

# Push-and-Step: From RL-Based Balance Recovery to Physical Simulation of Dense Crowds

## Supplementary Material

### 1. Additional experiments details

#### 1.1. Ablation process

In this section, we cover the process by which ablation studies are performed throughout the paper and supplementary material. For each ablation study, policies are compared according to a set of metrics, which are summarized in the following:

- foot sliding is computed by a cumulative sum over time of both feet’s instantaneous speed when in contact with the ground, divided by the timestep to result in a distance;
- heading deviation is computed using the quaternion distance between the final torso orientation and the starting torso orientation, converted to degrees in our results;
- kinetic energy is computed by a cumulative sum over time of the sum of each limb’s kinetic energy;
- impulse transmitted is the cumulative sum over time of the instantaneous force applied on other agents divided by the timestep;
- final hand height is the average between the right and left hand’s center of mass final height;
- maximal hand height is the average between the right and left hand’s center of mass maximal height throughout the simulation.

Values shown in Tables 1, 2 and 3 are averages of multiple pushes. The values are to be compared between policies exposed to the same set of pushes. For all the metrics, a lower value indicates better performance, except for maximal hand height (see Section 6.2). For each ablation study, the range of push strength, direction, and agent distribution configuration is specified in the relevant section.

#### 1.2. Pretraining

##### 1.2.1. Training results

Figure S2 illustrates the simulated responses over time of the pretrained policy  $\pi_{\text{pretrain}}$  to external pushes of varying magnitudes, alongside reference motions from our collected dataset (bottom row). As elaborated in Section 4.1 of the main text, our pretraining stage is grounded in eight core motions corresponding to push directions spaced at  $\frac{\pi}{4}$  intervals, while the resulting policies can adapt the character’s gait to the push strength, preserving the motion style of the reference dataset when facing novel pushes outside the dataset. Although multiple characters are shown in Figure S2 for illustrative purposes, each training episode and simulation at this pretraining stage involves only a single agent.

As we can see from the figure, agents can perform stepping behaviors to maintain balance in response to moderate forces, but would fail and fall if the push is too strong (300N). We also observed that task difficulty varies with push direction: for instance, at 240N, a frontal push requires substantial recovery distance, while at 300N, only a direct push from behind results in successful balance retention.

##### 1.2.2. Ablation results

Figure S3 visualizes the ablation study results when a 120N push is applied to the agent’s back for 1s. The statistical results are shown in Figure S1. To obtain the statistical results, a total of 80 pushes were applied spanning 16 directions (at 22.5 intervals) across five maximum force levels: [30, 90, 150, 210, 270]N. With the full reward  $r_{\text{pretrain}}$  (see Eq. 7), Figure S3(a) shows the agent performing a coordinated response combining hand raising and a stabilizing step. Removing the motion quality reward term  $r_{\text{quality}}$  causes foot sliding and excessive forward lean (Figure S3(b) and S1(b)). It also results in unnecessary energy expenditure, as we can see the kinetic energy staying positive instead of returning to zero (Figure S1(c)). Removing the general balance term  $r_{\text{balance}}$  leads to falls (Figure S3(c)). Without the imitation reward  $r_{\text{imit}}$ , the agent adopts an unconventional recovery strategy—spinning on one leg to dissipate momentum (Figure S3(d)), which also produces strong heading deviation (Figure S1(b)).

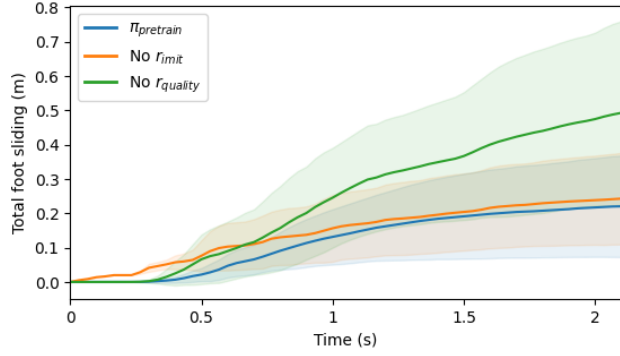
Figure S4 displays balance outcomes across a wide range of push angles and strengths. The comparison highlights a substantial reduction in the range of manageable push strengths when  $r_{\text{balance}}$  balance is removed (blue) versus the full reward (green). The area of manageable pushes without  $r_{\text{balance}}$  averages a maximum strength of only 21N, compared to 230N with the full reward. Additionally, the asymmetry in the resulting policy’s capabilities reflects both the reinforcement learning dynamics and the inherent asymmetry of the reference dataset.

#### 1.3. Adaptation

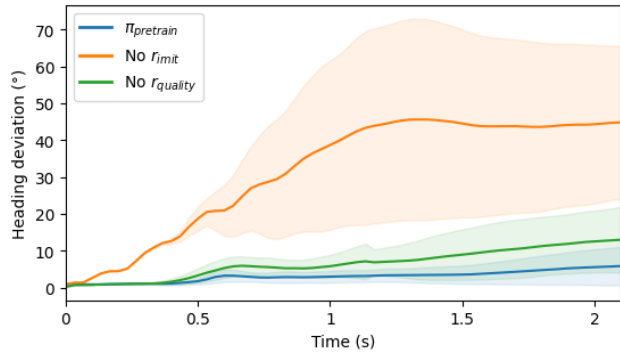
##### 1.3.1. Training results

Figure S6 illustrates the behavior of the adapted policy  $\pi_{\text{adapt}}$  across the three training scenarios, highlighting how the agent learns to deploy hand contacts for balance control based on its surroundings and the direction of the applied push. In scenario (a), where no other agents obstruct the push direction, the agent learns not to raise its hands as seen in (d), in contrast to the pretraining stage where hand-

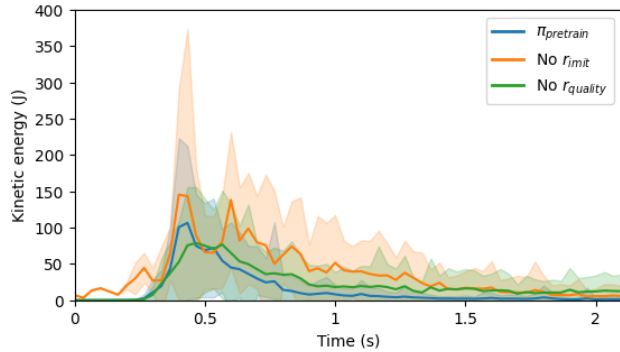




(a) Total foot sliding over time



(b) Heading deviation over time



(c) Kinetic energy over time

Figure S1. Ablation results for the pretrained policy  $\pi_{\text{pretrain}}$  averaged over 80 pushes across multiple directions and magnitudes.

raising was common due to imitation of reference motions. This shift reflects the influence of the adaptation stage. In scenario (b), with two agents positioned in a line ahead of the pushed agent, two distinct behaviors emerge. When the push is applied at an angle (e.g., second column from the left), the agent raises only one hand by selecting the side

that can effectively contribute to momentum dissipation as seen in (e). In contrast, when pushed directly from behind (middle column), the agent raises both hands and places them on the shoulders of the agent directly in front for balancing, as seen in (f). Scenario (c) features two agents side by side in the push direction. For a direct backward push (middle columns), the agent raises both hands and makes contact with the shoulders of the two separate agents as seen in (g), demonstrating spatial awareness and adaptive coordination of the control policy  $\pi_{\text{adapt}}$ . These results collectively demonstrate that the heuristic and adaptation stage enable the policy to learn adaptive and socially aware hand contact strategies: raising hands only when necessary and targeting appropriate contact points based on environmental constraints.

### 1.3.2. Ablation results

Figure S7 presents the ablation study results with respect to the components in the adaptation reward  $r_{\text{adapt}}$  (Eq. 11), while the full adapted behavior is shown in Figure S6. To obtain these results, a total of 90 pushes were applied spanning 9 directions at 22.5 intervals, with only pushes in the direction of the front of the agent, and across five maximum force levels in [30, 90, 150, 210, 270]N. In Figure S7(a), removing the social reward term  $r_{\text{social}}$  causes the agent to keep its hands resting on the pelvis or alongside the body, resulting in collisions with the head and torso at full force. In Figure S7(b), when no hand placement reward  $r_{\text{hand}}$  is used, the hands are always kept in the air and do not return to the side of the body, even if raising the hands is no longer necessary.

Figures S8(a) and (b) provide statistical analysis of hand height and impulse transmitted through hand-shoulder contacts. Three distinct hand height patterns emerge: without  $r_{\text{social}}$ , hands remain at a resting height of 0.81m and are not raised; without  $r_{\text{hand}}$ , hands are raised but fail to return to a neutral pose; and with the full reward, hands are raised when needed and return to resting height afterward which is consistent with the qualitative observations. Impulse transmission also varies across conditions. Without  $r_{\text{social}}$ , a large impulse is delivered through the body, followed by no further interaction. Without  $r_{\text{hand}}$ , the hands remain raised, resulting in prolonged contact and a gradual increase in total impulse. With the full reward, impulse is applied swiftly and then dissipated.

In Figure S7(c), we show that training from scratch without the pretraining stage, even when using the full reward, the policy fails to maintain proper orientation, takes unnecessary steps, and keeps the arms raised for no reason. This result highlights the necessity of our two-stage training scheme, where the pretraining policy focuses on learning through imitation. Table S1 further compares one-stage versus two-stage training, highlighting the benefits of the staged approach.

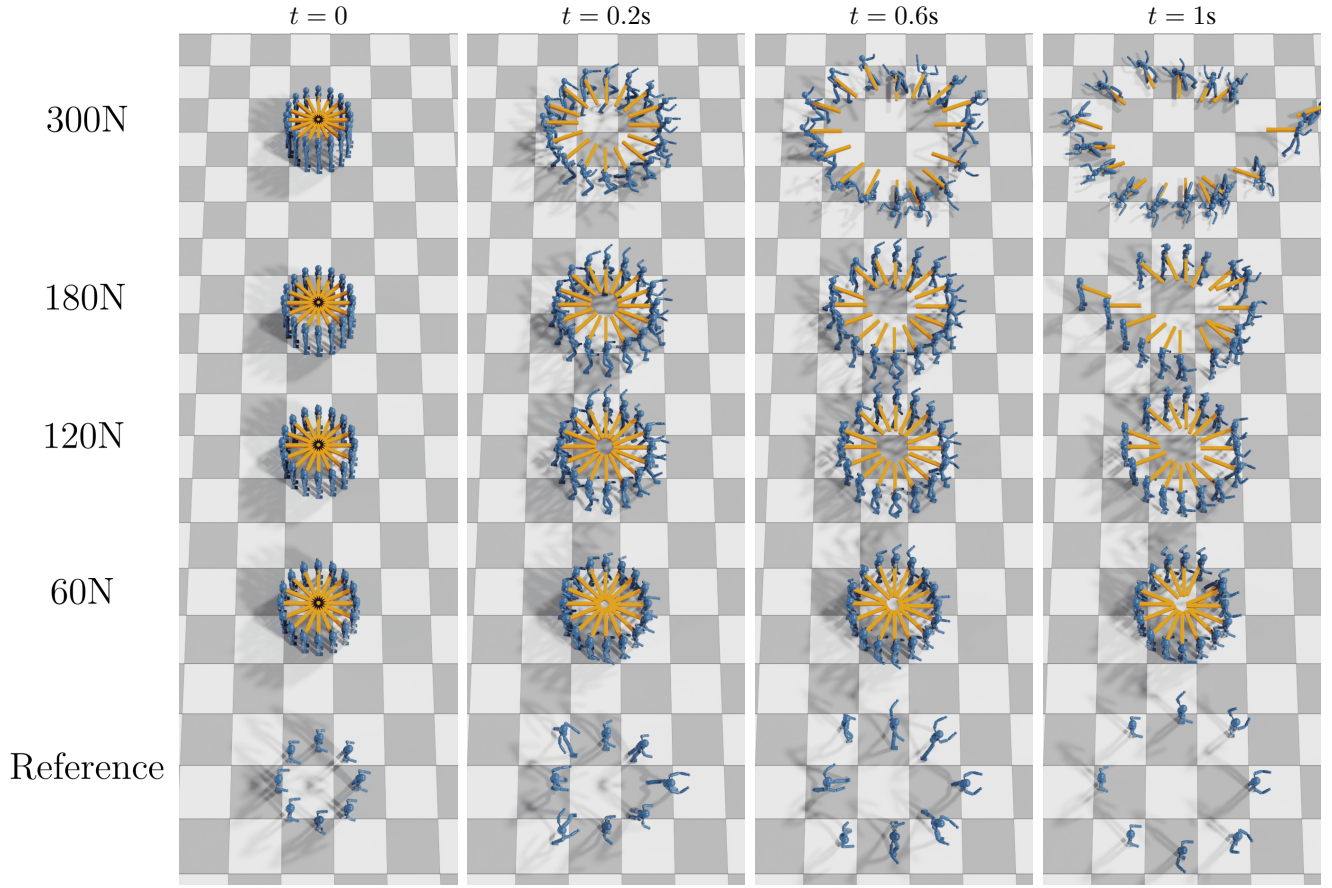


Figure S2. Simulation results of  $\pi_{\text{pretrain}}$  under pushes from multiple directions and varying force magnitudes (top to bottom: high to low). Reference motions from the dataset are shown in the bottom row for comparison.

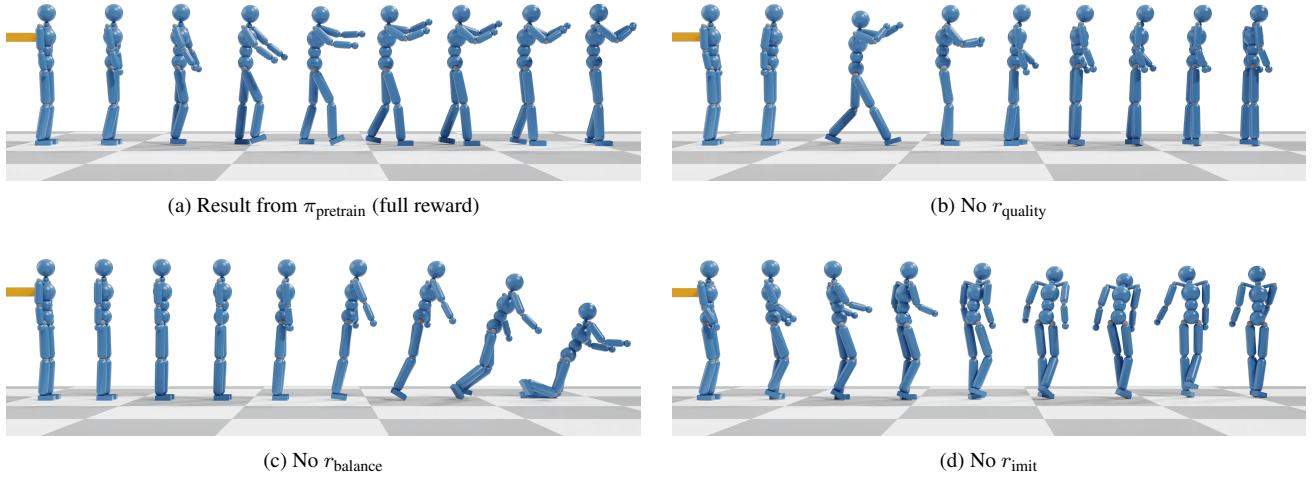


Figure S3. Agent state over time for variations of the pretrained policy  $\pi_{\text{pretrain}}$  following a medium-strength push from the back. Each still illustrates the qualitative behavior resulting from different reward configurations in the ablation study.

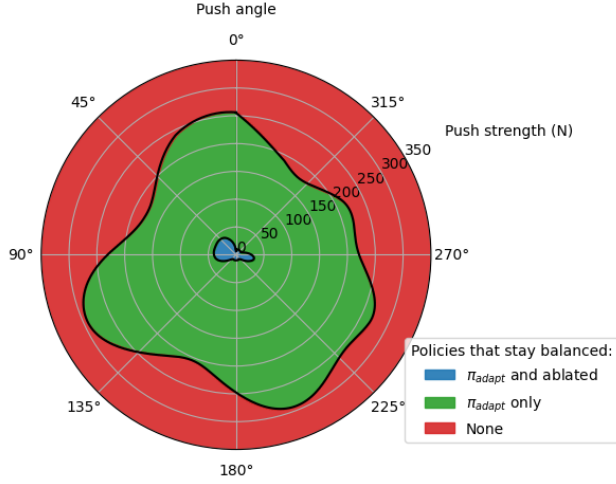


Figure S4. Impact of ablating  $r_{\text{balance}}$  on the agent's ability to maintain balance across varying push directions and magnitudes.

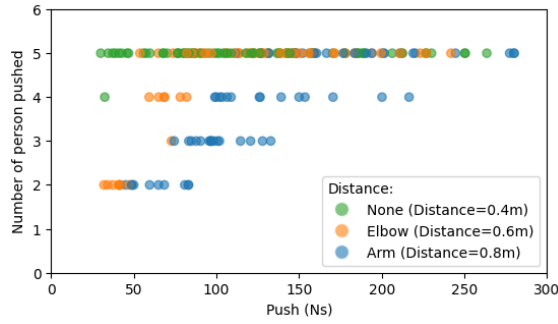


Figure S5. Number of agents pushed in the line scenarios with respect to the push impulse and interpersonal distance.

Stage ablation results		
	Two-stage	One-stage
Heading deviation↓	<b>10.09°</b>	22.67°
Foot sliding↓	<b>27cm</b>	62cm
Kinetic energy↓	<b>2270J</b>	9115J
Impulse transmitted↓	<b>40Ns</b>	105Ns
Final hand height↓	<b>0.81m</b>	1.17m

Table S1. Comparison between the full two-stage policy  $\pi_{\text{adapt}}$  and a single-stage variant trained without pretraining. Metrics include heading deviation, total foot sliding, total kinetic energy, and impulse transmitted averaged over 90 push trials. Lower values (↓) indicate better performance.

## 1.4. Applications

### 1.4.1. Line

Figure S9 shows the simulation results when agents are pushed while standing in a line, under varying levels of interpersonal density. In the subfigures (a), (b), and (c), the push strength remains constant and the propagation speed is unchanged. Reducing the distance between agents increases the total number of individuals affected. In contrast, increasing the push strength, as seen in the comparison between (b) and (d), directly influences the propagation speed. Complementing this and Figure 6 in the main text, Figure S5 quantifies the number of individuals affected as a function of push impulse and interpersonal spacing.

Overall, our simulations reveal that higher interpersonal density leads to increasingly uncontrolled interactions, leaving little room or time for agents to raise their hands to shoulder level for balance. This phenomenon aligns with the empirical observations of real human behaviors in previous literature [?], where human subjects preemptively raised their hands in anticipation of incoming pushes. Such behavior suggests that at extreme densities, returning to a neutral standing pose may no longer be the best strategy.

### 1.4.2. Crowd

In this section, we explore additional results for the crowd scenario. Figure S10 shows heatmaps of the kinetic energy in dense crowds over time, and highlights the role of density in the propagation of kinetic energy. At medium density (a), the kinetic energy peaks close to the push location, whereas at high density (b), it actually increases as the push propagates. As seen in Figure S12, pushes are applied to the crowd through a moving wall, which is shown in orange color, displaced 50cm over a period of 2 seconds. Though the external pushing force is triggered in the same way, agents in the crowd exhibit different behaviors depending on the crowd configurations. At the low (a) and medium (b) densities, push propagation is limited, and agents primarily rely on stepping to maintain balance. In high-density conditions (c), the push spreads more broadly and is mitigated through coordinated hand contacts on neighboring shoulders. At the extreme density (d), agents lack the space to raise their hands, resulting in energy accumulation. As each agent attempts to stay upright by exerting force forward, kinetic energy builds and transfers through the crowd. Eventually, agents at the edge absorb the residual energy, often with no option but to step to dissipate it. While a few manage to remain balanced, most lose stability and fall.

An additional result using  $\pi_{\text{pretrain}}$  without any adaptation is also provided for comparison. In this case, agents are unaware of each other and fail to coordinate, leading to falling by stepping on each other and interlocking limbs. This highlights that without the second adaptation stage, the policy cannot scale to multi-agent scenarios well.

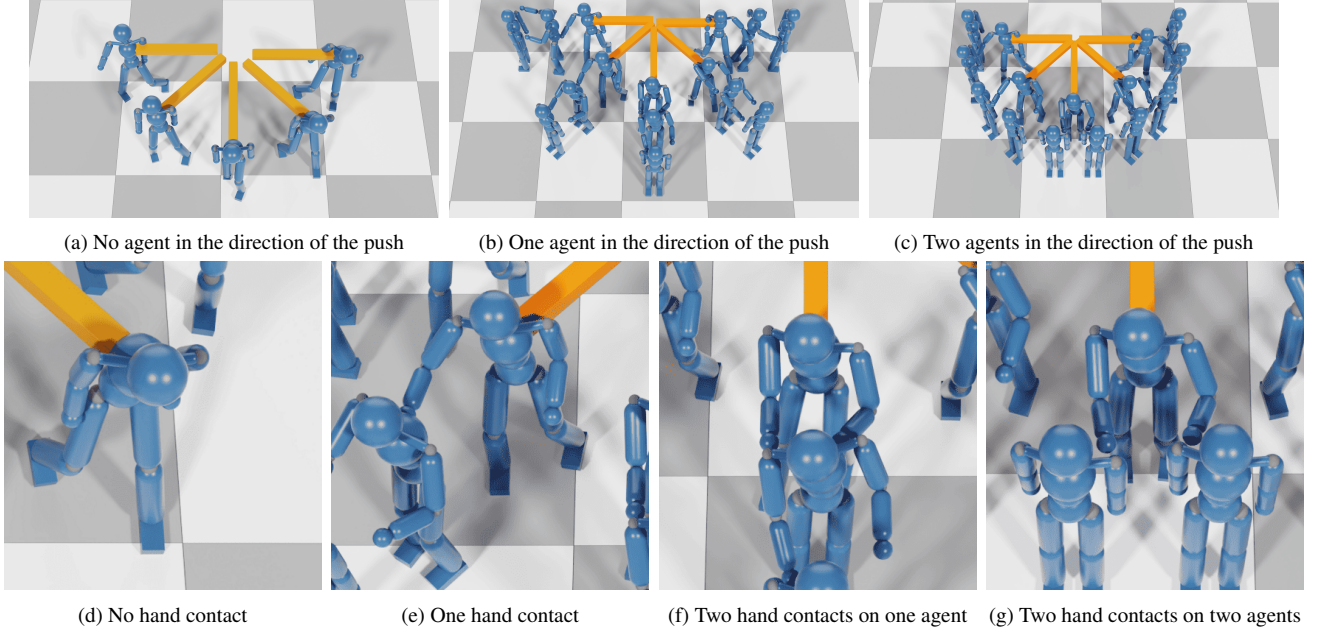


Figure S6. Simulation results of the final policy  $\pi_{\text{adapt}}$  under strong pushes on the back, with varying numbers and configurations of obstacle agents.



Figure S7. Simulation results obtained with variations of the final policy  $\pi_{\text{adapt}}$  under strong pushes on the back. Each still illustrates the qualitative behavior resulting from different reward configurations in the ablation study.

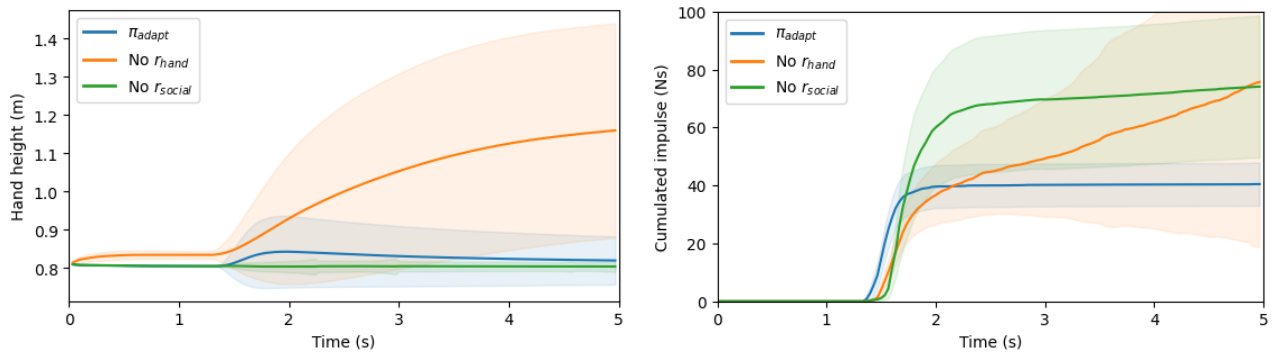


Figure S8. Ablation results for the final policy  $\pi_{\text{adapt}}$  averaged over 90 pushes.



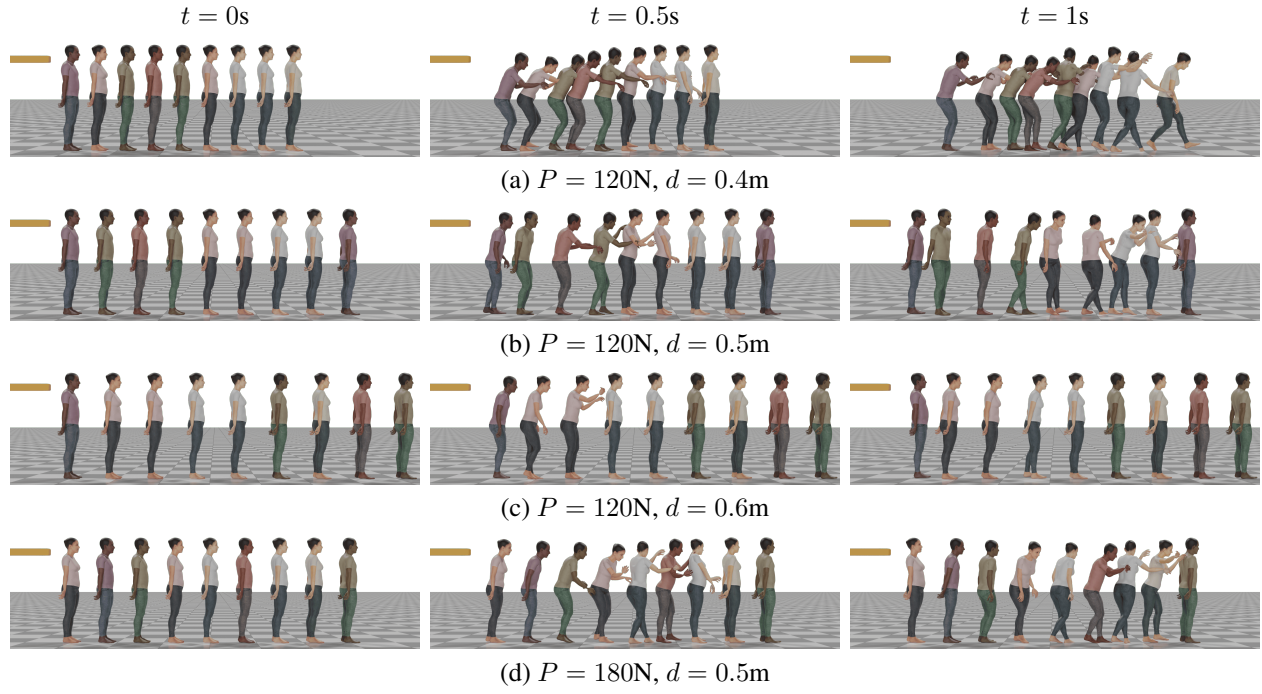


Figure S9. Simulation results of  $\pi_{\text{adapt}}$  over time for a line of people at varying peak strength  $P$  and interpersonal distance  $d$ .

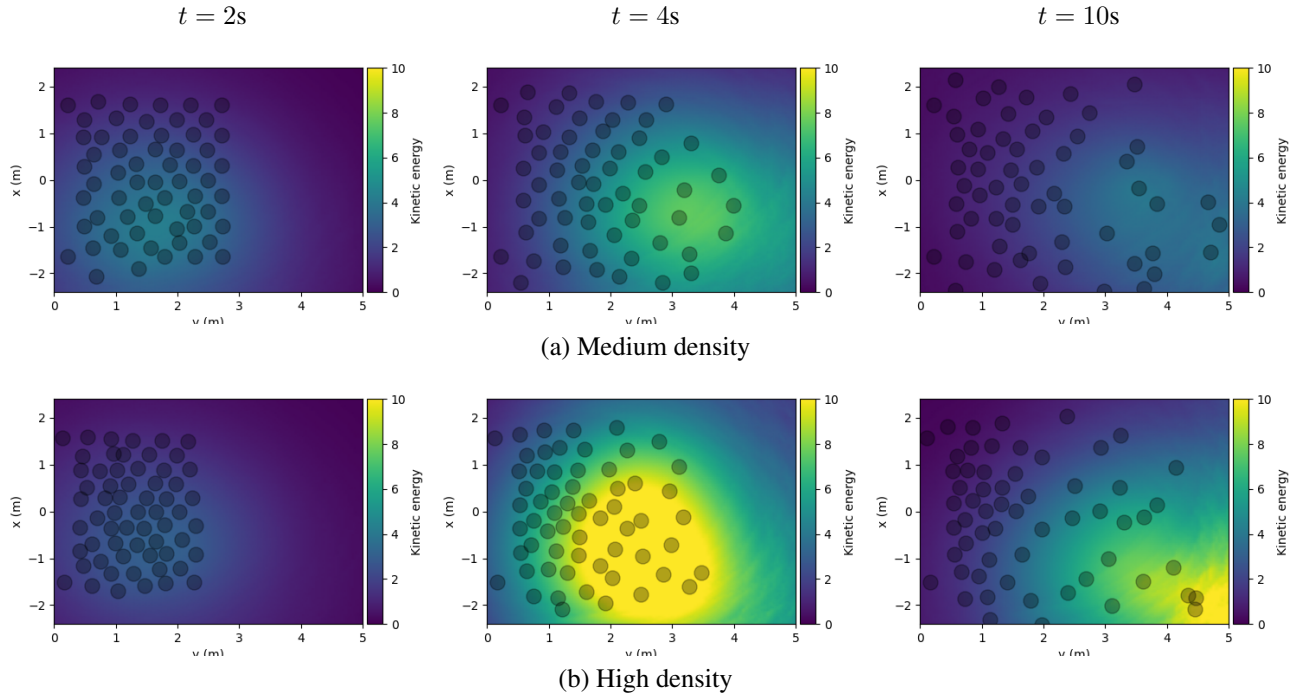


Figure S10. Simulation heatmap of kinetic energy for a push originating at  $(0,0)$ .

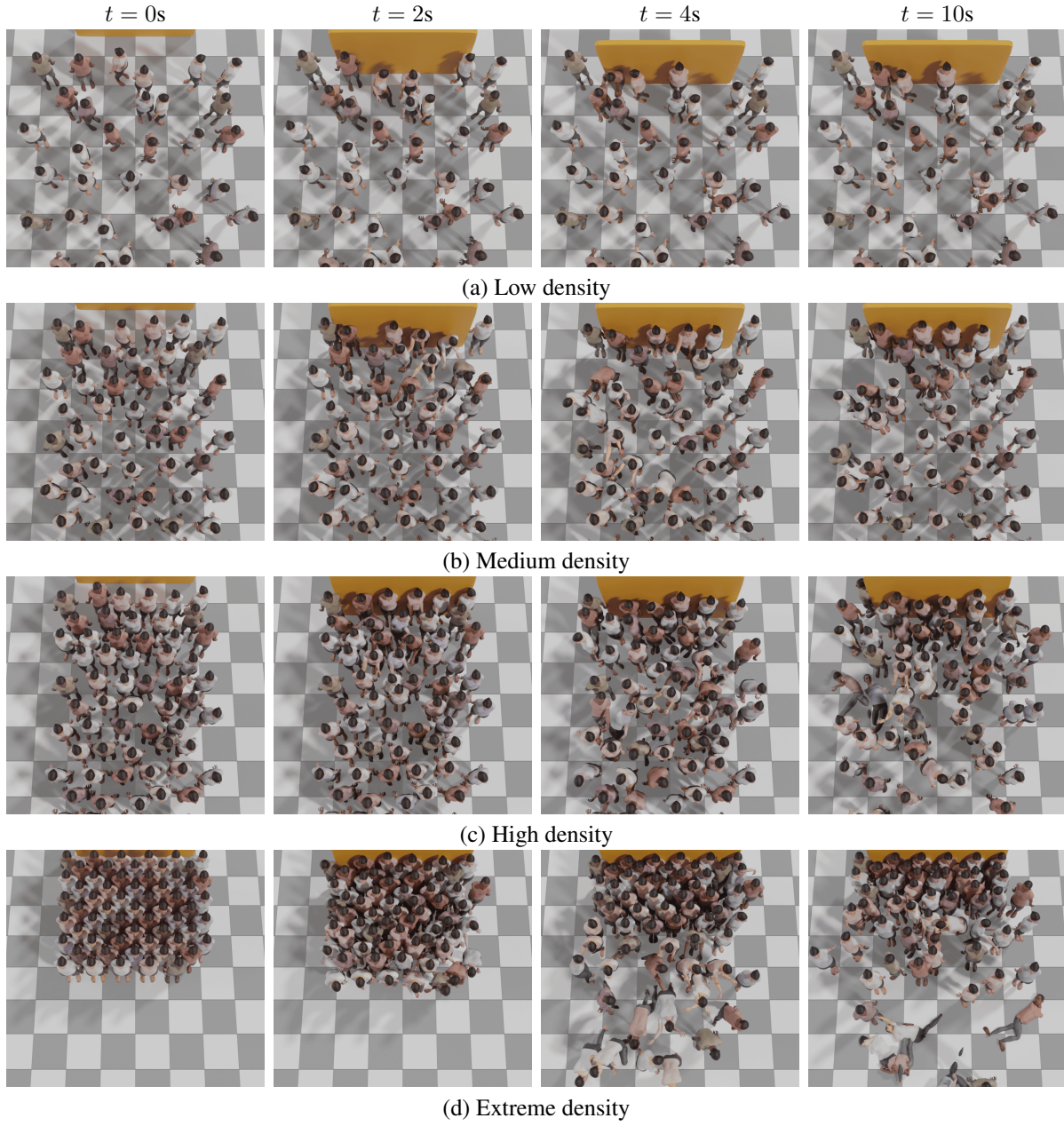


Figure S11. Simulation results of  $\pi_{\text{adapt}}$  for a crowd of people at varying densities responding to a push from a moving obstacle.

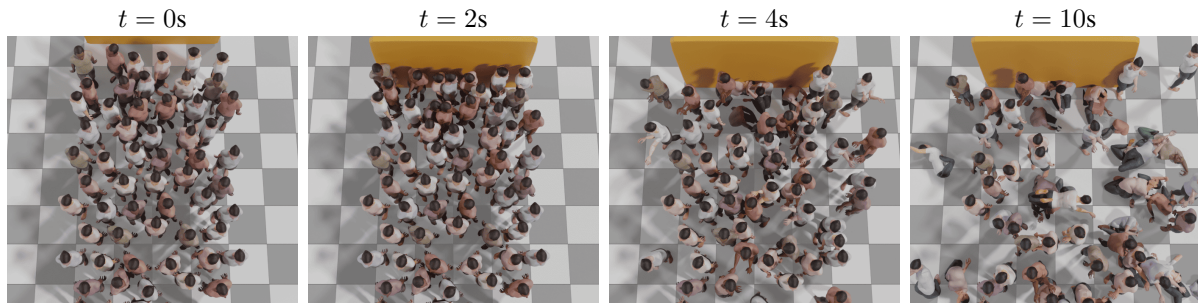


Figure S12. Simulation results of  $\pi_{\text{pretrain}}$  for a crowd of people at a medium density responding to a push from a moving obstacle.

## 2. Implementation Details

### 2.1. Training environment

Each simulated character has 15 body links with 34 degrees of freedom, where every joint is modeled as a 3-dimensional spherical joint except for the elbow and knee, which are 1-dimensional revolute joints. Each joint relies on a PD Controller to achieve the target goal provided by the policy. The physics simulation runs at 600Hz, while the policy runs at 30Hz. We use PyBullet [?] as the backend physics simulator in our implementation. The reference balance dataset was captured by a single person who consented beforehand to be pushed to perform the push recovery motions.

Policy optimization is done through distributed PPO (DPPO) [?] and uses the Adam optimizer [?] for neural network optimization. Training is performed on a dual RTX2080 GPU setup, with 8 environment instances running in parallel. A batch size of 1024 was used with an actor learning rate of 1e-5, a critic learning rate of 1e-4, and a discriminator learning rate of 1e-5. An analytic entropy term with a coefficient of 1e-4 is also used in the loss function to encourage policy exploration. Training sequences are terminated and penalized by setting the reward to 0 if the controller agent falls to the ground. Rewards in the implementation have empiric scaling factors applied to the physical quantities measured in the simulation, which are shown in Table S2. Applying the heuristic in Section 5.2 yields one target per hand. For each target, the agent’s state vector is augmented with the target’s shoulder position and orientation, along with the root position of the corresponding neighbor. This results in an additional input state in  $\mathbb{R}^{10 \times 2}$  during the adaptation training.

Reward units and scaling factors			
	Unit	Factor	Related equations
Distance	m	10	Eq. 3, 8, 10
Angle error	rad	5	Eq. 5, 8, 10
Speed	$m.s^{-1}$	5	Eq. 5
Kinetic energy	J	5	Eq. 5
Force	N	1	Eq. 9

Table S2. Units and scaling factors of the physical quantities used in the pretraining and adaptation rewards.

### 2.2. Heuristic algorithm

We elaborate on the heuristic of target hand selection in Algorithm 1. In lines 1 to 10, as seen in Figure S13, we go through every shoulder in the scene and compute its current position  $A$  and future position  $B$ . This is done using the linear momentum  $\vec{L}$ , which reflects the force  $F$  being applied to the agent. For each shoulder, the goal is to find the closest collision shoulder  $s_{col}$  and the best support shoulder  $s_{sup}$ .

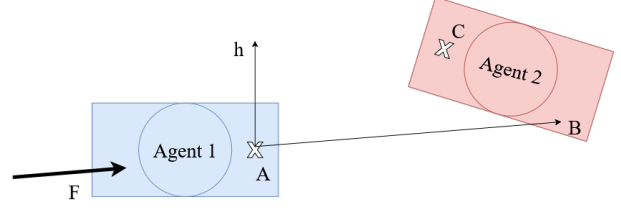


Figure S13. Physical quantities used to predict the shoulder trajectory of the main agent and evaluate candidate shoulders from neighboring agents.

To find them, we compute several physical quantities in lines 11 to 19, used for the selection process in Figures S14 and S15. Every other shoulder is scanned, and their position  $C$  is used to compute 4 different values. We begin by computing  $d_1$ , which is the closest distance among every point along the trajectory  $\vec{AB}$  and  $C$ . This is done by computing the distance  $t$ , the projection of  $\vec{AC}$  on the line  $AB$  divided by the length of the vector  $\vec{AB}$  and clamped between 0 and 1. We then compute  $d_2$ , which is simply the distance between the final position of the shoulder  $B$  and the current position of the candidate shoulder  $C$ . We make use of the heading direction of the main agent  $\vec{h}$ , computed using the torso orientation. It allows us to compute  $c_1$ , the collinearity between the heading direction and the direction of the candidate shoulder. We also compute  $c_2$ , the collinearity between the push direction and the direction of the candidate shoulder.

As seen in Figure S14, lines 20 to 23 check for the closest collision risk. Multiple conditions need to be met. First, the closest distance between the main shoulder trajectory and the candidate shoulder,  $d_1$ , must be less than 0.25 (about a half-torso width). The collinearity  $c_1$  between the heading direction and the candidate shoulder direction must be positive, as we only consider shoulders in front of the agent. Finally, the distance to the shoulder  $\|\vec{AC}\|$  must be the minimal among all candidates, which is done using  $min_{col}$ . After scanning every candidate shoulder,  $s_{col}$  is the closest collision risk among them.

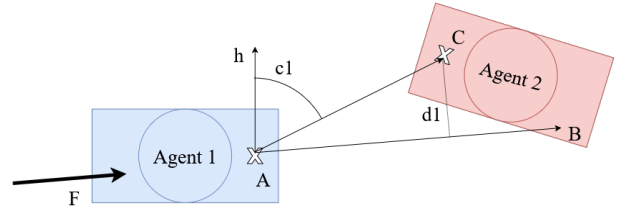


Figure S14. Physical quantities used to detect collision risks with neighboring shoulders.

As seen in Figure S15, lines 24 to 28 check for support shoulders, keeping the most efficient one. First, the candidate shoulder must be reachable. This is checked using  $d_2$



being under 0.6. This corresponds to a bit less than an average arm length, as we don't want to do contacts with a fully stretched arm, but instead with leverage [? ]. Like before, the collinearity  $c_1$  between the heading direction and the candidate shoulder direction must be positive, as we only consider shoulders in front of the agent. Only values of  $c_2$  above 0.02 are kept to filter out valid but inefficient supports, using  $max_{col}$ . After scanning every candidate shoulder,  $s_{sup}$  is the most efficient balance support among them. Lines 29-36 make the final selection: if a collision risk is found, it becomes the target. Otherwise, if a support shoulder is found, it becomes the target. Otherwise, no target is set, defaulting to targeting the side of the body at hip-level.

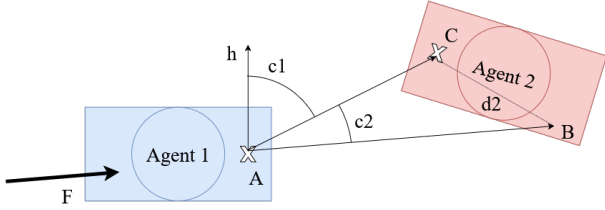


Figure S15. Physical quantities used to select reachable and efficient support shoulders for hand contacts.

---

**Algorithm 1** Hand targets algorithm

---

```

1:  $S$  = List of shoulders in the scene, each shoulder  $s$  with
   its position  $s_{pos}$  and corresponding agent  $s_{agent}$ 
2: for  $s^1 \in S$  do
3:    $a = s^1_{agent}$ 
4:    $\vec{L} = \frac{1}{M} \sum_{l \in a_{limbs}} (m_l \vec{v}_l)$ ,  $M = a_{mass}$ 
5:    $A = s^1_{pos}$ 
6:    $B = s^1_{pos} + \vec{L}$ 
7:    $min_{col} = \infty$ 
8:    $s_{col} = \text{None}$ 
9:    $max_{sup} = 0.02$ 
10:   $s_{sup} = \text{None}$ 
11:  for  $s^2 \in S$  with  $s^1_{agent} \neq s^2_{agent}$  do
12:     $C = s^2_{pos}$ 
13:     $t = \frac{\vec{AC} \cdot \vec{AB}}{\vec{AB} \cdot \vec{AB}}$ 
14:     $t = \text{CLAMP}(t, 0, 1)$ 
15:     $d_1 = \|A + t \frac{\vec{AB}}{\|\vec{AB}\|} - \vec{AC}\|$ 
16:     $d_2 = \|\vec{BC}\|$ 
17:     $\vec{h}$  is the heading direction of  $s^1_{agent}$ 
18:     $c_1 = \vec{h} \cdot \vec{AC}$ 
19:     $c_2 = \vec{L} \cdot \vec{AC}$ 
20:    if  $d_1 < 0.25$  &  $c_1 > 0$  &  $\|\vec{AC}\| < min_{col}$ 
      then
21:       $min_{col} = \|\vec{AC}\|$ 
22:       $s_{col} = s^2$ 
23:    end if
24:    if  $d_2 < 0.6$  &  $c_1 > 0$  &  $c_2 > max_{sup}$  then
25:       $max_{sup} = c_2$ 
26:       $s_{sup} = s^2$ 
27:    end if
28:  end for
29:  if  $s_{col}$  is not None then
30:     $s^1_{target} = s_{col}$ 
31:  else if  $s_{sup}$  is not None then
32:     $s^1_{target} = s_{sup}$ 
33:  else
34:     $s^1_{target} = \text{default}$ 
35:  end if
36: end for

```

---