CLOSE ENCOUNTERS: DECENTRALIZED MULTI-AGENT POLICY LEARNING FOR CONTACT-RICH EMBODIED INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Contact-rich interaction is central to physical assistance and close cooperation, yet poses a challenging coordination problem: agents must make intentional contact, avoid interpenetration, and continually adapt to one another while using only partial, egocentric observations. We formulate this problem as a decentralized multi-agent control task involving two virtual human agents. Each agent has access to partial egocentric observations and fine-grained contact goals specified by body-surface marker pairs. Collaboration emerges through online multi-agent reinforcement learning (MARL) of a single policy that serves both agents across all tasks. We learn this coordination without captured two-person interaction data by introducing a shared latent action space for planning and control. Both agents operate in this space, which is learned from single-human motion data. The same representation enables us to generate synthetic interaction demonstrations through centralized trajectory optimization in the joint latent action space of the two agents, and use these demonstrations to bootstrap online RL. Across nine tasks with variation in body shape and initial pose, this combination increases success over online RL from 36% to 83% on two-contact tasks and from 23% to 75% on four-contact tasks.

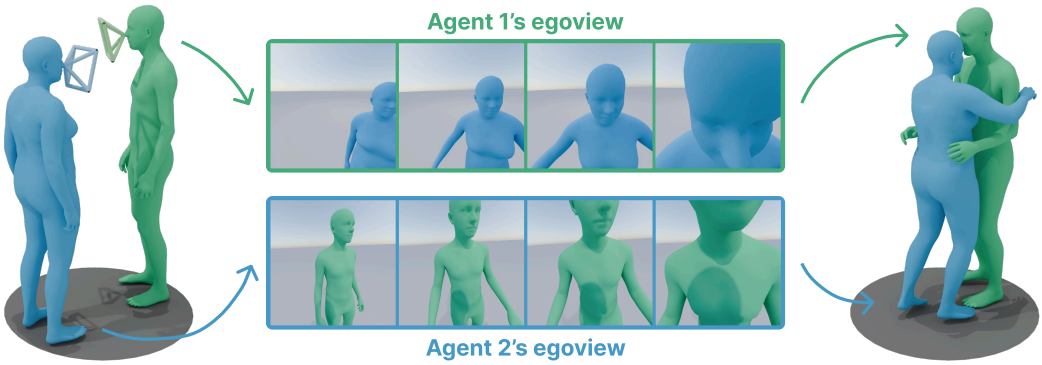


Figure 1: **Problem overview.** We study contact-rich interactions between agents with different embodiments. We learn perception–action policies where each agent acts based on egocentric observations and adapts online to its partner in order to achieve shared contact goals.

1 INTRODUCTION

Contact-rich interaction poses a challenging coordination problem for embodied agents. Establishing contact without penetration requires each agent to adjust its movement to a partner whose actions also depend on the ongoing interaction. Multiple simultaneous contact goals, differences in body shape, and partial observations of the partner compound these challenges. Assistive care illustrates this challenge: a carer must continuously monitor and react to a patient's motion in real time, using only partial egocentric perception while making intentional and penetration-free contact.

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

Existing approaches to HHI modeling are not well-suited to the embodied interaction setting. Generative approaches can synthesize coordinated motion for multiple participants (Sui et al., 2026), however, jointly generating both trajectories does not by itself provide a closed-loop controller that acts on one participant from partial observations of the others. At the same time, learning interaction policies directly from demonstration data is severely constrained by data availability. Datasets involving close bodily contact are scarce (Khirodkar et al., 2024; Yin et al., 2023), difficult to scale (Fieraru et al., 2020; Xu et al., 2024a; Saadatnejad et al., 2025), and typically provide skeleton-level rather than mesh-level supervision despite the importance of surface contact and penetration. More fundamentally, a general interaction policy must handle variations in body shape, initial pose, and coordination strategy. As we show experimentally, behavior cloning fails even when trained with multiple synthetic demonstrations, as closed-loop execution suffers from covariate shift and compounding errors (Ross et al., 2011).

In our work, we formulate contact-rich embodied interaction as a cooperative decentralized multi-agent control problem. We develop a single contact-conditioned policy shared across both agents and all interaction tasks, with each agent acting from partial egocentric marker observations and its specified contact pairs. We learn this policy with online Multi-Agent Reinforcement Learning, and use shared rewards to encourage the specified contacts while penalizing interpenetration, and per-agent rewards to encourage plausible motion.

We study the control and coordination challenges of real-world embodied interaction through the proxy of kinematic simulation of virtual human interaction Sec. 3.1. This provides a practical and scalable setting for training and evaluating closed-loop interaction policies without the safety and data-collection barriers of real embodied interaction. Closed-loop evaluation is essential because small errors at individual time steps can compound as the interaction unfolds. Virtual humans also enable controlled variation in body shape and initial pose, while vertex-level contact goals provide fine-grained task specifications for diverse body-contact interactions. Despite its abstraction from real-world dynamics, this setting preserves the core algorithmic structure of embodied interactions such as human-robot assistance: decentralized coordination, body-surface contact, partial observability, and mesh-level collision avoidance.

To learn this policy without captured two-person interaction data, in Sec. 3.2 we use a latent action space learned from single-human motion. A causal VAE decoder supports both online policy execution by individual agents and centralized optimization of two-agent trajectories, allowing us to generate demonstrations directly in the policy's action space through trajectory optimization. Behavior cloning of these demonstrations initializes both the actor and critic, providing structured interaction behavior before online MARL exposes the policy to errors and states encountered during closed-loop execution.

Across nine tasks, demonstration-based initialization substantially improves online learning: success increases from 36% to 83% on two-contact tasks and from 23% to 75% on four-contact tasks compared with PPO from scratch. We further analyze training configuration, visibility, and variation in body shape and initial pose (Sec. 4). Together, these results demonstrate that a shared motion latent space can bridge centralized planning and decentralized control, enabling agents to learn coordinated body-surface contact from partial observations without captured two-person interaction demonstrations. [1]

2 Related work

Human-human interaction datasets support data-driven modeling of two-person motion (Fieraru et al., 2020; Xu et al., 2024a; Khirodkar et al., 2024; Yin et al., 2023; Liang et al., 2024). Close-contact datasets with mesh-level geometry remain comparatively small: Hi4D contains 20 subject pairs and 100 sequences (Yin et al., 2023), while Harmony4D contains 24 subjects and 208 video sequences (Khirodkar et al., 2024). Chi3D (Fieraru et al., 2020) and Inter-X (Xu et al., 2024a) provide motion-capture trajectories, and HHI Assist (Saadatnejad et al., 2025) focuses on physical assistance. These resources provide valuable interaction examples, but limited coverage of the body

[1] Videos are available on the anonymous gallery webpage. All training and evaluation code, synthetic demonstrations, and checkpoints will be made publicly available.

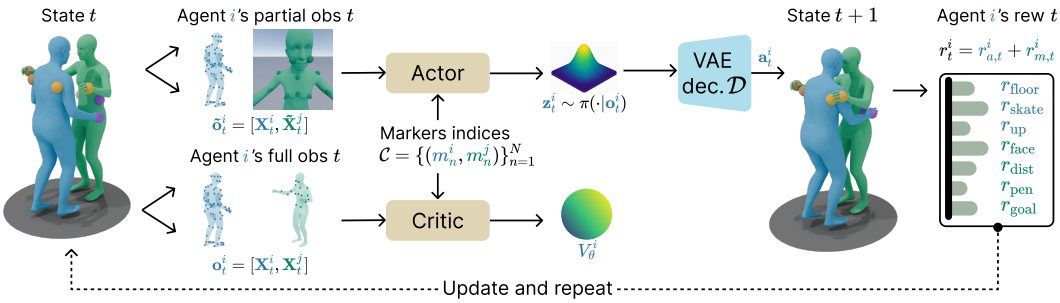


Figure 2: **Decentralized execution with centralized training.** Both agents share the same contact-conditioned actor network, predicting latent actions from partial egocentric marker observations. A causal motion decoder updates each body. During training, a privileged critic and shared contact rewards support cooperative policy learning.

shapes, poses, and surface-contact goals considered here. We instead combine single-human motion data with synthetic interaction demonstrations.

Human-human interaction synthesis is well studied (Sui et al., 2026). Actor–reactor methods generate one person's motion conditioned on a partner (Men et al., 2022; Xu et al., 2024b; Ghosh et al., 2024; Chopin et al., 2023; Ye et al., 2026). Other methods jointly generate interactions from class or text conditions (Goel et al., 2022; Tanaka & Fujiwara, 2023; Xu et al., 2023; Maheshwari et al., 2022; Xu et al., 2024a; Liang et al., 2024; Shafir et al., 2024; Fan et al., 2024; Lei et al., 2026; Manjotho et al., 2026). Autoregressive approaches support temporally extended generation (Javed et al., 2025; Ruiz-Ponce et al., 2026; Jiang et al., 2025; Maluleke et al., 2025; Li et al., 2024b; Ghosh et al., 2025). Two methods are particularly relevant. InterControl (Wang et al., 2024) uses single-human motion data and joint-distance constraints to synthesize interaction sequences, but its coupled sequence generation does not directly implement control from each agent's partial observations. Ready-to-React (Cen et al., 2025) already supports online per-agent reactions with a shared policy, but learns from paired interaction motions and conditions on motion histories of both, fully-observed agents. Adapting either to our benchmark would require changes to its observation and goal interface, including body-surface contact specifications. We focus on learning these goal-conditioned behaviors from synthetic demonstrations and online interaction.

Contact-rich multi-character simulation includes trajectory optimization (Liu et al., 2006; Mordatch et al., 2012) and multi-agent RL for coordinated humanoid behavior (Bansal et al., 2018; Won et al., 2021; Haarnoja et al., 2024). Recent work studies humanoid interaction, retargeting, and reusable skills (Liu et al., 2025; Chen et al., 2026; Huang et al., 2025). AssistMimic (Shibata et al., 2026) learns interacting policies to track assistive motion demonstrations in a physics simulator. Its reference-tracking objective and dynamics differ from our kinematic, contact-goal benchmark, so its reported results are not directly comparable. Our experiments instead compare learning strategies under a common environment, observation model, and action space.

Egocentric control and imitation learning. EgoGen (Li et al., 2024a) and CLOPS (Diomataris et al., 2025) explore perception-based control for individual virtual humans. We consider two agents whose actions jointly determine the contact geometry and future observations. Diffusion Policy (Chi et al., 2023) offers expressive, multimodal imitation policies and is a relevant direction for adapting our demonstration learner. It is not an existing solution to this benchmark: a comparison would require training a contact-conditioned, decentralized variant on the same demonstrations. Our present comparisons isolate demonstration-based training and online refinement within a common policy.

3 Method

Problem formulation. We consider two agents that must establish specified body-surface contacts while avoiding interpenetration. Each controls its own body and acts from a partial observation of its partner, making the task a cooperative multi-agent control problem. In Fig. 2, agent $i \in \{1, 2\}$ observes $\tilde{\mathbf{o}}_t^i = f^i(\mathbf{s}_t)$ and samples $\mathbf{z}_t^i \sim \pi_\theta(\cdot \mid \tilde{\mathbf{o}}_t^i, \mathcal{C}^i)$, where $\mathcal{C}^i$ specifies the desired contact pairs from its perspective. The joint action $\mathbf{z}_t = [\mathbf{z}_t^1, \mathbf{z}_t^2]$ advances the environment to $\mathbf{s}_{t+1}$.

During training the agents receive a shared centralized reward $r^c(\mathbf{s}_{t+1}, \mathbf{z}_t^1, \mathbf{z}_t^2)$ for their interaction quality, alongside with individual rewards $r^i(\mathbf{s}_{t+1}, \mathbf{z}_t^i),\ i \in \{1, 2\}$ for their motion quality. The objective of policy learning is to maximize the cumulative expected reward of the two agents:

$$(\pi_1^*, \pi_2^*) = \arg\max_{\pi_1, \pi_2} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1}(r_t^c + r_t^1 + r_t^2)\right]. \tag{1}$$

We follow MAPPO (Yu et al., 2022), sharing the policy parameters across agents $\pi_1 = \pi_2 = \pi_\theta$, while maintaining decentralized execution. The episode's contact specification $\mathcal{C}$ contains marker identities on the agent's own body and its partner's body. Different observations and contact goals allow this shared network to produce complementary behaviors, including asymmetric interactions. We use Proximal Policy Optimization (PPO) (Schulman et al., 2017) for online policy learning, with the objective: $\mathcal{L}_{\text{PPO}} = \mathcal{L}_{\text{CLIP}} + c_1\mathcal{L}_{\text{VF}} + c_2 S_\pi$, where $\mathcal{L}_{\text{CLIP}}$ is the policy surrogate loss, $\mathcal{L}_{\text{VF}}$ the value function loss, and S_π the entropy bonus. The actor and critic use separate observation encoders. During training, following (Pinto et al., 2018), the critic receives a privileged full observation $\mathbf{o}_t^i$, including the partner's full body markers, while the actor receives partial egocentric observations $\tilde{\mathbf{o}}_t^i$. Only the actor is used at execution time.

We next describe the mesh-based interaction environment and rewards (Sec. 3.1), the causal motion action space (Sec. 3.2), and demonstration-based initialization (Sec. 3.2).

3.1 Mesh-based interaction environment

Mesh-based environment. We describe every agent by their body mesh, which allows us to define fine-grained vertex-to-vertex scenarios and helps us measure and eventually avoid agent collisions and penetrations. For this, we use the SMPL model (Loper et al., 2015) which uses body shape $\boldsymbol{\beta}$, articulated pose $\boldsymbol{\theta}$, and global translation $\mathbf{T}$ parameters to synthesize a high-resolution mesh $S = \text{SMPL}(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{T}) \in \mathbb{R}^{6890\times 3}$. We randomly sample each agent's body shape $\boldsymbol{\beta}$, root orientation $\boldsymbol{\theta}_{\text{root}}$, and arm angles $\boldsymbol{\theta}_{\text{arms}}$ in order to model realistic conditions and train robust policies. We express agent interaction through the relationship of their meshes S^1, S^2, while adding a temporal context of F frames that provides kinematic information per-agent: $\mathbf{S}_t^i = \left[S_{t-F+1}^i, \ldots, S_t^i\right]$. The fully-observed environment state: $\mathbf{s}_t = \{\mathbf{S}_t^1, \mathbf{S}_t^2\}$, is used internally for simulation, reward computation, and planning, but is not directly accessible to the agents. In order to obtain a compact input to the policy network, we first reduce the high-resolution mesh S to a sparse semantic marker set by applying a fixed marker-selection operator: $\mathbf{m} = \Pi_{\mathcal{M}}(S) \in \mathbb{R}^{M\times 3}$, using the $M = 69$ markers used in (Zhang et al., 2021). As shown in Fig. 2, the policy input consists of two components: the agent's own body markers $\mathbf{X}^i$ (blue) and the partner's markers, expressed in the agent's local coordinate frame, $\tilde{\mathbf{X}}^j$ (green). The actor has only access to egocentric partial observations through the visible partner markers $\tilde{\mathbf{o}}^i = f^i(\mathbf{s})$. The critic, only used at training time, has access to privileged information consisting of the partner's full marker set $\mathbf{o}^i$.

Marker visibility. We attach a virtual pinhole camera to the head of agent i, aligned with its gaze direction and with a 90° field of view. A partner marker m_k^j is considered visible if (i) it lies in front of the camera, within its viewing frustum, and (ii) it faces the camera. Specifically for (ii), if $\mathbf{p}_k$ is the marker position in camera coordinates and $\mathbf{n}_k$ is the surface normal at the marker, we require $\mathbf{n}_k^\top \mathbf{p}_k < 0$, which ensures that the surface normal points toward the camera.

Interaction scenario and rewards. An interaction task is defined by a set of target marker pairs $\mathcal{C} = \{(m_n^1, m_n^2)\}_{n=1}^N$, where $N = 1$ denotes simple contact and larger N corresponds to multi-contact interactions such as embraces. An episode is successful if all target contacts are simultaneously satisfied (within a distance threshold τ) and no penetration occurs (within a tolerance δ) throughout the episode. Episodes terminate if the task is not completed within the maximum number of timesteps. We decompose the training reward into common interaction r_t^c and per-agent motion r_t^i rewards:

$$R_t^i = \underbrace{\lambda_{\text{goal}} r_{\text{goal},t} + \lambda_{\text{pen}} r_{\text{pen},t} + \lambda_{\text{dist}} r_{\text{dist},t} + \lambda_{\text{face}} r^i_{\text{face},t}}_{r_t^c} + \underbrace{\lambda_{\text{floor}} r^i_{\text{floor},t} + \lambda_{\text{skate}} r^i_{\text{skate},t} + \lambda_{\text{up}} r^i_{\text{up},t}}_{r_t^i}. \tag{2}$$

The interaction reward r_t^c in Eq. 2 captures the task-specific structure of our setting, depending on the joint state and actions of both agents, $r_t^c = r^c(\mathbf{s}_{t+1}, \mathbf{z}_t^1, \mathbf{z}_t^2)$. The completion reward r_{goal} is a sparse

reward delivered upon episode success, while r_{dist} is a dense reward, measuring the rate of decrease of distance between target marker pairs at every step. r_{face} encourages agents to face each other. The interpenetration reward r_{pen} promotes mesh-level collision avoidance, based on the signed distance ϕ^j of agent i to the partner body: $r_{\text{pen}} = -\sum_{i \neq j \in \{1,2\}} \sum_{k=1}^{M} \text{softplus}\Big(\alpha\big(\delta - \phi^i(m_k^j)\big)\Big)$, where m_k^j is a marker of agent j, δ is the penetration tolerance margin, and α sets the sharpness. This gives a differentiable penetration loss that later enables gradient-based trajectory optimization.

The per-agent r_t^i rewards in Eq. 2 depend only on agent's state and action $r_t^c = r^c(\mathbf{s}_{t+1}, \mathbf{z}_t^i)$, and are generic terms to regularize and encourage physically plausible motion where r_{floor} penalizes floor penetration and floating to enforce physical grounding, r_{skate} discourages foot sliding, and r_{up} encourages the agent to stand upright. We provide thorough descriptions of the reward terms in the supplementary material.

3.2 A Shared Latent Space for Control and Demonstration Generation

Action space design We design the action space of our policies to reflect prior knowledge about human motion available in single-human motion capture datasets. This encourages plausible local motions and acts as a good preconditioner for policy learning and trajectory optimization. For this, we train a 1D convolutional VAE (Jiang et al., 2023; Zhang et al., 2023) on the HumanML3D dataset (Guo et al., 2022). This provides us with a single-human motion decoder $\mathcal{D}$ that maps latents to realistic SMPL parameters.

Instead of predicting SMPL parameters directly, the policy acts in the VAE latent space: at each time step, agent i samples $\mathbf{z}_t^i \sim \pi^i(\cdot \mid \tilde{\mathbf{o}}_t^i)$, which is decoded into SMPL parameters via $\mathbf{a}_t = \mathcal{D}(\mathbf{z}_t)$.

Since execution is online, agents cannot condition on future latent codes or revise previously decoded poses. We therefore use *causal* 1D convolutions in the decoder, ensuring that only current and past latent actions influence the next SMPL parameter command. This enables online generation while preserving previously decoded poses and maintaining temporal consistency.

Synthetic demonstration data from trajectory optimization PPO can perform well on simple interactions but may struggle on more complex multi-contact tasks such as hugging. This motivates using planning-based demonstrations for difficult interaction cases.

Trajectory optimization. We generate interaction demonstrations by jointly optimizing both agents' latent trajectories. Gradients propagate through the motion decoder, mesh construction, and geometric contact and penetration costs.

For a horizon T, we collect both agent actions in $\mathbf{Z} = \{(\mathbf{z}_t^1, \mathbf{z}_t^2)\}_{t=1}^T$ and optimize $J(\mathbf{Z}) = -\sum_{t=1}^{T} \Big(r^c(\mathbf{s}_{t+1}, \mathbf{z}_t) + \sum_{i=1}^{2} r_t^i(\mathbf{s}_{t+1}, \mathbf{z}_t)\Big)$ with respect to $\mathbf{Z}$ (see Eq. 2). We update the latent trajectory using BFGS (Nocedal & Wright, 2006), computing gradients by differentiating through the full rollout.

We make optimization more robust with a continuation strategy on the α parameter of the penetration reward expression: early iterations use a softer penalty to find coarse approach trajectories, while later iterations sharpen the penalty to enforce penetration-free contact. This is useful for close-range interactions, where the agents must first reach a collision-free configuration before satisfying all target contacts. The resulting optimized latent-action sequences are executed in the environment and recorded as expert demonstrations.

Behavior cloning pretraining and PPO fine-tuning. Since large-scale human-human interaction demonstrations are not readily available for our setting, the optimized trajectories form our demonstration dataset $\mathcal{B}$. We use this dataset to pretrain both the actor and critic before online PPO fine-tuning (Ramrakhya et al., 2023; Kernbach et al., 2026). Actor pretraining uses partial observations and contact specifications; critic pretraining uses privileged observations.

For the actor, we regress the policy mean to the demonstrated latent action:

$$\mathcal{L}_{\text{BC}}^{\text{actor}} = \mathbb{E}_{(\tilde{\mathbf{o}}_t, \mathbf{z}_t) \sim \mathcal{B}} \left[\left\| \mu_\pi(\tilde{\mathbf{o}}_t) - \mathbf{z}_t \right\|^2 \right], \tag{3}$$

	Method	1-contact			2-contacts					4-contacts			
		H2H	H2B	**Avg.**	Low	Mid	Up	Cross	**Avg.**	Low	Up	Cross	**Avg.**
Success	Planner	0.98	0.76	0.87	0.31	0.73	0.94	0.50	0.62	0.69	0.59	0.55	0.61
	BC	0.28	0.05	0.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	PPO	0.85	0.83	0.84	0.01	0.65	0.67	0.10	0.36	0.23	0.00	0.47	0.23
	BC+PPO	0.94	0.95	0.95	0.82	0.88	0.88	0.74	0.83	0.74	0.74	0.78	0.75
Returns	Planner	3.39	2.44	2.92	1.23	2.58	3.76	1.83	2.35	1.89	1.85	1.58	1.77
	BC	2.99	2.10	2.54	1.50	1.72	1.77	1.63	1.65	1.55	1.67	1.52	1.58
	PPO	5.01	4.98	4.99	2.21	4.45	4.56	2.59	3.45	2.94	2.35	3.75	3.01
	BC+PPO	5.32	5.41	5.36	4.80	5.22	5.26	4.66	4.99	4.59	4.77	4.77	4.71

Table 1: **Performance comparison across baselines and environments.** We report the results over 500 evaluation episodes.

where $\mu_\pi(\tilde{\mathbf{o}}_t)$ denotes the mean of the stochastic policy $\pi(\cdot \mid \tilde{\mathbf{o}}_t)$. During pretraining, we optimize only the policy mean, since negative-log-likelihood training can lead to variance collapse. For the ensuing RL training we initialize the variance to a smaller value than in training from scratch so that early fine-tuning remains close to the pretrained behavior.
For the critic V^i of agent i we regress values to Monte Carlo returns from the demonstration rollouts (the critic reads the full observation $\mathbf{o}_t$):

$$\mathcal{L}_{\mathrm{BC}}^{\mathrm{critic}} = \mathbb{E}_{(\mathbf{o}_t, R_t)\sim\mathcal{B}}\left[\left(V_\theta^i(\mathbf{o}_t) - R_t^i\right)^2\right], \tag{4}$$

where $R_t^i = \sum_{k=0}^{T-t} \gamma^k \left(r_{t+k}^c + r_{t+k}^i\right)$ is the discounted return (Eq. 2). After behavior cloning, we fine-tune the decentralized policies with PPO using online rollouts. Overall, the final policy is decentralized and reactive, but benefits from centralized trajectory optimization during training.

4 Experiments

4.1 Experimental setup

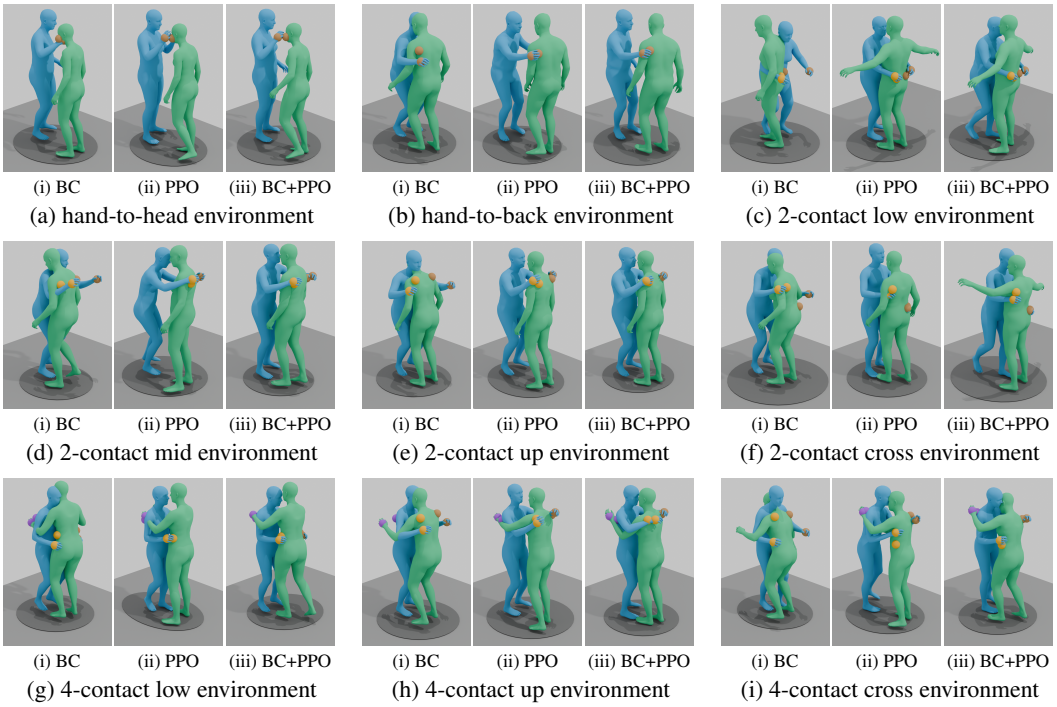
Environments. We study human–human interaction tasks of increasing complexity, with tasks with 1, 2, and 4 contact points. Single-contact tasks include hand-to-head (H2H) and hand-to-back (H2B), while multi-contact settings consist of 2- and 4-contact hugging scenarios with multiple variations (e.g., low, up, mid, cross) that capture increasingly complex and coordinated interactions. We show examples in Fig. 4, with additional illustrations and videos in the supplementary material.

Methods. We compare three policy-training strategies under a common observation and action interface: behavior cloning (*BC*) from planner demonstrations, *PPO* from random initialization, and demonstration-based initialization followed by PPO (*BC+PPO*). Each uses one contact-conditioned policy across both agents and all nine tasks. The demonstration planner provides a centralized reference with full-state access and joint trajectory optimization. BC-based methods use 500 demonstrations per task (4,500 total).

We evaluate success over 500 episodes, requiring all target pairs to be within 10 cm simultaneously and penetration to remain within 2.6 cm throughout the episode. We provide further details in the supplementary material.

4.2 Baseline comparison

Quantitative results. Tab. 1 shows how success and return vary as the number of required contacts increases. The *planner* achieves a high success on 1-contact tasks, with an average of 0.87, while maintaining robust performance on 2- and 4-contact tasks, with average success of 0.62 and 0.61, respectively. However, its returns are lower than those of BC+PPO, e.g., 2.92 compared with 5.36 on 1-contact tasks. It suggests that the planner may exploit the cost function in ways that achieve the task objectives but result in suboptimal behaviors. Also, the *planner*



(a) hand-to-head environment (b) hand-to-back environment (c) 2-contact low environment

(d) 2-contact mid environment (e) 2-contact up environment (f) 2-contact cross environment

(g) 4-contact low environment (h) 4-contact up environment (i) 4-contact cross environment

Figure 4: **Qualitative comparison of baselines.** We compare the final poses for the three baselines (BC, PPO, BC+PPO) across all nine environments.

jointly optimizes both trajectories with full-state access, providing a centralized reference for the decentralized policies, but requiring significant computational cost due to the optimization. *BC* achieves 0.28 success on hand-to-head and 0.05 on hand-to-back, averaging 0.17 across 1-contact tasks, and records no successful episodes on multi-contact tasks. The relatively higher performance on hand-to-head may be attributed to the relative simplicity of this task, which requires less complex behavior. However, the small number of demonstrations provides limited coverage of the random initializations, making the learned policy non-robust to different initial conditions and leading to poor performance on more complex tasks. *PPO* learns only through exploration, achieving average success rates of 0.84, 0.36, and 0.23 on 1-, 2-, and 4-contact tasks, respectively. The comparison with BC+PPO measures the benefit of demonstration-based initialization, with gains in average success of 11, 47, and 52 pp, respectively. *BC+PPO* achieves 0.95, 0.83, and 0.75 average success across the three contact groups. It improves over BC and exceeds the planner's average success in each group, despite acting from partial observations. On 4-contact tasks, for example, success increases from the planner's 0.61 to 0.75. Its corresponding returns are 5.36, 4.99, and 4.71. These show the value of combining synthetic demos with online refinement for multi-contact coordination.

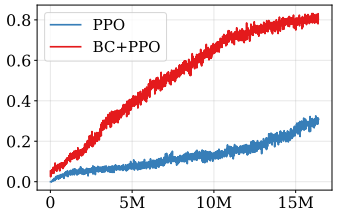


Figure 3: **Success vs training** averaged over the 9 tasks.

The learning curves in Fig. 3 complement the final scores by comparing learning progress. Demonstration-based initialization improves both sample efficiency and final performance. *BC+PPO* starts from a higher initial success rate ($\sim$0.05 versus 0 for *PPO*), reaches the final training success of *PPO* ($\sim$0.30) after about 3.3M environment steps, roughly $5\times$ fewer than the full 16.4M-step budget, and at that same budget it reaches $\sim$0.80 versus $\sim$0.30 for *PPO*.

Qualitative results. Fig. 4 illustrates contact alignment and body configurations across the nine tasks. BC rollouts exhibit inaccurate contacts and inter-agent penetration (e.g., Fig. 4.d.i). The PPO examples include penetration and awkward poses (Fig. 4.g.ii, Fig. 4.i.ii), while BC+PPO produces

	Training	1-contact			2-contacts					4-contacts			
		H2H	H2B	**Avg.**	Low	Mid	Up	Cross	**Avg.**	Low	Up	Cross	**Avg.**
Success	Per-task	0.97	0.94	0.96	0.90	0.91	0.97	0.92	0.92	0.84	0.81	0.91	0.85
	Independent	0.95	0.91	0.93	0.80	0.86	0.86	0.82	0.83	0.11	0.61	0.10	0.28
	Unified (ours)	0.94	0.95	0.95	0.82	0.88	0.88	0.74	0.83	0.74	0.74	0.78	0.75
Returns	Per-task	5.58	5.40	5.49	5.08	5.29	5.67	5.24	5.32	4.92	5.02	5.18	5.04
	Independent	5.36	5.29	5.33	4.76	5.15	5.22	4.92	5.01	2.55	4.40	2.57	3.17
	Unified (ours)	5.32	5.41	5.36	4.80	5.22	5.26	4.66	4.99	4.59	4.77	4.77	4.71

Table 2: **Influence of training config**. *Per-task*: one policy per task; *Independent*: no parameter sharing (Yu et al., 2022); *Unified (ours)*: single shared policy.

Visibility	1-contact		2-contacts		4-contacts	
	Succ.	Ret.	Succ.	Ret.	Succ.	Ret.
Full	0.97	5.44	0.85	5.05	0.80	4.87
Partial (ours)	0.95	5.36	0.83	4.99	0.75	4.71

Table 3: **Influence of partial visibility.**

Arms	Orient.	Shapes	1-contact		2-contacts		4-contacts	
			Succ.	Ret.	Succ.	Ret.	Succ.	Ret.
-	-	-	0.99	5.61	0.96	5.49	0.94	5.37
✓	-	-	0.99	5.61	0.97	5.54	0.92	5.33
-	✓	-	0.98	5.58	0.94	5.45	0.68	4.55
-	-	✓	0.98	5.49	0.91	5.26	0.86	5.09
✓	✓	✓	0.95	5.36	0.83	4.99	0.75	4.71

Table 4: **Influence of random initialization.**

coordinated contact configurations (Fig. 4.i.iii). In back-target tasks, the receiving agent raises its arms to let the partner pass underneath (Fig. 4.c.iii, Fig. 4.f.iii). Videos of successful and failed interactions are available on our gallery[2], and analysis in appendix.

4.3 Ablation study and sensitivity analysis

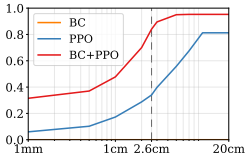
We study the effects of the training configuration (Tab. 2), partial visibility (Tab. 3), random initialization (Tab. 4), as well as the sensitivity to penetration and distance tolerances (Fig. 5, Fig. 6). Unless stated otherwise, these experiments use the unified BC+PPO policy.

Influence of training configuration. Tab. 2 separates two choices: unified model across tasks (*Per-task*), and parameter sharing across agents (Yu et al., 2022) (*Independent*, with separate networks for each agent). One unified policy achieves 95%, 83%, and 75% success across the three task groups, compared with 96%, 92%, and 85% for task-specific policies. While the unified policy shows a small decrease in performance compared to the task-specific policies, it has the advantage of covering all nine interactions with a single policy. It is also more training-efficient, as only one model needs to be trained instead of a separate model for each task. The unified policy achieves comparable or better performance than the independent policy, with higher success on 1-contact (0.95 vs. 0.93) and 4-contact tasks (0.75 vs. 0.28), while maintaining the same success on 2-contact tasks (0.83). This suggests that parameter sharing is beneficial, as the shared network can learn from samples across all task groups, effectively providing roughly twice as much training data per network compared to the independently trained policies.

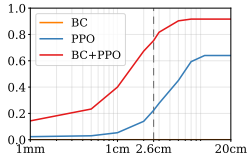
Influence of partial visibility. Tab. 3 compares full and partial visibility of partner markers, with success rates of 97%, 85%, and 80% for full observations versus 95%, 83%, and 75% for partial observations on 1-, 2-, and 4-contact tasks, respectively. This small performance drop under partial visibility suggests that the agent can infer the locations of unseen markers from the visible ones, remaining robust to random variations in partner shape.

Influence of environment random initialization. To analyze sensitivity to the initialization factors described in Sec. 3, we selectively enable arm, orientation, and body-shape variation and report results in Tab. 4. Comparing deterministic initialization with the fully varying setting (row 1 vs. row 5), success decreases from 99% to 95% for 1-contact tasks, from 96% to 83% for 2-contact

[2] https://close-encounters.github.io/close-encounters-gallery/

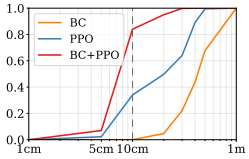


(a) 2-contacts

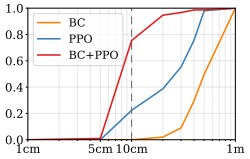


(b) 4-contacts

Figure 5: **Penetration tolerance δ sensitivity.**



(a) 2-contacts



(b) 4-contacts

Figure 6: **Distance threshold τ sensitivity.**

tasks, and from 94% to 75% for 4-contact tasks. The larger drop for multi-contact tasks shows that coordinating several contacts becomes more challenging when the starting configurations vary. The individual factors have different effects on performance. On 2-contact tasks, arm variation alone gives 97% success, orientation variation gives 94%, and shape variation gives 91%. It shows that orientation and shape variations have a larger impact, while arm variation has a smaller effect. When all factors are enabled, the policy achieves 83% success on 2-contact tasks, showing that the effects are not additive and that combining all variations leads to an even larger performance drop.

Influence of penetration tolerance δ and distance threshold τ. Fig. 5 and Fig. 6 show averaged success on 2- and 4-contact tasks as the penetration and contact-distance tolerances vary, respectively, while keeping the other criterion fixed at its default (10 cm and 2.6 cm, dashed lines). These results illustrate the sensitivity of success to the two contact criteria. As the penetration tolerance is tightened, success decreases gradually. At 1 mm, BC+PPO retains 31% and 14% success on 2- and 4-contact tasks, compared with 6% and 2% for PPO. With a relaxed tolerance, BC+PPO saturates at about 95% and 92% beyond $\sim$5 cm, while PPO plateaus at 81% and 64%. BC never succeeds and therefore remains at 0% across the tested penetration tolerances. For all methods, once the success rate plateaus as the penetration tolerance is relaxed, the remaining failures are due to missed contacts rather than penetration. Varying the contact-distance threshold has a stronger effect on BC+PPO. Its success increases sharply between 5 and 10 cm and exceeds 95% at 20 cm, suggesting that many failures at the default threshold correspond to near misses. PPO requires approximately 50 cm to reach a similar success level, while BC remains below 5% up to 20 cm and reaches 50% only around 40–50 cm. The advantage of BC+PPO over BC is therefore maintained across all tested thresholds below 1 m, reaching 90 and 93 percentage points at 20 cm.

5 Limitations

Our setting isolates geometric contact coordination with kinematic bodies and structured marker observations. It does not model contact forces, balance, or image-based perception. Connecting these policies to physics-based control (Shibata et al., 2026), visual sensing (Wu et al., 2026), and robot embodiments (Chen et al., 2026) is an important next step. The present tasks also focus on individual touches and embraces; longer assistive routines require composing such behaviors with higher-level goals (Sferrazza et al., 2024). Remaining failures include missed contacts and interpenetration. Our current policy is unimodal, and constructing the demonstration dataset requires manual rejection of unsuccessful planner trajectories. More expressive policies (Chi et al., 2023) and methods that use imperfect demonstrations (Yang et al., 2023; Wagenmaker et al., 2025; Zhou et al., 2025) could broaden the behaviors learned from our currently curated planner dataset.

6 Conclusion

We study close-contact interaction as cooperative control from partial egocentric observations. A shared motion latent space connects centralized planning to a single contact-conditioned policy used by both agents across nine tasks. Synthetic demonstrations provide an initialization, and online RL substantially improves success over behavior cloning. This setting isolates coordination requirements relevant to agents interacting with independently moving partners, including robots assisting people, while enabling controlled study of contact geometry, perception, and body variation.

REFERENCES

Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. In *ICLR*, 2018.

Zhi Cen, Huaijin Pi, Sida Peng, Qing Shuai, Yujun Shen, Hujun Bao, Xiaowei Zhou, and Ruizhen Hu. Ready-to-react: Online reaction policy for two-character interaction generation. In *ICLR*, 2025.

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

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Baptiste Chopin, Hao Tang, Naima Otberdout, Mohamed Daoudi, and Nicu Sebe. Interaction transformer for human reaction generation. *Transactions on Multimedia*, 2023.

Robin Courant, David Loiseaux, Xi Wang, Marc Christie, and Vicky Kalogeiton. Pulp motion: Framing-aware multimodal camera and human motion generation. In *ICLR*, 2026.

Markos Diomataris, Berat Mert Albaba, Giorgio Becherini, Partha Ghosh, Omid Taheri, and Michael J Black. Moving by looking: Towards vision-driven avatar motion generation. *arXiv preprint arXiv:2509.19259*, 2025.

Ke Fan, Junshu Tang, Weijian Cao, Ran Yi, Moran Li, Jingyu Gong, Jiangning Zhang, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. Freemotion: A unified framework for number-free text-to-motion synthesis. In *ECCV*, 2024.

Mihai Fieraru, Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, Vlad Olaru, and Cristian Sminchisescu. Three-dimensional reconstruction of human interactions. In *CVPR*, 2020.

Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Remos: 3d motion-conditioned reaction synthesis for two-person interactions. In *ECCV*, 2024.

Anindita Ghosh, Bing Zhou, Rishabh Dabral, Jian Wang, Vladislav Golyanik, Christian Theobalt, Philipp Slusallek, and Chuan Guo. Duetgen: Music driven two-person dance generation via hierarchical masked modeling. In *SIGGRAPH*, 2025.

Aman Goel, Qianhui Men, and Edmond SL Ho. Interaction mix and match: Synthesizing close interaction using conditional hierarchical gan with multi-hot class embedding. In *Computer Graphics Forum*, 2022.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, 2022.

Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H Huang, Dhruva Tirumala, Jan Humplik, Markus Wulfmeier, Saran Tunyasuvunakool, Noah Y Siegel, Roland Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, 2024.

Yiming Huang, Zhiyang Dou, and Lingjie Liu. Modskill: Physical character skill modularization. In *ICCV*, 2025.

Krishna Murthy Jatavallabhula, Edward Smith, Jean-Francois Lafleche, Clement Fuji Tsang, Artem Rozantsev, Wenzheng Chen, Tommy Xiang, Rev Lebaredian, and Sanja Fidler. Kaolin: A pytorch library for accelerating 3d deep learning research. *arXiv preprint arXiv:1911.05063*, 2019.

Muhammad Gohar Javed, Xingyu Li, et al. Intermask: 3d human interaction generation via collaborative masked modeling. In *ICLR*, 2025.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *NeurIPS*, 2023.

Wentao Jiang, Jingya Wang, Kaiyang Ji, Baoxiong Jia, Siyuan Huang, and Ye Shi. Arflow: Human action-reaction flow matching with physical guidance. *arXiv preprint arXiv:2503.16973*, 2025.

Andreas Kernbach, Amr Elsheikh, Nicolas Grupp, René Nagel, and Marco F Huber. Actor-critic pretraining for proximal policy optimization. *arXiv preprint arXiv:2602.23804*, 2026.

Rawal Khirodkar, Jyun-Ting Song, Jinkun Cao, Zhengyi Luo, and Kris Kitani. Harmony4d: A video dataset for in-the-wild close human interactions. *NeurIPS*, 2024.

Nan Lei, Yuan-Ming Li, Ling-An Zeng, Liang Xu, Zhi-Wei Xia, Hui-Wen Huang, Fa-Ting Hong, and Wei-Shi Zheng. Physigen: Integrating collision-aware physical constraints for high-fidelity human-human interaction generation. In *ICASSP*, 2026.

Gen Li, Kaifeng Zhao, Siwei Zhang, Xiaozhong Lyu, Mihai Dusmanu, Yan Zhang, Marc Pollefeys, and Siyu Tang. Egogen: An egocentric synthetic data generator. In *CVPR*, 2024a.

Siyao Li, Tianpei Gu, Zhitao Yang, Zhengyu Lin, Ziwei Liu, Henghui Ding, Lei Yang, and Chen Change Loy. Duolando: Follower gpt with off-policy reinforcement learning for dance accompaniment. In *ICLR*, 2024b.

Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *IJCV*, 2024.

Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021.

C Karen Liu, Aaron Hertzmann, and Zoran Popović. Composition of complex optimal multi-character motions. In *SIGGRAPH*, 2006.

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

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH*, 2015.

Shubh Maheshwari, Debtanu Gupta, and Ravi Kiran Sarvadevabhatla. Mugl: Large scale multi person conditional action generation with locomotion. In *WACV*, 2022.

Vongani H Maluleke, Kie Horiuchi, Lea Wilken, Evonne Ng, Jitendra Malik, and Angjoo Kanazawa. Diffusion forcing for multi-agent interaction sequence modeling. *arXiv preprint arXiv:2512.17900*, 2025.

Ali Asghar Manjotho, Tekie Tsegay Tewolde, Ramadhani Ally Duma, and Zhendong Niu. Coshmdm: Contact and shape-aware latent motion diffusion model for human interaction generation. *Transactions on Visualization and Computer Graphics*, 2026.

Qianhui Men, Hubert PH Shum, Edmond SL Ho, and Howard Leung. Gan-based reactive motion synthesis with class-aware discriminators for human–human interaction. *Computers & Graphics*, 2022.

Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. In *ACM TOG*, 2012.

Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer, 2006.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Mathis Petrovich, Or Litany, Umar Iqbal, Michael J Black, Gul Varol, Xue Bin Peng, and Davis Rempe. Multi-track timeline control for text-driven 3d human motion generation. In *CVPRW*, 2024.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *RSS*, 2018.

Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *CVPR*, 2023.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.

Pablo Ruiz-Ponce, Sergio Escalera, José García-Rodríguez, Jiankang Deng, and Rolandos Alexandros Potamias. Interact2ar: Full-body human-human interaction generation via autoregressive diffusion models. In *CVPR*, 2026.

Saeed Saadatnejad, Reyhaneh Hosseininejad, Jose Barreiros, Katherine Tsui, and Alexandre Alahi. Hhi-assist: A dataset and benchmark of human-human interaction in physical assistance scenario. *Robotics and Automation Letters*, 2025.

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

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

Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *ICLR*, 2024.

Yuto Shibata, Kashu Yamazaki, Lalit Jayanti, Yoshimitsu Aoki, Mariko Isogawa, and Katerina Fragkiadaki. Learning to assist: Physics-grounded human-human control via multi-agent reinforcement learning. *arXiv preprint arXiv:2603.11346*, 2026.

Kewei Sui, Anindita Ghosh, Inwoo Hwang, Bing Zhou, Jian Wang, and Chuan Guo. A survey on human interaction motion generation. *IJCV*, 2026.

Mikihiro Tanaka and Kent Fujiwara. Role-aware interaction generation from textual description. In *ICCV*, 2023.

Andrew Wagenmaker, Perry Dong, Raymond Tsao, Chelsea Finn, and Sergey Levine. Posterior behavioral cloning: Pretraining bc policies for efficient rl finetuning. *arXiv preprint arXiv:2512.16911*, 2025.

Zhenzhi Wang, Jingbo Wang, Yixuan Li, Dahua Lin, and Bo Dai. Intercontrol: Zero-shot human interaction generation by controlling every joint. *NeurIPS*, 2024.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Control strategies for physically simulated characters performing two-player competitive sports. In *ACM TOG*, 2021.

Zhen Wu, Xiaoyu Huang, Lujie Yang, Yuanhang Zhang, Koushil Sreenath, Xi Chen, Pieter Abbeel, Rocky Duan, Angjoo Kanazawa, Carmelo Sferrazza, et al. Perceptive humanoid parkour: Chaining dynamic human skills via motion matching. *arXiv preprint arXiv:2602.15827*, 2026.

Liang Xu, Ziyang Song, Dongliang Wang, Jing Su, Zhicheng Fang, Chenjing Ding, Weihao Gan, Yichao Yan, Xin Jin, Xiaokang Yang, et al. Actformer: A gan-based transformer towards general action-conditioned 3d human motion generation. In *ICCV*, 2023.

Liang Xu, Xintao Lv, Yichao Yan, Xin Jin, Shuwen Wu, Congsheng Xu, Yifan Liu, Yizhou Zhou, Fengyun Rao, Xingdong Sheng, Yunhui Liu, Wenjun Zeng, and Xiaokang Yang. Inter-x: Towards versatile human-human interaction analysis. In *CVPR*, 2024a.

Liang Xu, Yizhou Zhou, Yichao Yan, Xin Jin, Wenhan Zhu, Fengyun Rao, Xiaokang Yang, and Wenjun Zeng. Regennet: Towards human action-reaction synthesis. In *CVPR*, 2024b.

Hanlin Yang, Chao Yu, Siji Chen, et al. Hybrid policy optimization from imperfect demonstrations. *NeurIPS*, 2023.

Yaoqin Ye, Yiteng Xu, Qin Sun, Xinge Zhu, Yujing Sun, and Yuexin Ma. Remogen: Real-time human interaction-to-reaction generation via modular learning from diverse data. In *CVPR*, 2026.

Yifei Yin, Chen Guo, Manuel Kaufmann, Juan Jose Zarate, Jie Song, and Otmar Hilliges. Hi4d: 4d instance segmentation of close human interaction. In *CVPR*, 2023.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre M. Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. In *NeurIPS*, 2022.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *CVPR*, 2023.

Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *CVPR*, 2021.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019.

Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *ICLR*, 2025.

A AI use statement **14**

B Detailed method **14**

B.1 Detailed rewards 14

B.2 SDF calculation 15

C Detailed experiments **15**

C.1 Detailed architecture. 15

C.2 Detailed environments 17

C.3 Additional qualitative results 17

A AI USE STATEMENT

In this work, we used generative AI tools to assist with coding, writing, and the review of related work. These tools were used to support the research process, including code development and debugging, language editing, and identifying and organizing relevant literature. All AI-assisted outputs were reviewed and verified by the authors, who remain fully responsible for the content, correctness, and originality of the paper.

B DETAILED METHOD

B.1 DETAILED REWARDS

For each agent i , we decompose the reward into an *interaction* component r^c, and a *per-agent* component r^i:

$$r = \underbrace{\lambda_{\text{goal}} r_{\text{goal}} + \lambda_{\text{pen}} r_{\text{pen}} + \lambda_{\text{dist}} r_{\text{dist}} + \lambda_{\text{face}} r^i_{\text{face}}}_{r^c} + \underbrace{\lambda_{\text{floor}} r^i_{\text{floor}} + \lambda_{\text{skate}} r^i_{\text{skate}} + \lambda_{\text{up}} r^i_{\text{up}}}_{r^i}, \quad (5)$$

where each λ denotes a scalar weight, as detailed in Tab. 5.

Interaction rewards. To encourage agents to orient toward each other and avoid turning away, we define a facing reward based on their body orientations. Let $\mathbf{R}^i_z \in \mathbb{R}^2$ denote the normalized horizontal body orientation of agent i, computed from torso joints and projected onto the ground plane. The facing reward is defined as

$$r_{\text{face}} = \frac{1}{2}\left(1 - \mathbf{R}^{1\top}_z \mathbf{R}^2_z\right), \quad (6)$$

To encourage agents to reach their respective targets, we define a distance-based shaping reward. Let $\mathbf{X}^i_t \in \mathbb{R}^{M\times 3}$ denote the target markers of agent i at time t. The distance reward is defined as:

$$r_{\text{dist}} = \left\|\mathbf{X}^1_{t-1} - \mathbf{X}^2_{t-1}\right\|_2 - \left\|\mathbf{X}^1_t - \mathbf{x}^2_t\right\|_2. \quad (7)$$

To discourage interpenetration between agents, we use a signed distance field (SDF) formulation. Let ϕ_j be the SDF of agent i to the partner body, and m^j_k a marker of agent i. We define the penetration penalty given a penetration margin δ as:

$$r_{\text{pen}} = -\sum_{i\neq j\in 1,2}\sum_{k=1}^{M} \text{softplus}\Big(\alpha(\delta - \phi^i(m^j_k))\Big). \quad (8)$$

where α controls the penalty sharpness.

Finally, we define a success indicator that captures task completion, requiring both target proximity (τ) and the absence of significant interpenetration (δ):

$$r_{\text{goal}} = \mathbb{I}\left[\left\|\mathbf{X}^1 - \mathbf{X}^2\right\|_2 < \tau_{\text{goal}} \ \wedge \ \max_{0<k<V} \mathbf{P}_k > \tau_{\text{pen}}\right]. \quad (9)$$

with $\mathbf{P}_k$ the SDF for marker k.

Per-agent rewards. To discourage unrealistic foot motion, we define a foot-skating penalty using a velocity threshold τ_{skate}. Let F denote the set of foot markers. For a marker $\mathbf{x} \in F$, we denote its height by $\mathbf{x}_z \in \mathbb{R}$ and its velocity by $\mathbf{v} \in \mathbb{R}$. The skating reward is defined as:

$$r_{\text{skate}} = \exp\left(-\max\left(0, \min_{\mathbf{v}\in F} \mathbf{v} - \tau_{\text{skate}}\right)\right). \tag{10}$$

We further penalize floor contact violations, including both penetration and floating, based on a height threshold τ_{floor}:

$$r_{\text{floor}} = \exp\left(-\max\left(0, \min_{\mathbf{x}\in F} \mathbf{x}_z - \tau_{\text{floor}}\right)\right). \tag{11}$$

To encourage agents to stand upright and avoid bending to reach a target, we define a upright reward based on their spine orientation. Let $\mathbf{e}_z \in \mathbb{R}^3$ be the world up-axis. Let $\mathbf{j}_{\text{pelvis}}, \mathbf{j}_{\text{head}} \in \mathbb{R}^3$ denote the pelvis and head joints of the agent. The upright reward is defined as

$$\mathbf{r}_{\text{up}} = \frac{\mathbf{j}_{\text{head}} - \mathbf{j}_{\text{pelvis}}}{\|\mathbf{j}_{\text{head}} - \mathbf{j}_{\text{pelvis}}\|_2} \cdot \mathbf{e}_z \tag{12}$$

We detail all values of each reward parameter in Table 5.

Parameter	Symbol	Value
Foot skating reward coefficient	λ_{skate}	0.20
Floor reward coefficient	λ_{floor}	0.20
Facing reward coefficient	λ_{face}	0.10
Upright reward coefficient	λ_{up}	0.50
Distance reward coefficient	λ_{dist}	6.00
Penetration reward coefficient	λ_{pen}	0.30
Goal reward coefficient	λ_{goal}	3.00
Foot skating velocity threshold	τ_{skate}	0.075
Floor height threshold	τ_{floor}	0.050
Penetration threshold	δ	-0.026
Goal distance threshold	τ	0.100
Penetration sensitivity	α	40

Table 5: **Reward parameters**.

Differentiable formulation. The geometric contact and penetration costs support differentiation through the motion decoder and mesh construction. In practice, we replace $\max$ and $\min$ operators with smooth softmax variants, and compute signed distances using *Kaolin* (Jatavallabhula et al., 2019). The binary success indicator is distinct from these smooth costs.

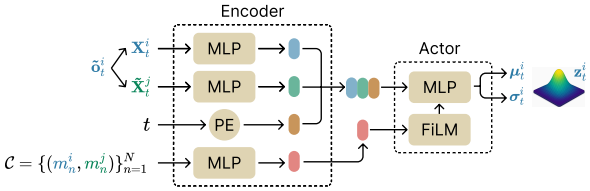
B.2 SDF CALCULATION

We compute SDFs between the two meshes using *kaolin* (Jatavallabhula et al., 2019). Since SMPL meshes contain 6,890 vertices (Loper et al., 2015) and SDF computation is required at every timestep for both agents, this operation is computationally expensive. To reduce overhead, we compute SDFs on a decimated SMPL mesh provided by MeshTransformer (Lin et al., 2021), reducing the vertex count to 431 while maintaining a good trade-off between efficiency and accuracy.

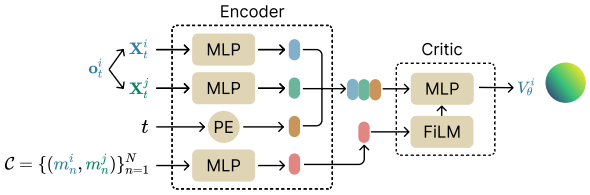
C DETAILED EXPERIMENTS

C.1 DETAILED ARCHITECTURE.

Parameter sharing and asymmetric critic. A single network is shared by both agents and all tasks following MAPPO (Yu et al., 2022); agents differ only through their own observations and



(a) Actor architecture.



(b) Critic architecture.

Parameter	Value
PPO	
Learning rate	1×10^{-4}
Batch size	2048
Buffer size	8208
Step per collect	8208
Repeat per collect	10
Discount factor	0.99
GAE coefficient	0.95
PPO clipping range	0.20
Value function coefficient	1.0
Entropy coefficient	0.0
Max gradient norm	10
KL early-stopping threshold	0.004
Environment steps	16.4M
Behavior cloning	
Demonstrations	9×500
Held-out episodes	100
Epochs	30
Batch size	128

Table 6: **Training hyperparameters.**

goals (their own target markers and those of their partner). Execution remains decentralized, as each agent acts from its own egocentric observation. The critic, used only during training, has its own encoder that receives the partner's full marker set instead of the visible subset (Pinto et al., 2018).

Actor-critic network. Fig. 7a and Fig. 7b show the actor and critic. They have separate observation encoders: each encodes the agent's own markers and partner markers using MLPs, together with a positional encoding of timestep t. The actor receives visible partner markers, while the critic receives the full partner marker set during training (Pinto et al., 2018). Each network's parameters are shared across agents and tasks Yu et al. (2022). The resulting features and contact-goal code feed the corresponding prediction head.

The actor predicts the mean and standard deviation $(\boldsymbol{\mu}_t^i, \boldsymbol{\sigma}_t^i)$ of a Gaussian distribution, from which a latent action vector $\mathbf{z}_t^i$ is sampled. The sampled latent is then passed through a $\tanh$ to constrain the action range.

Goal encoder. The policy is conditioned on the target contacts $\mathcal{C}$ of the current task, given only as marker identities. Each target pair (m_n^i, m_n^j) is turned into a token by concatenating learned 32-dimensional embeddings of its two marker indices, taken from a single embedding table shared by both sides. Tokens are encoded by a shared MLP, mean-pooled and linearly projected into a 512-dimensional goal code. Tasks with fewer than $N = 4$ pairs are padded by repeating their pairs, which leaves the pooled code and the reward unchanged. The goal code modulates every residual block of the actor and critic heads with FiLM (Perez et al., 2018).

Multi-task training. We train a single model on the nine tasks of Fig. 8. At every episode, a task k is drawn with probability proportional to $(1 - \hat{s}_k) + 0.1$, where $\hat{s}_k$ is an exponential moving average of its success rate (decay 0.98), and the share of any task is capped at 15%. This focuses training on the tasks that still fail, while solved tasks are still sampled often enough to detect regressions. Behavior cloning uses 500 demonstrations per task (4,500 in total), pretraining for 30 epochs; 100 held-out episodes are used to select the pretrained checkpoint.

Causal VAE decoder. We decode predicted latent actions into SMPL parameters using a causal VAE decoder pretrained on the HumanML3D motion dataset (Guo et al., 2022). We build on the 1D convolutional VQ-VAE architecture of MotionGPT (Jiang et al., 2023) and T2M-GPT (Zhang et al., 2023), adapting it in two ways: (i) we replace vector quantization with a continuous VAE to enable a fully differentiable pipeline (notably for the trajectory-optimization pipeline), and (ii) we enforce causality in the decoder to ensure that, during rollout, predictions depend only on past and current

latent actions. This is achieved by using left padding only in temporal convolutions, preventing access to future timesteps. The latent action dimension is 128.

We use the SMPL 199-dimensional feature representation of Petrovich et al. (2024); Courant et al. (2026), which includes the pelvis Z (up) coordinate, pelvis linear velocities, angular velocity around the Z axis, the first 22 SMPL (Loper et al., 2015) joint rotations (in 6D representation (Zhou et al., 2019)), and the corresponding 22 joint positions (excluding the pelvis).

PPO traing hyperparameters. For online policy training, we use PPO (Schulman et al., 2017), we provide training hyperparameters in Tab. 6.

C.2 Detailed environments

Environments. Figure Fig. 8 illustrates the nine interaction tasks considered in this work:

- **Two 1-contact tasks** (Figs.8a, 8a): hand-to-head (H2H) and hand-to-back (H2B). These are relatively simple tasks involving a single target contact; H2H requires a direct approach, while H2B is slightly more challenging as the agent must move around its partner to reach the back.
- **Four 2-contact tasks** (Figs.8c, 8d, 8e, 8f): low, mid, up, and cross. These tasks involve a designated actor and receiver, highlighting the need for coordinated motion.
- **Three 4-contact tasks** (Figs.8g, 8h, 8i): low, up, and cross. In these settings, both agents act symmetrically, making them the most challenging scenarios and requiring tight coordination.

Initialization. We show in Fig. 9 the three sources of randomness used at initialization:

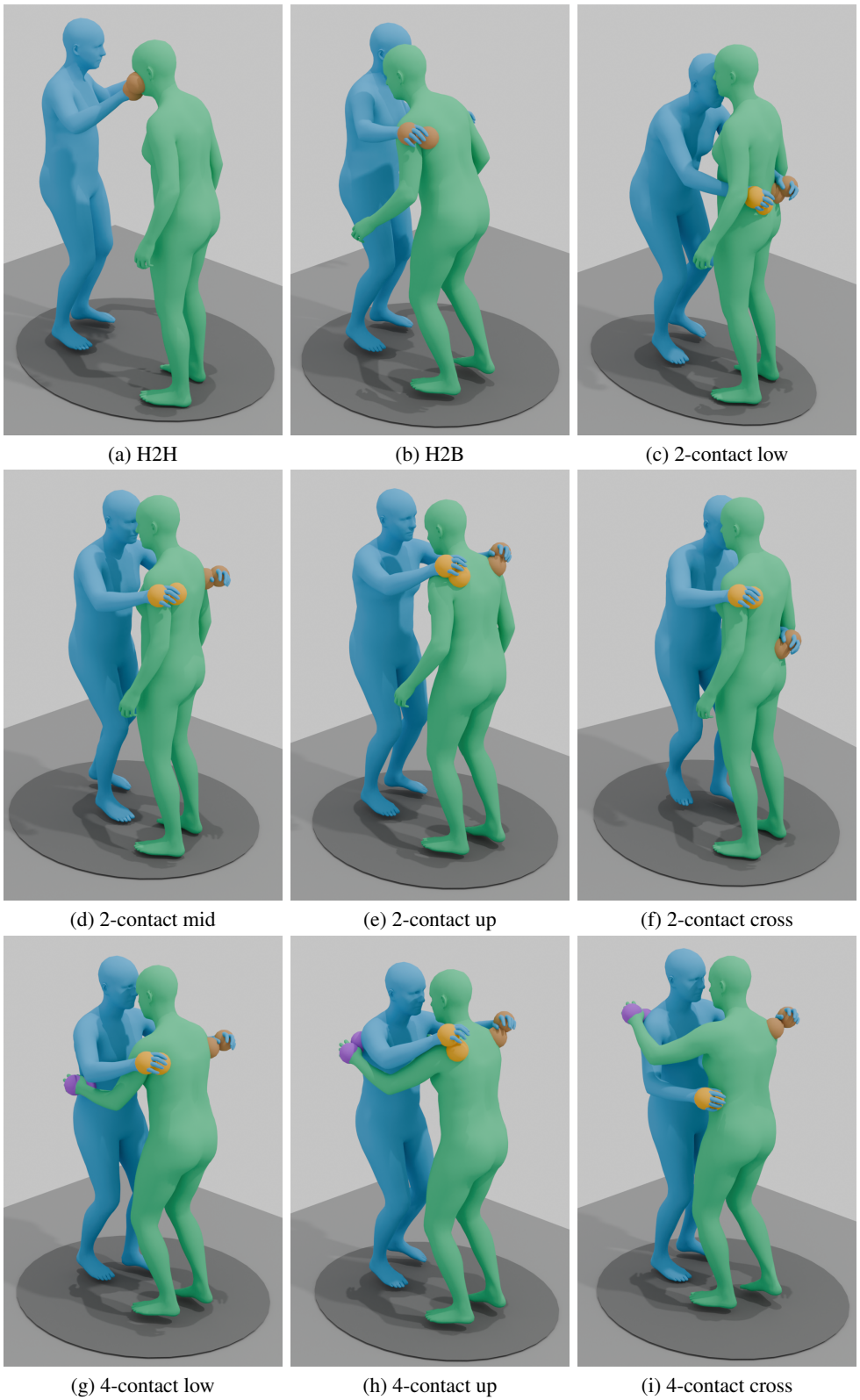
- **Random shapes** (Fig. 9a): SMPL shape parameters (Loper et al., 2015) $\boldsymbol{\beta}$ are sampled from a standard normal distribution and clipped to $[-1, 1]$.
- **Random arm poses** (Fig. 9b): Gaussian noise with scale 0.2 is added to the axis-angle representation of the six arm joints $\boldsymbol{\theta}$, while keeping the rest of the body unchanged.
- **Random orientations** (Fig. 9c): random yaw angles are uniformly sampled in $[-30°, 30°]$ and applied independently to the global orientation of each agent around the vertical axis, preserving the relative body configuration.

C.3 Additional qualitative results

We provide additional qualitative comparisons in Fig. 10 (see also the accompanying video for more results). We show final poses for all baselines across the nine proposed environments. The observations are consistent with those in the main manuscript: *BC* exhibits frequent inter-agent penetration and inaccurate contacts due to limited generalization, as the 500 demonstrations do not sufficiently cover random environment initializations, leading to imprecise motions. *PPO* often fails to establish valid contacts, resulting in penetrations, and produces unnatural poses in more complex tasks even when successful. In contrast, *BC+PPO* leverages high-quality demonstrations to produce more natural interactions. We also observe emergent coordination in back-target tasks, where the hugged agent raises its arms to allow passage underneath, enabling successful contacts without interpenetration.

We further provide videos of both success and failure cases on our gallery webpage[3]. For each task, we display rollouts from 5 different random seeds. Successful episodes exhibit natural human motions, with plausible approach phases and interaction poses. Failures are mainly due to interpenetration between the agents, and occur notably under extreme initial conditions, such as large body-size differences between the two agents (third column of the gallery).

[3] https://close-encounters.github.io/close-encounters-gallery/

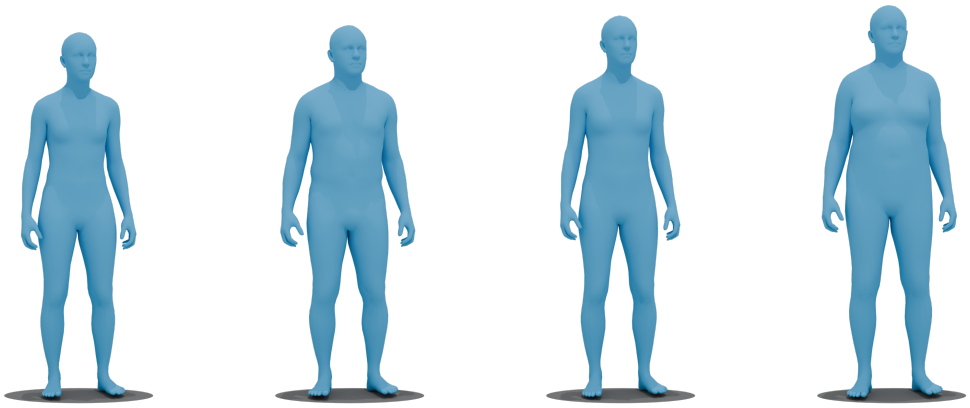


(a) H2H (b) H2B (c) 2-contact low

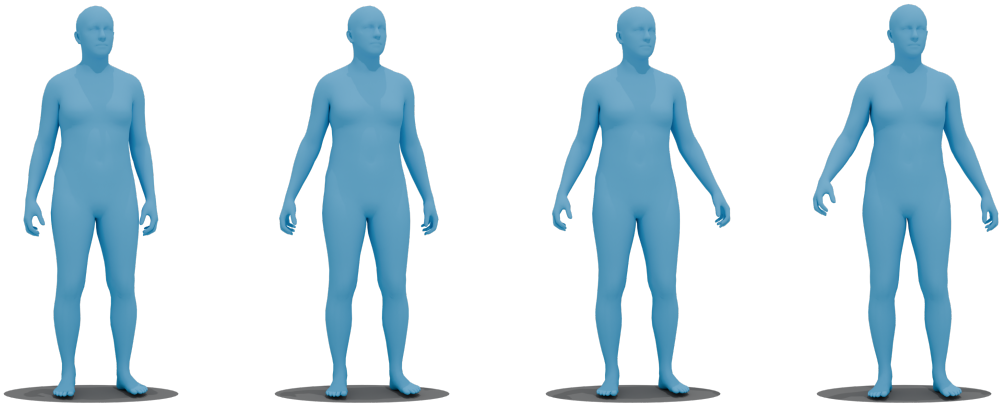
(d) 2-contact mid (e) 2-contact up (f) 2-contact cross

(g) 4-contact low (h) 4-contact up (i) 4-contact cross

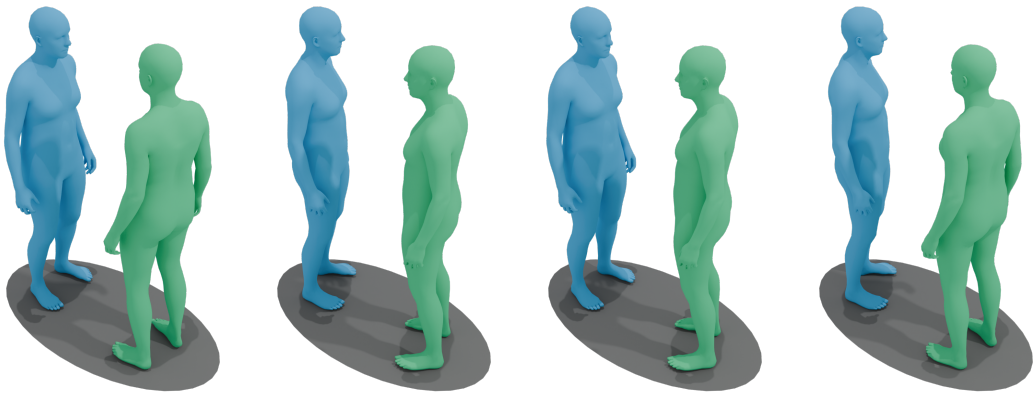
Figure 8: **Overview of evaluation environments.** The nine tasks used to test our framework.



(a) Examples of shape initializations

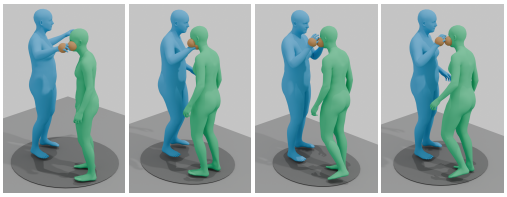


(b) Examples of arm initializations



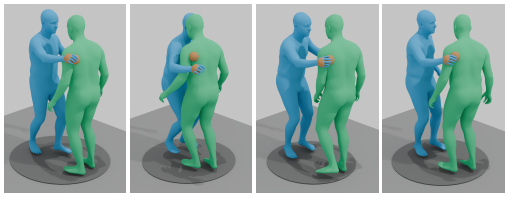
(c) Examples of orientation initializations

Figure 9: **Overview of the various sources of randomness for initialization.**



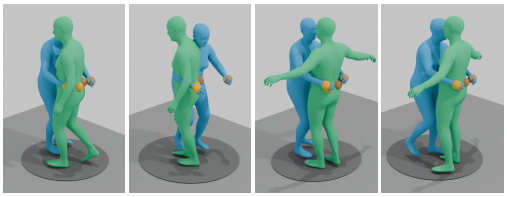
(i) Planner (ii) BC (iii) PPO (iv) BC+PPO

(a) hand-to-head environment



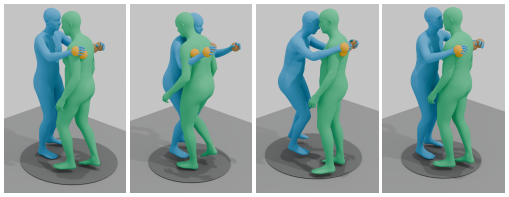
(i) Planner (ii) BC (iii) PPO (iv) BC+PPO

(b) hand-to-back environment



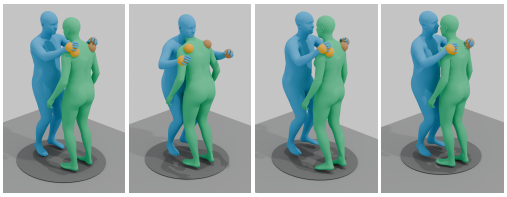
(i) Planner (ii) BC (iii) PPO (iv) BC+PPO

(c) 2-contact low environment



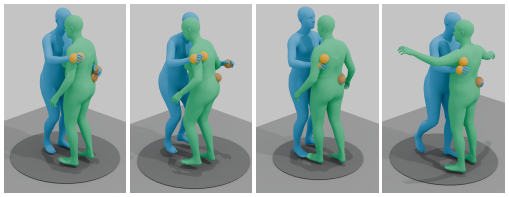
(i) Planner (ii) BC (iii) PPO (iv) BC+PPO

(d) 2-contact mid environment



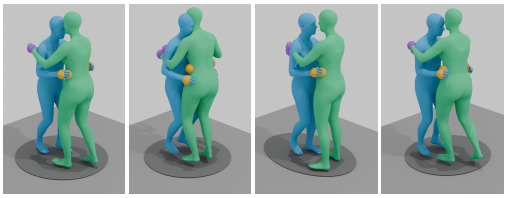
(i) Planner (ii) BC (iii) PPO (iv) BC+PPO

(e) 2-contact up environment



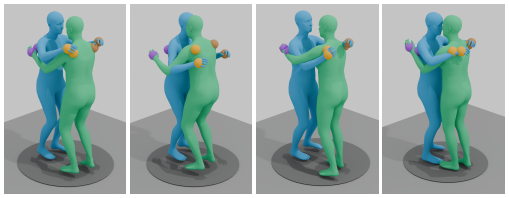
(i) Planner (ii) BC (iii) PPO (iv) BC+PPO

(f) 2-contact cross environment



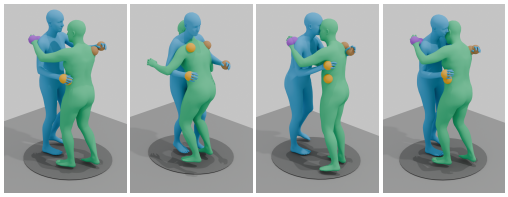
(i) Planner (ii) BC (iii) PPO (iv) BC+PPO

(g) 4-contact low environment



(i) Planner (ii) BC (iii) PPO (iv) BC+PPO

(h) 4-contact up environment



(i) Planner (ii) BC (iii) PPO (iv) BC+PPO

(i) 4-contact cross environment

Figure 10: **Comprehensive qualitative comparison of baselines.** We compare the final poses for the four baselines (Planner, BC, PPO, and BC+PPO) across all nine environments.