Reward.** We incorporate a contact-consistency cost,  $r_t^c$ , that measures discrepancies between predicted and reference contact states for both hands. We want to highlight that our policy utilizes the minimum contact guidance, *i.e.*, binary hand contacts, without the need for annotations of contact points, nor other contact region. Our method can automatically obtain and update the reliable contact points. For the reference hand contact label  $\hat{c}_t$ , we give the left hand and right hand a binary label respectively, by identifying whether the hand vertices’ minimum distance toward the object is smaller than 1cm, from the ground truth mesh. We obtain the hand contact state  $c_t$  in simulation through the force sensor on the humanoid hand.

The contact points are defined as pairs between the hand joints and the object vertices that are in contact. Since the results from kinematic reconstruction are noisy, the estimated

contact points are typically inconsistent and not reliable. Therefore, we set the contact points to be undefined when we start the RL training and we anticipate to obtain better contact points during training. With our adaptive sampling and the mechanism of dual propagation with kinematics update, we can identify the successful rollouts with good contacts. The update of the contact points means that we keep track of the contact points of these successful rollouts (*i.e.*, the index of the object vertices closest to every hand joint). Overall, we have:

$$r_t^c = \exp(-\lambda_c E_{c,t}) \exp(-\lambda^{cp} E_t^{cp}), \quad (8)$$

where:

$$E_{c,t} = \sum \left( \|c_t^{\text{lhand}} - \hat{c}_t^{\text{lhand}}\| \odot \hat{c}_t^{\text{lhand}} + \|c_t^{\text{rhand}} - \hat{c}_t^{\text{rhand}}\| \odot \hat{c}_t^{\text{rhand}} \right) \quad (9)$$

$$E_t^{cp} = \|p_t^{ch} - p_t^{co}\|_2^2 \odot \hat{c}_t. \quad (10)$$

Here,  $p_t^{ch}$  and  $p_t^{co}$  are the positions of a paired hand joint and object vertex respectively that are in contact, while  $\lambda_c, \lambda^{cp}$  are hyperparameters controlling the reward weights. The second term is considered only when the contact points have been updated.

**Interaction Distance (Proximity) Reward.** The interaction distance (proximity) reward,  $r_t^d$ , encourages the human-object spatial relationship to match the kinematic reference:

$$r_t^d = \exp(-\lambda_d E_{d,t}), \quad E_{d,t} = \|d_t - \hat{d}_t\|_2^2. \quad (11)$$

Here,  $d_t$  is calculated using the minimum distance of every human joint towards the object vertices and  $\lambda_d$  is a reward hyperparameter.

Finally, as is standard in related work [5, 13, 16], we also include an energy-based reward. This term is common in prior approaches, but we include it here for completeness.

**Energy Reward.** The energy reward,  $r_t^e$ , penalizes abrupt motion and sudden contact forces, encouraging smoother and more stable human-object interactions. The energy terms are defined as:

$$E_{h,t}^e = \sum \|a_{h,t}\|, \quad (12)$$

$$E_{o,t}^e = \sum \|a_{o,t}\|, \quad (13)$$

$$E_{c,t}^e = \max \|f_t\|, \quad (14)$$

where  $a_{h,t}$  denotes human joint accelerations,  $a_{o,t}$  the object acceleration, and  $f_t$  the contact forces on human rigid bodies. The final energy reward is:

$$r_t^e = \exp(-\lambda_h^e E_{h,t}^e - \lambda_o^e E_{o,t}^e - \lambda_c^e E_{c,t}^e), \quad (15)$$

where  $\lambda_h^e, \lambda_o^e, \lambda_c^e$  are hyperparameters that control the weights of the corresponding terms. All the reward weights are shown in Table S.2.

Reward Weight	Value
$\lambda_\theta^h, \lambda_p^h$	30, 2.5
$\lambda_\theta^o, \lambda_p^o, \lambda_p^o$	5, 0.1, 0.1
$\lambda^{cp}, \lambda_c$	10, 5
$\lambda_d$	5
$\lambda_h^e, \lambda_o^e, \lambda_c^e$	2e-5, 2e-5, 1e-9

Table S.2. **Reward weights in our RL training.**

Hyperparameter	Value
Action distribution	153D Continuous
Discount factor $\gamma$	0.99
Generalized advantage estimation $\lambda$	0.95
Entropy regularization coefficient	0.0
Optimizer	Adam [1]
Learning rate (Actor)	2e-5
Learning rate (Critic)	1e-4
Minibatch size	16384
Horizon length $H$	32
Action bounds loss coefficient	10
Maximum episode length	300
Number of epochs	2000

Table S.3. **Hyperparameters used in our RL training.**

### S.3. Algorithmic details

In this section, we provide more details of our algorithm. In our implementation, we store the kinematics, contact points and rollout information in a buffer with 3 slots for each frame. In our adaptive sampling, the sampling rate of a certain frame is proportional to the average length of stored rollouts in its slots. During RL training, for the kinematics buffer used for initialization, if a rollout passes a certain frame, it will replace the associated kinematic in the buffer if the remaining length is 10 frames longer than the shortest length in the slots or if the difference is within 10 frames but the rollout has a higher reward than stored rollout rewards. When the kinematics get updated, we also update with the associated rollout length, contact point and rewards. Since the calculation of rollout length is different for forward and backward process, we have different buffers for the forward and backward process. For each direction we have buffers, one for the direction itself and one for the inter-direction update. We also have another separate buffer used for the tracking target for different direction. The conditions for inter-direction cross update and tracking target update are stricter than the intra-direction update. Please see Algorithm 1 for a quick summary, or Algorithms 2 and 3 for more details. The input parameters  $\gamma$ ,  $L_{\text{valid},1}$ ,  $L_{\text{valid},2}$ ,  $m$ ,  $n_1$ ,  $n_2$ , and  $\text{Traverse\_RSI\_Epochs}$  are 0.99, 25, 40, 10, 30,

**Algorithm 1** Adaptive Sampling and Kinematics Update

---

```

1: Input: Noisy kinematic sequence  $K$  of length  $T$ ,
   threshold  $m$ , discount factor  $\gamma$ , valid length threshold
    $L_{\text{valid}}$ , pre-trained policy  $\pi_{\text{pre}}$ 
2: Output: Kinematic Buffer  $\mathcal{B}$ , trained policy  $\pi$ 
3: Initialize  $\mathcal{B}$ , where each frame  $t$  has 3 slots storing
    $\{K[t], \text{None}, 0, 0\}$  for  $\{\text{Kin}, CP, L, R\}$ 
4: Initialize policy  $\pi \leftarrow \pi_{\text{pre}}$ 
5: for epoch = 1, 2, ... do
6:   Sample  $t_{\text{start}} \in \{1, \dots, T\}$  with  $P(t) \propto$ 
      $\sum_{s=1}^3 \mathcal{B}[t][s].L$   $\triangleright$  Adaptive sampling
7:   Sample  $s_{\text{start}} \in \{1, 2, 3\}$  with  $P(s) \propto \mathcal{B}[t_{\text{start}}][s].L$ 
8:    $\text{Kin}_{\text{init}} \leftarrow \mathcal{B}[t_{\text{start}}][s_{\text{start}}].\text{Kin}$ 
9:   Execute rollout initialized from  $\text{Kin}_{\text{init}}$  at frame  $t_{\text{start}}$ 
     using  $\pi$ , terminating at  $t_{\text{end}}$ 
10:  Let  $\tau$  be the collected trajectory with step rewards
11:  if  $t_{\text{end}} - t_{\text{start}} \leq L_{\text{valid}}$  then  $\mathcal{U} \leftarrow \{t_{\text{start}}\}$  else  $\mathcal{U} \leftarrow \tau$ 
12:  for each frame  $t \in \mathcal{U}$  do  $\triangleright$  Buffer update
13:     $L_{\text{new}} \leftarrow |t_{\text{end}} - t|$ 
14:     $R_{\text{new}} \leftarrow \sum_{k=0}^{L_{\text{new}}} \gamma^k r_{t+k}$ 
15:    Obtain physics state  $\text{Kin}_{\text{new}}$  and contact point
      $CP_{\text{new}}$  from simulation at frame  $t$ 
16:     $s_{\text{min}} \leftarrow \arg \min_s \mathcal{B}[t][s].L$   $\triangleright$  Get the worst slot
17:    Retrieve  $\{L_{\text{min}}, R_{\text{min}}\}$  from  $\mathcal{B}[t][s_{\text{min}}]$ 
18:    if  $(L_{\text{new}} > L_{\text{min}} + m)$  or  $(L_{\text{min}} < L_{\text{new}} \leq$ 
      $L_{\text{min}} + m$  and  $R_{\text{new}} > R_{\text{min}})$  then
19:       $\mathcal{B}[t][s_{\text{min}}] \leftarrow \{\text{Kin}_{\text{new}}, CP_{\text{new}}, L_{\text{new}}, R_{\text{new}}\}$ 
20:    end if
21:  end for
22: end for

```

---

60, and 30, respectively.

## S.4. Comparison using InterTrack

InterTrack [15] is a state of the art template-free 4D HOI reconstruction method. For our pipeline, this can be used as an alternative to VisTracker for providing the input kinematic reconstruction. One challenge with InterTrack is that it produces time-varying scales for both the human and the object, giving inconsistent shapes across time. This is problematic for simulation, which requires fixed human and object scales. Regardless, we evaluated InterTrack as initialization in Table S.4. We report results on 9 clips from BEHAVE. The results are consistently worse compared to using VisTracker, which we attribute to the aforementioned time-varying scale. However, we observe the same trend, where our method using InterTrack initializations performs better than InterMimic with InterTrack input.

Method	InterTrack		VisTracker	
	SR-B $\uparrow$	SR-F $\uparrow$	SR-B $\uparrow$	SR-F $\uparrow$
InterMimic (direct)	0	3.7	0	7.6
InterMimic (ft)	11.1	27.4	22.2	34.2
Ours	<b>44.4</b>	<b>60.5</b>	<b>66.7</b>	<b>73.6</b>

Table S.4. **Comparison with InterTrack Reconstruction.** We report results on 9 sequences from BEHAVE.

## S.5. Training Details

Our method adopts a single-sequence optimization paradigm [7, 13] instead of a universal physical tracker [5, 16]. The control policy  $\pi$  is trained using PPO [12] with the policy gradient  $L(\psi) = \mathbb{E}_t[\min(r_t(\psi)A_t, \text{clip}(r_t(\psi), 1 - \epsilon, 1 + \epsilon)A_t)]$ . Here,  $\psi$  are the parameters of  $\pi$ , and  $r_t(\psi)$  quantifies the difference in action likelihoods between the updated and old policies.  $\epsilon$  is a small constant, and  $A_t$  is the advantage estimate given by the generalized advantage estimator GAE( $\lambda$ ) [11]. The training parameters are shown in Table S.3. Each sequence takes about 16 hours on a A6000 GPU for training.

## S.6. Evaluation Details

**Why different values of our method on 3D metrics in comparison with VisTracker [14] and InterMimic [16].** For comparison with VisTracker (Table 1), results on 3D metrics are evaluated on the successful segments of our physical tracker. In terms of fairness, failure cases of our physical tracker can be reliably detected, and in such cases the kinematic estimate can be used as a fallback. For comparison with InterMimic (Table 2), results on 3D metrics are evaluated on intersection of successful segments of our physical tracker and InterMimic. We follow this choice because it is meaningless to evaluate on failure rollouts (e.g., object drop, humanoid fall down). We report the successful frames on BEHAVE and InterCap dataset in Table S.5 for the comparison of our method with VisTracker, and in S.6 our method with InterMimic.

**Early Termination and Failure Detection.** Following InterMimic [16], the early termination during training and the failure during evaluation are triggered whenever any of the following conditions occur: (i) object points deviate from their reference kinematics by more than 0.5 m on average, (ii) required body-object contact is lost for over 10 consecutive frames, (iii) the humanoid root joint is under the height of 0.15 m, (iv) the humanoid joints are, on average, more than 0.5 meters from their reference kinematics. The reference kinematics here denotes the kinematic reconstruction results because we do not want the policy to have access to ground-truth kinematics before evaluation on 3D metrics.

---

**Algorithm 2** Adaptive Sampling and Kinematics Update (Part 1)

---

```

1: Input: Noisy kinematic sequence  $K$  of length  $T$ , discount factor  $\gamma$ , valid length thresholds  $L_{\text{valid},1}, L_{\text{valid},2}$ , pre-trained policy  $\pi_{\text{pre}}$ , thresholds  $m, n_1, n_2$ 
2: Output: Initialization self Forward Buffer  $\mathcal{B}_{\text{self}}^F$ , Backward Buffer  $\mathcal{B}_{\text{self}}^B$ , Tracking Target Buffers  $\mathcal{B}_{\text{track}}^F, \mathcal{B}_{\text{track}}^B$ , policies  $\pi_F, \pi_B$ 
3: Initialize self buffers  $\mathcal{B}_{\text{self}}^F$  and  $\mathcal{B}_{\text{self}}^B$ , where each frame  $t$  has 3 slots storing  $\{K[t], \text{None}, 0, 0\}$  for  $\{\text{Kin}, CP, L, R\}$   $\triangleright$  kinematics, contact points, rollout length and reward
4: Initialize cross buffers  $\mathcal{B}_{\text{cross}}^F$  and  $\mathcal{B}_{\text{cross}}^B$ , where each frame  $t$  has 3 slots storing  $\{K[t], \text{None}, 0\}$  for  $\{\text{Kin}, CP, L\}$   $\triangleright$  prepared for intra-direction update
5: Initialize tracking buffers  $\mathcal{B}_{\text{track}}^F$  and  $\mathcal{B}_{\text{track}}^B$ , where each frame  $t$  has 1 slot storing  $K[t]$  for Kin
6: Initialize flags  $\text{AdaptiveMode}^F \leftarrow \text{False}$  and  $\text{AdaptiveMode}^B \leftarrow \text{False}$ 
7: Initialize policies  $\pi_F \leftarrow \pi_{\text{pre}}$  and  $\pi_B \leftarrow \pi_{\text{pre}}$ 
8: for epoch = 1, 2, ... do
9:   for direction  $d \in \{F, B\}$  in parallel do
10:    if not  $\text{AdaptiveMode}^d$  and epoch >  $\text{Traverse\_RSI\_Epochs}$  and  $\sum_{t=1}^T \sum_{s=1}^3 \mathbb{I}(\mathcal{B}_{\text{self}}^d[t][s].L > L_{\text{valid},1}) > 3$  then
     $\triangleright \mathbb{I}(\cdot)$  is indicator
11:       $\text{AdaptiveMode}^d \leftarrow \text{True}$ 
12:    end if
13:    if not  $\text{AdaptiveMode}^d$  then
14:      Sample  $t_{\text{start}}$  uniformly without replacement from  $\{1, 2, \dots, T\}$   $\triangleright$  Traverse RSI
15:       $\text{Kin}_{\text{init}} \leftarrow K[t_{\text{start}}]$ 
16:    else
17:      Sample  $t_{\text{start}}$  with probability  $\propto \sum_{s=1}^3 \mathcal{B}_{\text{self}}^d[t_{\text{start}}][s].L$   $\triangleright$  Adaptive sampling
18:      Sample  $s_{\text{start}} \in \{1, 2, 3\}$  with probability  $\propto \mathcal{B}_{\text{self}}^d[t_{\text{start}}][s].L$ 
19:       $\text{Kin}_{\text{init}} \leftarrow \mathcal{B}_{\text{self}}^d[t_{\text{start}}][s_{\text{start}}].\text{Kin}$ 
20:    end if
21:    Execute rollout initialized from  $\text{Kin}_{\text{init}}$  at frame  $t_{\text{start}}$  using  $\pi_d$ , terminating at  $t_{\text{end}}$ 
22:    Let  $\tau$  be the collected trajectory with step rewards
23:    if not ( $\text{AdaptiveMode}^d$  and  $t_{\text{end}} - t_{\text{start}} > L_{\text{valid},2}$ ) then  $\mathcal{U} \leftarrow \{t_{\text{start}}\}$  else  $\mathcal{U} \leftarrow \tau$ 
24:    for each frame  $t \in \mathcal{U}$  do  $\triangleright$  Intra-direction update
25:      Obtain physics state  $\text{Kin}_{\text{new}}$  and contact point  $CP_{\text{new}}$  from simulation at frame  $t$ 
26:       $L_{\text{new}}^{\text{self}} \leftarrow |t_{\text{end}} - t|$   $\triangleright$  Update self buffer
27:       $R_{\text{new}} \leftarrow \sum_{k=0}^{L_{\text{new}}^{\text{self}}} \gamma^k r_{t+k}$ 
28:       $s_{\text{min}}^{\text{self}} \leftarrow \arg \min_s \mathcal{B}_{\text{self}}^d[t][s].L$ 
29:      Retrieve  $\{L_{\text{min}}^{\text{self}}, R_{\text{min}}^{\text{self}}\}$  from  $\mathcal{B}_{\text{self}}^d[t][s_{\text{min}}^{\text{self}}]$ 
30:      if ( $L_{\text{new}}^{\text{self}} > L_{\text{min}}^{\text{self}} + m$ ) or ( $L_{\text{min}}^{\text{self}} < L_{\text{new}}^{\text{self}} \leq L_{\text{min}}^{\text{self}} + m$  and  $R_{\text{new}} > R_{\text{min}}^{\text{self}}$ ) then
31:         $\mathcal{B}_{\text{self}}^d[t][s_{\text{min}}^{\text{self}}] \leftarrow \{\text{Kin}_{\text{new}}, CP_{\text{new}}, L_{\text{new}}^{\text{self}}, R_{\text{new}}\}$ 
32:      end if
33:       $L_{\text{new}}^{\text{cross}} \leftarrow |t - t_{\text{start}}|$   $\triangleright$  Update cross buffer
34:       $s_{\text{min}}^{\text{cross}} \leftarrow \arg \min_s \mathcal{B}_{\text{cross}}^d[t][s].L$ 
35:      Retrieve  $L_{\text{min}}^{\text{cross}}$  from  $\mathcal{B}_{\text{cross}}^d[t][s_{\text{min}}^{\text{cross}}]$ 
36:      if  $L_{\text{new}}^{\text{cross}} > L_{\text{min}}^{\text{cross}}$  then
37:         $\mathcal{B}_{\text{cross}}^d[t][s_{\text{min}}^{\text{cross}}] \leftarrow \{\text{Kin}_{\text{new}}, CP_{\text{new}}, L_{\text{new}}^{\text{cross}}\}$ 
38:      end if
39:    end for
40:  end for

```

---

For condition (ii), we only consider hand-object contact because we only incorporate reference hand contact label in our method and InterMimic finetune.

## References

- [1] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

---

**Algorithm 3** Adaptive Sampling and Kinematics Update (Part 2)

---

```
41: if epoch (mod 10) == 0 then ▷ Inter-direction cross update
42:   for  $t = 1, 2, \dots, T$  do
43:     for direction  $d \in \{F, B\}$  do
44:       Let  $\bar{d}$  be the opponent direction of  $d$ 
45:        $s_{\text{best}} \leftarrow \arg \max_s \mathcal{B}_{\text{cross}}^d[t][s].L$ 
46:       Retrieve  $\{\text{Kin}_d, CP_d, L_d\}$  from  $\mathcal{B}_{\text{cross}}^d[t][s_{\text{best}}]$ 
47:       Retrieve  $\{L_{\bar{d}}, R_{\bar{d}}\}$  from  $\mathcal{B}_{\text{self}}^{\bar{d}}[t][3]$  ▷ Slot 3 is reserved for inter-direction states
48:       if  $L_d > L_{\bar{d}} + m$  then
49:          $\mathcal{B}_{\text{self}}^d[t][3] \leftarrow \{\text{Kin}_d, CP_d, L_d, R_{\bar{d}}\}$ 
50:       end if
51:     end for
52:   end for
53: end if
54: if epoch (mod 100) == 0 then ▷ Tracking Buffer update
55:   for direction  $d \in \{F, B\}$  do
56:     Let  $\bar{d}$  be the opponent direction of  $d$ 
57:      $t_{\text{best}}, s_{\text{best}} \leftarrow \arg \max_{t,s} \mathcal{B}_{\text{self}}^d[t][s].L$ 
58:     Retrieve  $\text{Kin}_{\text{init}}$  from  $\mathcal{B}_{\text{self}}^d[t_{\text{best}}][s_{\text{best}}]$ 
59:     Execute rollout from  $\text{Kin}_{\text{init}}$  at  $t_{\text{best}}$  using  $\pi_d$  (inference mode), terminating at  $t_{\text{end}}$ 
60:     Let  $\tau_{\text{track}}$  be the collected trajectory
61:     for each frame  $t \in \tau_{\text{track}}$  do
62:       Obtain physics state  $\text{Kin}_t$  from simulation at frame  $t$ 
63:        $L_{\text{new}}^{\text{self}} \leftarrow |t_{\text{end}} - t|$ ,  $L_{\text{new}}^{\text{cross}} \leftarrow |t - t_{\text{best}}|$ 
64:        $L_{\text{min}}^{\text{self}} \leftarrow \min_s \mathcal{B}_{\text{self}}^d[t][s].L$ ,  $L_{\text{min}}^{\text{opp-self}} \leftarrow \min_s \mathcal{B}_{\text{self}}^{\bar{d}}[t][s].L$ 
65:       if  $L_{\text{new}}^{\text{self}} > L_{\text{min}}^{\text{self}} + n_1$  then
66:          $\mathcal{B}_{\text{track}}^d[t] \leftarrow \text{Kin}_t$  ▷ Update own tracking buffer
67:       end if
68:       if  $L_{\text{new}}^{\text{cross}} > L_{\text{min}}^{\text{opp-self}} + n_2$  then
69:          $\mathcal{B}_{\text{track}}^{\bar{d}}[t] \leftarrow \text{Kin}_t$  ▷ Update opponent's tracking buffer
70:       end if
71:     end for
72:   end for
73: end if
74: end for
```

---

- [2] Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. Dynamics-regulated kinematic policy for egocentric pose estimation. *NeurIPS*, 2021. 1
- [3] Zhengyi Luo, Shun Iwase, Ye Yuan, and Kris Kitani. Embodied scene-aware human pose estimation. *NeurIPS*, 2022.
- [4] Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582*, 2023.
- [5] Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *ICCV*, 2023. 1, 2, 3
- [6] 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
- [7] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018. 3
- [8] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. ROS: an open-source robot operating system. In *ICRA workshop on open source software*, page 5. Kobe, 2009. 1
- [9] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *CVPR*, 2023. 1
- [10] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM TOG*, 36(6), 2017. 1
- [11] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous con-

Dataset	Subject	Object	Action/Seg	Start Frame	End Frame
BEHAVE	03	backpack	hug	0	444
	03	boxlarge	lift	0	359
	03	boxmedium	lift	0	419
	03	chairblack	lift	0	149
	03	chairblack	sitstand	0	299
	03	chairwood	sit	0	299
	03	monitor	move	0	269
	03	plasticcontainer	hold	0	147
	03	stool	lift	0	229
	03	stool	sit	0	449
	03	suitcase	lift	2	179
	03	tablesmall	lean	0	299
	03	tablesmall	lift	0	359
	03	tablesquare	lift	0	359
	03	tablesquare	move	0	233
	03	tablesquare	sit	0	369
	03	trashbin	lift	10	169
	03	yogaball	play	0	299
	03	yogaball	sit	0	299
	03	yogamat	play	0	299
InterCap	09	obj01	Seg_2	0	235
	09	obj03	Seg_0	0	320
	10	obj03	Seg_0	0	240
	10	obj03	Seg_1	0	310
	09	obj04	Seg_1	0	295
	10	obj04	Seg_0	0	200
	10	obj04	Seg_1	0	280
	09	obj06	Seg_0	0	150
	09	obj06	Seg_1	0	245
	09	obj06	Seg_2	0	145
	10	obj06	Seg_0	50	290
	10	obj06	Seg_1	60	309
	09	obj07	Seg_0	0	340
	09	obj07	Seg_1	0	380
	10	obj07	Seg_0	0	260
	10	obj07	Seg_1	0	260
	09	obj08	Seg_1	0	215
	09	obj08	Seg_2	0	235
	09	obj10	Seg_0	0	270
	09	obj10	Seg_1	0	120
	10	obj10	Seg_0	0	279
	10	obj10	Seg_1	0	200

Table S.5. Successful frames of our method on BEHAVE and InterCap dataset. These frames are used for the evaluation reported in Table 1 of the main manuscript.

Dataset	Subject	Object	Action/Seg	Start Frame	End Frame
BEHAVE	03	backpack	hug	0	62
	03	boxmedium	lift	0	418
	03	chairblack	sitstand	0	63
	03	monitor	move	0	268
	03	stool	lift	10	83
	03	stool	sit	0	449
	03	tablesmall	lean	0	299
	03	tablesmall	lift	0	131
	03	tablesquare	move	0	232
	03	tablesquare	sit	0	342
	03	yogaball	sit	0	298
InterCap	09	obj03	Seg_0	0	229
	10	obj03	Seg_0	0	239
	10	obj03	Seg_1	0	177
	10	obj04	Seg_0	0	96
	10	obj04	Seg_1	0	139
	09	obj06	Seg_1	43	244
	09	obj07	Seg_0	0	339
	10	obj07	Seg_0	0	259
	10	obj07	Seg_1	0	259
	09	obj10	Seg_0	0	269
	09	obj10	Seg_1	0	119
	10	obj10	Seg_0	0	279
	10	obj10	Seg_1	0	68

Table S.6. Intersection of successful frames for our method and InterMimic on BEHAVE and InterCap dataset. The frames in this intersection are used for the evaluation reported in Table 2 of the main manuscript.

- trol using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 3
- [12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3
- [13] Yinhuai Wang, Jing Lin, Ailing Zeng, Zhengyi Luo, Jian Zhang, and Lei Zhang. PhysHOI: Physics-based imitation of dynamic human-object interaction. *arXiv preprint arXiv:2312.04393*, 2023. 2, 3
- [14] Xianghui Xie, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Visibility aware human-object interaction tracking from single RGB camera. In *CVPR*, 2023. 3
- [15] Xianghui Xie, Jan Eric Lenssen, and Gerard Pons-Moll. InterTrack: Tracking human object interaction without object templates. In *3DV*, 2025. 3
- [16] Sirui Xu, Hung Yu Ling, Yu-Xiong Wang, and Liang-Yan Gui. InterMimic: Towards universal whole-body control for physics-based human-object interactions. In *CVPR*, 2025. 1, 2, 3