InfCoiL: Coordinated Planner-Controller Learning for Closed-Loop Physics-Based Human-Object Interaction

Yude Zou[1,2,*] **Yifei Yao**[2,*] **Yiwei Hao**[1,*] **Hanqing Wang**[2] **Xing Gao**[2,†]
[1]Shanghai Jiao Tong University [2]Shanghai Artificial Intelligence Laboratory

Abstract

Text- and goal-conditioned human-object interaction (HOI) provides an intuitive interface for specifying high-level intents in physics-based character animation. Mapping such intents to physically plausible, contact-rich behaviors naturally calls for a generative planner and a physics-based controller, yet existing planner-controller systems either execute plans open-loop or close the loop while keeping the planner fixed, so execution feedback never reshapes what is planned. We present InfCoiL, a closed-loop framework for physics-based HOI from coordinated planner-controller learning. InfCoiL couples a morphology-aware flow-matching planner with a unified multi-morphology controller through a shared morphology interface in an autoregressive plan-and-execute loop, and alternately fine-tunes both components with reinforcement learning from the same physical rollouts. To optimize the flow-matching planner online, we introduce Semantic Flow Policy Optimization (SFPO), which stochasticizes selected denoising transitions with channel-wise semantic covariance over heterogeneous HOI features and assigns group-relative, sequence-level rewards from complete closed-loop executions. On held-out OMOMO sequences, InfCoiL outperforms open-loop hierarchical baselines in goal-reaching task success and physical tracking accuracy. Project page: `https://infcoil.github.io/`.

1 Introduction

Text- and goal-conditioned physics-based human-object interaction (HOI) is an important step toward controllable character animation and embodied agents. Language specifies high-level intents, while spatial goals ground object-centric interactions geometrically, together forming an intuitive interface for specifying diverse, physically executable interaction. Realizing this interface requires bridging high-level semantic intent to low-level physical control for executable contact-rich interactions.

Existing approaches address different parts of this hierarchy. Kinematic generative models, such as diffusion-based interaction generators [1, 2, 3], learn diverse motion manifolds from large-scale mocap data and support flexible, text-conditioned synthesis. Lacking physical constraints, though, they often suffer from physical implausibility such as human-object penetration and foot skating. Physics-based imitation policies [4, 5] improve dynamic plausibility by tracking reference motions in simulation, yet depend on prescribed references and cannot directly infer fine-grained physical behaviors from high-level semantic conditions. Hierarchical planner-controller systems combine these strengths by using a generative planner to produce references for a physical controller. However, existing hierarchies still lack closed-loop planner-controller co-adaptation: SimGenHOI [6] aligns its generator via supervised fine-tuning on simulator-executed trajectories but still executes generated plans open-loop, whereas CLoSD [7] replans online from simulated states for humanoid motion but adapts only the tracking controller while keeping the diffusion planner fixed.

A natural next step is to close this loop, allowing the planner to recurrently condition on the physical states reached by the controller and revise future interaction plans accordingly. In HOI, however, such

*Equal contribution. †Corresponding author: gaoxing@pjlab.org.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

coordination is not simply achieved by feeding simulator states back to a generative planner. Mocap-trained planners generate morphology-dependent interaction references, whereas controllers are often specialized to specific simulated embodiments, making physical feedback difficult to interpret across morphologies. Meanwhile, planner outputs may fall outside the controller's executable distribution, and controller rollouts may visit physical states unseen during the planner's supervised training. These gaps are further amplified by contact-rich object dynamics, where small errors in contact timing, position, or object pose can quickly lead to cascading failures. Together, these issues reveal *a fundamental coordination problem between generative interaction planning and physical control.*

To address this problem, we propose **InfCoiL**, a closed-loop framework for producing physics-based human-object interactions from coordinated planner-controller learning, as shown in Fig. 1. InfCoiL closes three coupled loops between a generative planner and a physics-based controller: an interface loop that maps realized human-object-contact states back into the planner's conditioning across morphologies, an execution loop that replans from these states, and a learning loop that adapts both modules from the same physical rollouts. The first two loops are realized by *an interface-compatible hierarchical architecture*. A morphology-aware flow-matching planner trained on diverse mocap data generates short-horizon interaction references from textual instructions and spatial goals, conditioned on the target morphology and the realized execution history. A unified multi-morphology controller, rather than separate embodiment-specific policies, executes these references across body shapes.

To narrow the gap between planner-generated references and what the controller can execute, the learning loop alternately fine-tunes the controller to track these references and the planner to generate references that lead to better execution. Online planner fine-tuning, however, requires transition likelihoods that deterministic flow sampling lacks, and exploration suited to HOI tokens, whose body, object, and contact channels differ in physical sensitivity and error tolerance. We therefore propose Semantic Flow Policy Optimization (SFPO), which turns a fixed subset of denoising steps into Gaussian policy steps with semantic, channel-wise covariance. This yields tractable transition log-probabilities and channel-aware exploration while preserving efficient flow sampling. Together, SFPO and PPO controller fine-tuning align the planner's generative distribution with the controller's physical execution capability. Our contributions are summarized as follows:

- We formulate closed-loop physics-based HOI as a planner-controller coordination problem and propose **InfCoiL**, which closes interface, execution, and learning loops between a morphology-aware flow-matching planner and a unified multi-morphology controller.
- We propose SFPO, an online RL method for flow-matching HOI planners that stochasticizes selected denoising transitions with semantic channel-wise covariance and optimizes group-relative, sequence-level outcomes of complete closed-loop executions, and embed it in an alternating planner-controller learning scheme.
- Under a task-oriented evaluation on OMOMO [8] sequences held out from training, we show that InfCoiL improves task success over open-loop and SFT-aligned cascades, and provide component-wise ablations showing that replanning, controller learning, and planner learning address different failure sources, with planner learning most effective after the controller enlarges the executable region.

2 Methodology

We consider text- and goal-conditioned physics-based human-object interaction (HOI), where a simulated humanoid interacts with an object according to semantic and spatial task specifications. At simulation step t, the physical state s_t consists of humanoid, object, and contact states. The task context is defined as

$$c_t = (\ell, \xi_t^p, \xi^o, \beta, o, h_t), \tag{1}$$

where ℓ is the textual instruction, ξ_t^p denotes the local pelvis sub-goal for the current planning window, ξ^o specifies the final object goal, β represents the humanoid morphology using SMPL-X shape parameters [9], o encodes object geometry with BPS features [10], and h_t summarizes recent physical execution history. Given c_t and s_t, the system outputs low-level actions a_t to realize the specified interaction while satisfying spatial goals and physical constraints.

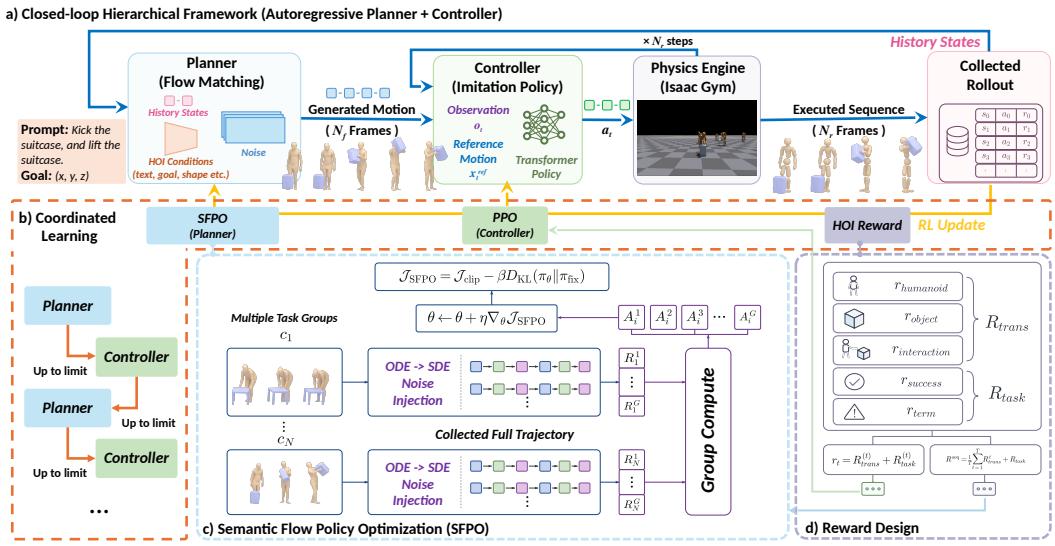

Figure 1: Overview of InfCoiL. (a) Closed-loop framework: the flow-matching planner generates an N_f-frame reference window, the controller executes its first N_r frames in Isaac Gym, and the realized human, object, and contact states are fed back for receding-horizon replanning. (b) Coordinated learning: controller (PPO) and planner (SFPO) training alternate to narrow the gap between planner-generated references and what the controller can execute. (c) SFPO: for each task condition, a group of plans is sampled with semantic noise injected during denoising and executed in closed loop. Their sequence rewards are normalized within the group into relative advantages that update the planner. (d) Reward design: the controller is trained on stepwise transition and task rewards, and each plan is scored by the time-averaged sequence reward. The interaction term is used only in open-loop controller training.

InfCoiL decomposes physics-based HOI into a high-level generative planner P_θ and a low-level physical controller π_ϕ, coupled through a closed-loop plan-and-execute interface, as detailed in Sec. 2.1. The planner predicts an N_f-frame reference window $\hat{x}_{t:t+N_f-1}$, while the controller executes the first N_r simulation steps before replanning, with $N_r \leq N_f$. Throughout the method, t denotes simulation time, t' denotes an execution step within a replanning segment, $\tau \in [0, 1]$ is reserved for flow-matching time, and K denotes the number of flow denoising steps. To further align planning and control, InfCoiL jointly optimizes both components via coordinated reinforcement learning, mitigating the mismatch between generated references and executable behaviors (Sec. 2.2).

2.1 Closed-loop Planner-Controller Framework for HOI

2.1.1 Closed-loop versus Open-loop Design

InfCoiL closes the planner-controller loop at the reference-trajectory level. In a conventional open-loop cascade, a generative model produces a complete kinematic interaction sequence, which is then tracked by a physics-based controller throughout the episode. This assumes that the simulated humanoid, object, and contacts remain close to the planned trajectory. However, contact-rich HOI easily violates this assumption: small errors in contact timing, object pose, root progress, or tracking accuracy accumulate over time and make the remaining reference physically infeasible.

InfCoiL therefore adopts a receding-horizon replanning strategy, where the planner iteratively generates short-horizon reference windows and refines future plans using the controller's realized physical states. Every N_r simulation steps, the planner P_θ predicts $\hat{x}_{t:t+N_f-1} = P_\theta(c_t)$, where c_t is the task context defined in Eq. (1). The controller π_ϕ executes the next segment by tracking these references, $a_{t'} = \pi_\phi(s_{t'}, \hat{x}_{t'})$ for $t' \in [t, t + N_r)$, after which the resulting body-object-contact states are fed back to update c_{t+N_r}. This reference-level closed loop preserves the diversity and goal controllability of kinematic generation while enabling future plans to adapt to physical execution. Additional implementation details are provided in Appendix C.

2.1.2 Morphology-aware Autoregressive Planner

The high-level planner synthesizes short-horizon HOI references using a morphology-aware flow-matching model [11] trained on multi-subject, multi-object interaction data [8].

Given task conditions and recent execution history, it recurrently predicts a short-horizon kinematic reference window for the controller to track. Each frame is represented as a structured HOI token containing body joints, hand joints, global body rotations, object translation and rotation, and hand contact labels. The planner is conditioned on text embeddings, local pelvis and object goals, object BPS features, temporal progress variables, fixed prefix frames, and a 16-dimensional SMPL-X shape code. The morphology condition allows the planner to adapt interaction references to different body shapes, since the same object-centric task may require different reaching poses, contact locations, and root motions across humanoid embodiments.

During training, we interpolate between a clean reference window x_0 and Gaussian noise ϵ,

$$x_\tau = \tau\epsilon + (1-\tau)x_0, \qquad \tau \in (0,1], \tag{2}$$

and train the planner to predict the clean state $\hat{x}_0 = f_\theta(x_\tau, \tau, c)$, where c denotes the full conditioning context, including text, goals, object geometry, morphology, progress, and prefix states. The supervised planning objective is

$$\mathcal{L}_{\text{plan}} = \|f_\theta(x_\tau, \tau, c) - x_0\|_2^2 + \lambda_{\text{obj}}\mathcal{L}_{\text{obj}} + \lambda_{\text{fk}}\mathcal{L}_{\text{fk}}, \tag{3}$$

where $\mathcal{L}_{\text{obj}}$ supervises object keypoints and $\mathcal{L}_{\text{fk}}$ supervises forward-kinematic hand and foot positions, and λ_* denotes corresponding hyperparameter. These auxiliary terms encourage the generated references to preserve object motion consistency and accurate end-effector geometry, which are critical for contact-rich HOI.

At inference time, the planner samples a reference window by integrating the induced velocity

$$v_\theta(x_\tau, \tau, c) = \frac{x_\tau - f_\theta(x_\tau, \tau, c)}{\tau} \tag{4}$$

from noise to data, while clamping the prefix frames from recent physical execution. Since x_τ is linear in x_0 and ϵ, Eq. (4) is the velocity induced by the clean-state prediction, so the trained network is exactly the one being integrated. Predicting x_0 rather than $v = \epsilon - x_0$ regresses the structured HOI state instead of a noise-mixed target and avoids the $1/\tau^2$ reweighting of a velocity-space loss [12]. The clean-state target directly supervises the positions, rotations, object states, and contact labels tracked by the controller, and Appendix B.6 shows that it reduces foot sliding and velocity error of the generated references compared with velocity prediction.

2.1.3 Multi-morphology Controller

The low-level controller provides a unified executable interface for morphology-conditioned planner references. This interface is essential for closed-loop HOI: since the planner generates references for the target humanoid morphology, the controller must execute them directly rather than retarget all motions to a canonical body. Unlike prior imitation frameworks such as InterMimic [4], which consolidate embodiment-specific policies into a shared student with a unified output representation, InfCoiL learns a morphology-conditioned controller that adapts its tracking behavior to different SMPL body shapes while remaining compatible with planner-generated references.

We initialize from morphology-specific InterMimic-style teachers trained for individual subjects and distill them into a single student controller through the distillation-to-RL curriculum of InterMimic [4]. Each environment is assigned a subject-object pair, and the corresponding teacher labels every visited state with its action mean. Rollouts are initially driven by the teacher and gradually handed over to the student by mixing teacher and student actions with a decaying probability, so that expert labels are increasingly collected on student-induced states. After critic warm-up, PPO actor updates are enabled when a critic-reliability criterion is satisfied, with the actor weight gradually increased and the imitation weight reduced, allowing the student to inherit subject-specific tracking skills while adapting to its own induced state distribution.

The student controller is conditioned on the same 16-dimensional SMPL-X shape code β used by the planner, together with a reference-tracking context $\hat{x}$ over four lookahead steps. These inputs encode both body morphology and near-future interaction targets, enabling morphology-adaptive

tracking under contact-rich dynamics. A compact transformer policy embeds the inputs and outputs a Gaussian action mean in the 153-dimensional action space. This yields a shared yet morphology-aware controller that supports closed-loop execution of planner-generated references across diverse humanoid shapes and object interactions.

Extending reference state initialization [13] and diverse resets [14], we further introduce contact-aware state initialization (CSI) to focus controller pretraining on critical HOI phases. For each demonstration, reset weights are computed from hand-contact label transitions, so that rollouts are more frequently initialized near contact onset and release. This exposes the controller to diverse and challenging body-object configurations where tracking errors are most consequential, reducing the need to discover rare contact states through long rollouts and improving contact handling under planner-generated references. Appendix B.1 shows that removing CSI lowers SR by up to 25 points.

2.2 Coordinated Planner-Controller Learning

The closed-loop framework in Sec. 2.1 enables physical execution feedback for future planning, but does not by itself resolve the planner-controller distribution mismatch, *i.e.*, generated references may be hard to execute and controller rollouts may reach states outside the planner's pretraining distribution. To bridge this shift, InfCoiL introduces coordinated planner-controller learning with shared closed-loop rollout feedback. The controller is optimized to execute planner-generated references under simulation dynamics, while the planner is fine-tuned to improve downstream execution outcomes, forming a coupled optimization between generative planning and physical control. We realize this coordination with SFPO for the planner, PPO-based controller optimization, and interaction-centric rewards. Together, these updates align the planner's generative distribution with the controller's executable behavior distribution.

2.2.1 SFPO for Planner Learning

We fine-tune the flow-matching planner with physical execution feedback from the closed-loop controller. This is nontrivial because rectified-flow sampling is typically deterministic, and thus does not directly provide the transition likelihoods required by policy-gradient objectives [15, 16, 17]. Moreover, HOI references are high-dimensional and heterogeneous, containing body motion, object states, and contact labels, making naive stochastic perturbations unstable and inefficient.

To address this, SFPO builds on the stochastic denoising path of Flow-GRPO [16] rather than the flow-matching likelihood surrogate of FPO [17]. Let $z_k = x_{\tau_k}$ $(k = 1, \ldots, K+1)$ be the states of the K-step Euler sampler, from $\tau_1 = 1$ to $\tau_{K+1} = 0$ with $\Delta\tau_k = \tau_{k+1} - \tau_k < 0$. SFPO turns a fixed subset $\mathcal{S}$ of $S \le K$ steps into Gaussian policy steps by Euler–Maruyama discretization of the marginal-preserving flow SDE and keeps the deterministic Euler update at all other steps:

$$\pi_\theta(z_{k+1} \mid z_k, c_t) = \mathcal{N}\big(\mu_{\theta,k},\, (-\Delta\tau_k)\Gamma_{\tau_k}\Gamma_{\tau_k}^\top\big), \qquad \mu_{\theta,k} = z_k + \bar{v}_\theta(z_k, \tau_k, c_t)\,\Delta\tau_k, \qquad k \in \mathcal{S}, \tag{5}$$

where $\bar{v}_\theta$ is a score-free drift computed from the velocity in Eq. (4). We use $K = 10$ and make only the final step stochastic ($\mathcal{S} = \{K\}$), which reduces log-probability recomputation to a single denoising step per plan. To match exploration to heterogeneous HOI tokens, the diffusion coefficient is semantic rather than isotropic, $\Gamma_\tau = \lambda_{\mathrm{n}}\sqrt{\tau/(1-\tau)}\,\mathrm{Diag}(\kappa)$: fixed hand-set scales $\kappa \in \mathbb{R}_+^D$ perturb object translation most and rotations least, and λ_{n} sets their global level.

Planner updates are on-policy. A frozen snapshot θ_{old} generates the plans and stores their transition log-probabilities, which are recomputed under the current θ on the same denoising path. For a task condition c_i, we execute G closed-loop rollouts, and every replanning event t of the g-th rollout forms one training sample with ratio $\rho_\theta^{i,g,t} = \exp(\bar{\ell}_\theta^{i,g,t} - \bar{\ell}_{\theta_{\mathrm{old}}}^{i,g,t})$, where $\bar{\ell}$ denotes the transition log-probability averaged over the generated elements and clipped for numerical stability. Since each plan is scored only by the executed trajectory it leads to, all samples of a rollout share one group-relative advantage [18] computed from the sequence reward R_i^g of rollout g (Eq. (10)):

$$A_i^g = \mathrm{clip}\left(\frac{R_i^g - \bar{R}_i}{\varsigma_i},\, -A_{\max},\, A_{\max}\right), \tag{6}$$

where $\bar{R}_i$ and ς_i are the mean and standard deviation of R_i^g over the valid rollouts of group i, and groups with near-zero reward variance are skipped. The planner then maximizes the clipped surrogate

$\mathcal{J}_{\text{clip}}$ with an optional fixed-reference penalty

$$\begin{aligned}\mathcal{J}_{\text{clip}} &= \mathbb{E}_{i,g,t}\Big[\min\big(\rho_\theta^{i,g,t}A_i^g,\ \text{clip}(\rho_\theta^{i,g,t}, 1-\epsilon_{\text{c}}, 1+\epsilon_{\text{c}})A_i^g\big)\Big],\\ \mathcal{J}_{\text{SFPO}} &= \mathcal{J}_{\text{clip}} - \lambda_{\text{KL}}\,\mathbb{E}_{i,g,t,\,k\in\mathcal{S}}\Big[D_{\text{KL}}\big(\pi_\theta(\cdot \mid z_k, c_t)\,\|\,\pi_{\text{fix}}(\cdot \mid z_k, c_t)\big)\Big].\end{aligned} \tag{7}$$

Here π_{fix} is a frozen pretrained planner, and λ_{KL} weights this optional penalty. We set $\lambda_{\text{KL}} = 0$ in the reported experiments, so that $\mathcal{J}_{\text{SFPO}} = \mathcal{J}_{\text{clip}}$. Appendix E.1 gives the derivation of $\bar{v}_\theta$, the per-channel values of κ, the exact log-probability aggregation, and all hyperparameters.

2.2.2 Human-Object Interaction-Centric Reward

We design an HOI-centric reward to guide both local reference tracking and episode-level task completion, following the interaction-aware reward principles of InterMimic [4]. The reward consists of a stepwise transition reward R_{trans}^t and a task reward R_{task} triggered only at the end of the episode.

The transition reward measures the consistency between the current simulated state and the reference interaction at each simulation step:

$$R_{\text{trans}}^t = w_{\text{h}} r_{\text{humanoid}}^t + w_{\text{o}} r_{\text{object}}^t + w_{\text{int}} r_{\text{interaction}}^t. \tag{8}$$

Here, r_{humanoid}^t and r_{object}^t are exponential tracking rewards based on the mean per-joint position error for the humanoid and the object center position error, respectively. The term $r_{\text{interaction}}^t$ measures the interaction consistency between the human and the object based on hand-object contact labels, contact point distances, and interaction graph errors.

The interaction term $r_{\text{interaction}}^t$ is used only in the open-loop reinforcement learning of the controller and is disabled during closed-loop training. When the planner and the controller adapt to each other, rewarding interaction-graph consistency invites reward hacking toward trajectories with weak or negligible contact, which lowers task success. Closed-loop training therefore relies on the humanoid and object tracking rewards and the terminal task reward.

The task reward is triggered only at the end of the episode:

$$R_{\text{task}} = w_{\text{succ}} R_{\text{success}} + w_{\text{term}} R_{\text{term}}. \tag{9}$$

In this formulation, R_{success} rewards episodes that successfully complete the target human-object interaction. The term R_{term} penalizes episodes that terminate early due to infeasible humanoid motions or excessive object deviation from the reference trajectory, or that reach the timeout without completing the task. The overall reward therefore combines dense local tracking signals with sparse task-level evaluation, allowing the controller and planner to improve physical execution while remaining aligned with the intended HOI objective.

2.2.3 Execution Feedback and Group-Relative Advantage

The controller and the planner share these reward components but use them differently. The controller is a stepwise policy trained with PPO and a learned value function to maximize the expected return of the per-step reward R_{trans}^t, with R_{task} added at the final step. The planner makes low-frequency decisions over long horizons whose quality is revealed only by the complete execution. Each rollout of length T is therefore scored by a sequence reward that averages the transition reward over time, so that the planner cannot gain reward merely by prolonging rollouts:

$$R^{\text{seq}} = \frac{1}{T}\sum_{t=1}^{T} R_{\text{trans}}^t + R_{\text{task}}. \tag{10}$$

As described in Sec. 2.2.1, planner optimization in SFPO is critic-free and uses group-relative advantages. Rollouts in a group share the interaction conditions, such as the start state and task goals, and differ only in the semantic denoising noise. Normalizing the sequence scores R_i^g within each group therefore compares plans under identical conditions and removes reward offsets caused by task difficulty or object scale, which encourages references that the controller can execute. The reward terms, group construction, valid-set masking, and terminal advantage assignment are detailed in Appendix E.3.

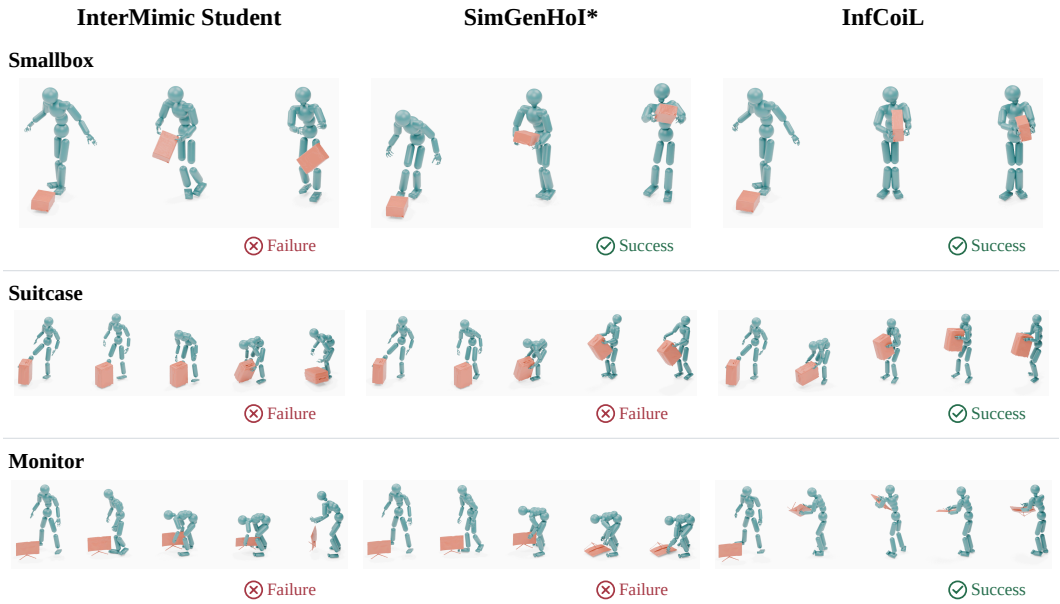

Figure 2: Qualitative comparison on three OMOMO sequences. The instructions, from top to bottom, are 'kick and lift smallbox', 'kick and lift suitcase', and 'lift and rotate monitor'. Green checks and red crosses denote recorded rollout success and failure. In the small-box example, InterMimic student lifts the box but loses its grasp; this rollout is a failure.

2.2.4 Coordinated Learning Paradigm

Optimizing either component in isolation is insufficient. A controller trained alone adapts to fixed planner references but cannot change the motion generation itself, which bounds task success. A planner updated against a frozen controller instead explores references that the controller cannot execute, so it receives little useful feedback. InfCoiL therefore alternates planner and controller updates. After supervised pretraining of the planner and open-loop distillation and RL of the controller, the controller is first fine-tuned with PPO on closed-loop rollouts. Once its tracking quality or success rate plateaus, the controller is frozen and the planner is updated with SFPO until its sequence-level scores plateau, after which training returns to the controller. The controller phase thus resolves the physical executability of current references, and the planner phase moves references toward what the controller can stably track. Algorithm 1 and Appendix E.4 give the full procedure.

3 Experiments

3.1 Evaluation Settings

Metrics and protocol. We evaluate all methods on a morphology-balanced re-split of OMOMO [8], which splits the sequences of every subject with the same train/test ratio. Planner training, controller training, and coordinated fine-tuning use only the training sequences, and all results are reported on the held-out test sequences. All methods share the same Isaac Gym environment, humanoid/object assets, physical parameters, and deterministic controller actions. Each test sequence is evaluated three times and metrics are averaged. SR@d counts an episode as successful when both the pelvis and the object end within d m of their goals. SR denotes SR@0.3 unless stated otherwise, and Table 2 and the appendix additionally report SR@0.2, SR@0.1, and their mean (Avg. SR). We also report pelvis/object goal errors and humanoid/object tracking errors E_h/E_o, all in centimeters. This protocol requires reaching the task goals, unlike the reference-tracking protocol of InterMimic [4], which tracks references on a teacher-supported subset, counts episodes within a 0.5 m deviation as successful, and reports the best sampled rollout. Appendix B.2 explains how this difference affects the InterMimic student results, and Appendix A gives further protocol and metric details.

Baselines and variants. We compare InfCoiL with two open-loop baselines and four internal variants. In all cases, the controller tracks reference trajectories produced by a high-level planner

Table 1: Task-oriented evaluation on the re-split OMOMO test set. Each sequence is evaluated three times and metrics are averaged. C and P denote controller (PPO) and planner (SFPO) fine-tuning phases: InfCoiL (C→P) uses 10k controller iterations followed by 500 planner iterations, and InfCoiL (C→P→C) adds 4k further controller iterations.

Method	SR@0.3↑	E_h ↓	E_o ↓	Pelvis Goal↓	Object Goal↓
InterMimic student (open-loop)	24.62	9.23	16.89	90.39	111.89
SimGenHOI* (open-loop + SFT)	52.14	9.69	18.00	50.12	68.15
InfCoiL open-loop	49.23	9.40	18.08	57.15	75.51
InfCoiL w/o FT P & C	56.09	**7.25**	16.10	46.89	69.83
InfCoiL w/o FT C	56.98	7.75	16.36	47.02	70.60
InfCoiL w/o FT P	73.37	8.44	14.67	35.72	49.45
InfCoiL (C→P)	77.24	8.90	14.76	32.90	45.37
InfCoiL (C→P→C)	**81.19**	8.87	**14.21**	**31.31**	**41.51**

and the key difference is whether these references are generated once in an open-loop manner or repeatedly generated in a closed-loop manner with execution feedback. InterMimic [4] student evaluates the distilled multi-morphology controller under the one-shot planner-reference protocol. SimGenHOI* [6] follows an open-loop cascade with two-stage supervised fine-tuning (SFT) with the same network architecture of planner and controller as ours for fair comparison: the planner is first fine-tuned on trajectories produced by controller tracking, and the controller is then fine-tuned on trajectories generated by the updated planner. InfCoiL open-loop uses our planner and controller with one-shot reference generation and no closed-loop replanning. The closed-loop variant w/o FT P & C instead uses closed-loop online generation with receding-horizon feedback, but freezes both modules. Finally, w/o FT C fine-tunes only the planner, w/o FT P fine-tunes only the controller, and full InfCoiL alternates controller (C) and planner (P) fine-tuning phases. We report InfCoiL (C→P), with one controller phase followed by one planner phase, and InfCoiL (C→P→C), which adds a further controller phase. Unless otherwise stated, InfCoiL refers to the latter.

3.2 Results and Analysis

Table 1 shows that InfCoiL substantially improves task completion over open-loop baselines. Compared with InterMimic student, InfCoiL raises SR from 24.62% to 81.19% and reduces pelvis/object goal errors from 90.39/111.89 cm to 31.31/41.51 cm. Compared with SimGenHOI* (open-loop + SFT), InfCoiL increases SR from 52.14% to 81.19%, while reducing pelvis/object goal errors from 50.12/68.15 cm to 31.31/41.51 cm. These gains show that closed-loop planner-controller learning improves task completion beyond one-shot planner-reference generation. InfCoiL also improves physical tracking quality over open-loop baselines while achieving higher task success. Compared with InterMimic student and SimGenHOI*, InfCoiL reduces E_h from 9.23/9.69 cm to 8.87 cm and E_o from 16.89/18.00 cm to 14.21 cm. This indicates that closed-loop learning improves not only goal reaching, but also the executability of planner-generated references. As shown in Fig. 2, InfCoiL achieves more stable contacts and physically plausible object interactions, and Appendix B.8 analyzes the failure modes. Furthermore, Figs. 3 and 4 demonstrate the generalization to novel object-humanoid morphology combinations.

3.3 Framework Ablation

Component summary. Each component of InfCoiL is isolated by a controlled comparison. Closed-loop replanning alone raises SR from 49.23% to 56.09%, and lowers both goal errors and tracking errors, showing that replanning corrects part of the mismatch between generated references and realized physical states. Controller adaptation (w/o FT P) adds a further 17.28 points, and alternating updates are compared with controller-only training at a matched controller budget below. SFPO is compared with alternative planner RL objectives in Sec. 3.4, online RL is compared with supervised fine-tuning of the planner in Appendix B.4, and removing contact-aware state initialization lowers SR under the reference-tracking protocol of Appendix Table 3.

Effect of coordinated learning. Table 1 shows that fine-tuning only one component exposes the asymmetry between planner and controller adaptation. Planner-only fine-tuning (w/o FT C) provides only a small SR gain over the frozen closed-loop system, from 56.09% to 56.98%, suggesting that

Table 2: Planner RL ablation. All methods start from the same controller-adapted checkpoint (InfCoiL w/o FT P in Table 1) and update the planner with the controller frozen. SFPO with $\lambda_{\mathrm{n}} = 0.1$ corresponds to InfCoiL (C→P). BC is behavior-cloning anchor and Avg. SR averages the three thresholds, where errors are in cm.

Method	SR@0.3↑	SR@0.2↑	SR@0.1↑	Avg. SR↑	$E_h \downarrow$	$E_o \downarrow$	Pelvis↓	Object↓
Start point (w/o FT P)	73.37	67.96	47.54	62.96	8.44	14.67	35.72	49.45
FPO++ [19]	12.83	9.69	4.20	8.91	14.29	22.25	120.03	136.34
FPO++ + BC	73.45	67.64	46.57	62.55	8.52	**14.53**	37.44	51.23
ReinFlow PPO [15]	70.06	50.20	14.21	44.82	12.18	19.07	44.52	62.82
ReinFlow PPO + BC	73.61	67.55	44.31	61.82	**8.42**	14.86	36.78	49.40
Flow-GRPO [16] (isotropic)	22.52	17.59	6.86	15.66	19.02	22.79	95.75	120.75
SFPO ($\lambda_{\mathrm{n}} = 0.05$)	74.66	68.77	47.22	63.55	8.77	**14.53**	34.59	47.33
SFPO ($\lambda_{\mathrm{n}} = 0.2$)	75.14	69.09	44.15	62.79	8.98	16.42	36.89	48.48
SFPO ($\lambda_{\mathrm{n}} = 0.1$)	**77.24**	**72.07**	**48.59**	**65.97**	8.90	14.76	**32.90**	**45.37**

a frozen controller restricts the executable support and thus the useful reward signal for planner learning. Controller-only fine-tuning (w/o FT P) is much stronger, raising SR to 73.37% and reducing object-goal error to 49.45 cm, but it still leaves the planner distribution fixed. Alternating the two phases yields consistent gains: a subsequent SFPO planner phase raises SR to 77.24% (C→P), and a further controller phase reaches 81.19% (C→P→C) while reducing pelvis/object goal errors to 31.31/41.51 cm. In contrast, the matched-budget comparison in Appendix B.3 shows that continuing controller-only training to the same 14k controller iterations lowers SR to 71.99%, indicating that the gain comes from planner-controller co-adaptation rather than additional controller updates. Although the lowest humanoid tracking error E_h is attained by the frozen closed-loop variant, full InfCoiL achieves the best task success, goal accuracy, and object tracking error, suggesting that coordinated learning optimizes executable task completion rather than tracking fidelity alone.

3.4 Planner RL Ablation

We compare SFPO with planner-side reinforcement learning baselines under the same closed-loop setting. All methods start from the same controller-adapted checkpoint, keep the controller frozen, and update the planner with the same number of iterations. The baselines cover critic-based single-step updates, namely FPO++ [19], an improved variant of FPO [17], and ReinFlow PPO [15], each with and without a behavior-cloning (BC) anchor, as well as Flow-GRPO [16] with isotropic noise. Implementation details are provided in Appendix B.5.

Table 2 shows that critic-based updates do not improve on this strong starting point: without a BC anchor, FPO++ and ReinFlow PPO degrade to 12.83% and 70.06% SR@0.3, and with the anchor they at most preserve the initial policy (73.45% and 73.61% vs. 73.37%). Making the flow sampler stochastic with isotropic noise is also insufficient, as Flow-GRPO drops to 22.52%. In contrast, SFPO improves SR@0.3 over the starting point at every tested noise level ($\lambda_{\mathrm{n}} \in \{0.05, 0.1, 0.2\}$, 74.66–77.24%), and the default $\lambda_{\mathrm{n}} = 0.1$ achieves the best success at all thresholds together with the lowest goal errors, while a larger noise level trades strict-threshold success (SR@0.1) for coarse-threshold gains. These results indicate that channel-wise exploration matched to heterogeneous HOI tokens is what makes sequence-level planner RL stable in this setting. Appendix B.4 further compares online RL with supervised fine-tuning of the planner.

4 Related Works

Generative Models for Human-Object Interaction. Kinematic generative models have become a dominant approach for synthesizing human-object interactions (HOI). Diffusion-based methods incorporate various conditions to support object-guided manipulation and text-controllable interaction synthesis [1, 8, 20, 2]. Subsequent work improves conditioning and representation design through semantic-geometric context [21], object relations [22], kinematic-chain structures [23], unseen-object generalization [24], and interaction dynamics [25]. Other studies accelerate or refine the generative process itself, including asynchronous denoising for classifier-free interaction guidance [3] and consistency-based motion generators for fast inference [26, 27, 28]. However, flow-based

HOI generation remains underexplored. Moreover, existing generators are typically trained without physical execution feedback, limiting their ability to adapt plans to the states reached by the controller.

Physics-based Human-Object Interaction. Physics-based HOI methods aim to execute interactions under contact-rich dynamics rather than only synthesize kinematic trajectories. A direct paradigm is to learn task policies with reinforcement learning from manually designed rewards, which can produce executable behaviors for sports, games, and object manipulation but often depends on task-specific engineering and lacks demonstration-level realism [29, 30, 31]. Imitation-based methods reduce this burden by tracking human-object demonstrations in simulation, enabling contact-rich and reusable interaction skills [32, 33, 34]. However, these skills remain bounded by the objects, motions, and interaction modes covered by demonstrations, limiting their ability to synthesize diverse long-horizon behaviors from fine-grained textual and spatial instructions. Recent large-scale methods improve coverage by learning whole-body controllers from broader and imperfect mocap interaction data, including unified controllers for heterogeneous objects and sparse-intention-conditioned control [4, 5]. These methods ground HOI in simulation, but they remain less expressive than text-conditioned generative planners in semantic controllability and motion diversity. Hierarchical systems further combine interaction generators with imitation policies to connect planning and physical tracking [6], but they still generate static references before tracking, leaving the planner essentially open-loop. In contrast, InfCoiL addresses this gap with a closed-loop planner-controller framework, where the planner repeatedly conditions on language, goals, morphology, object states, and recent physical feedback, and coordinated reinforcement learning adapts both modules to each other.

5 Conclusion and Limitation

This work studies how high-level generative interaction planning can be coordinated with low-level physical control for text- and goal-conditioned human-object interaction. A series of validations on the OMOMO benchmark shows that the closed-loop planner-controller framework enables adaptive replanning from physical execution feedback, while coordinated learning further aligns motion-capture-based generation with executable simulated behavior. The ablations also indicate when each component matters: closed-loop replanning corrects recoverable drift from the generated references, controller adaptation enlarges the physically executable region, and planner learning is most effective once this region has expanded, where it raises the final task-success ceiling. Together, these findings suggest that physics-based HOI animation requires not only diverse and controllable generative priors, but also their closed-loop adaptation to the physical controller and contact-rich execution dynamics.

While InfCoiL demonstrates strong performance in simulation, several limitations remain. First, InfCoiL is a physics-based character animation system. All experiments use simulated SMPL-X humanoids, whose shape variation is not equivalent to the kinematics, actuation, and sensing of real humanoid robots, so real-robot deployment would require adaptation to hardware constraints, sensing noise, and real contact dynamics. Second, the current setting assumes structured object states and goal specifications, which offers a controlled testbed for studying planner-controller coordination but leaves perception-driven inputs and more complex scenes to future work. Third, the effective semantic range of language conditioning follows the OMOMO/CHOIS annotations, and we have not evaluated paraphrased or out-of-distribution instructions. Appendix F.2 discusses possible extensions.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 62401367 and in part by Shanghai Artificial Intelligence Laboratory.

References

[1] Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14928–14940, 2023.

[2] Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C Karen Liu. Controllable human-object interaction synthesis. In *European Conference on Computer Vision*, pages 54–72. Springer, 2024.

[3] Ziyin Wang, Sirui Xu, Chuan Guo, Bing Zhou, Jiangshan Gong, Jian Wang, Yu-Xiong Wang, and Liang-Yan Gui. Unleashing guidance without classifiers for human-object interaction animation. *arXiv preprint arXiv:2603.25734*, 2026.

[4] Sirui Xu, Hung Yu Ling, Yu-Xiong Wang, and Liang-Yan Gui. Intermimic: Towards universal whole-body control for physics-based human-object interactions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12266–12277, 2025.

[5] Sirui Xu, Samuel Schulter, Morteza Ziyadi, Xialin He, Xiaohan Fei, Yu-Xiong Wang, and Liangyan Gui. Interprior: Scaling generative control for physics-based human-object interactions. *arXiv preprint arXiv:2602.06035*, 2026.

[6] Yuhang Lin, Yijia Xie, Jiahong Xie, Yuehao Huang, Ruoyu Wang, Jiajun Lv, Yukai Ma, and Xingxing Zuo. Simgenhoi: Physically realistic whole-body humanoid-object interaction via generative modeling and reinforcement learning. *arXiv preprint arXiv:2508.14120*, 2025.

[7] Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit H Bermano, and Michiel van de Panne. Closd: Closing the loop between simulation and diffusion for multi-task character control. *arXiv preprint arXiv:2410.03441*, 2024.

[8] Jiaman Li, Jiajun Wu, and C Karen Liu. Object motion guided human motion synthesis. *ACM Transactions on Graphics (TOG)*, 42(6):1–11, 2023.

[9] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019.

[10] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4332–4341, 2019.

[11] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[12] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

[13] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[14] Patrick Yin, Tyler Westenbroek, Zhengyu Zhang, Joshua Tran, Ignacio Dagnino, Eeshani Shilamkar, Numfor Mbiziwo-Tiapo, Simran Bagaria, Xinlei Liu, Galen Mullins, et al. Emergent dexterity via diverse resets and large-scale reinforcement learning. *arXiv preprint arXiv:2603.15789*, 2026.

[15] Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. Reinflow: Fine-tuning flow matching policy with online reinforcement learning. *arXiv preprint arXiv:2505.22094*, 2025.

[16] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

[17] David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

[18] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[19] Brent Yi, Hongsuk Choi, Himanshu Gaurav Singh, Xiaoyu Huang, Takara E Truong, Carmelo Sferrazza, Yi Ma, Rocky Duan, Pieter Abbeel, Guanya Shi, et al. Flow policy gradients for robot control. *arXiv preprint arXiv:2602.02481*, 2026.

[20] Christian Diller and Angela Dai. Cg-hoi: Contact-guided 3d human-object interaction generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19888–19901, 2024.

[21] Peishan Cong, Ziyi Wang, Yuexin Ma, and Xiangyu Yue. Semgeomo: Dynamic contextual human motion generation with semantic and geometric guidance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17561–17570, 2025.

[22] Mengqing Xue, Yifei Liu, Ling Guo, Shaoli Huang, and Changxing Ding. Guiding human-object interactions with rich geometry and relations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22714–22723, 2025.

[23] Ling-An Zeng, Guohong Huang, Yi-Lin Wei, Shengbo Gu, Yu-Ming Tang, Jingke Meng, and Wei-Shi Zheng. Chainhoi: Joint-based kinematic chain modeling for human-object interaction generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12358–12369, 2025.

[24] Shujia Li, Haiyu Zhang, Xinyuan Chen, Yaohui Wang, and Yutong Ban. Genhoi: Generalizing text-driven 4d human-object interaction synthesis for unseen objects. *arXiv preprint arXiv:2506.15483*, 2025.

[25] Lin Wu, Zhixiang Chen, and Jianglin Lan. Hoi-dyn: Learning interaction dynamics for human-object motion diffusion. *arXiv preprint arXiv:2507.01737*, 2025.

[26] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. In *European Conference on Computer Vision*, pages 390–408. Springer, 2024.

[27] Lei Jiang, Ye Wei, and Hao Ni. Motionpcm: Real-time motion synthesis with phased consistency model. *arXiv preprint arXiv:2501.19083*, 2025.

[28] Yude Zou, Junji Gong, Xing Gao, Zixuan Li, Tianxing Chen, and Guanjie Zheng. Infbagel: Human-object-scene interaction generation with dynamic perception and iterative refinement. *arXiv preprint arXiv:2604.04843*, 2026.

[29] Josh Merel, Saran Tunyasuvunakool, Arun Ahuja, Yuval Tassa, Leonard Hasenclever, Vu Pham, Tom Erez, Greg Wayne, and Nicolas Heess. Catch & carry: reusable neural controllers for vision-guided whole-body tasks. *ACM Transactions on Graphics (TOG)*, 39(4):39–1, 2020.

[30] Libin Liu and Jessica Hodgins. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *Acm transactions on graphics (tog)*, 37(4):1–14, 2018.

[31] Xinyu Xu, Yizheng Zhang, Yong-Lu Li, Lei Han, and Cewu Lu. Humanvla: Towards vision-language directed object rearrangement by physical humanoid. *Advances in Neural Information Processing Systems*, 37:18633–18659, 2024.

[32] Yinhuai Wang, Jing Lin, Ailing Zeng, Zhengyi Luo, Jian Zhang, and Lei Zhang. Physhoi: Physics-based imitation of dynamic human-object interaction. *arXiv preprint arXiv:2312.04393*, 2023.

[33] Yinhuai Wang, Qihan Zhao, Runyi Yu, Hok Wai Tsui, Ailing Zeng, Jing Lin, Zhengyi Luo, Jiwen Yu, Xiu Li, Qifeng Chen, et al. Skillmimic: Learning basketball interaction skills from demonstrations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17540–17549, 2025.

[34] Runyi Yu, Yinhuai Wang, Qihan Zhao, Hok Wai Tsui, Jingbo Wang, Ping Tan, and Qifeng Chen. Skillmimic-v2: Learning robust and generalizable interaction skills from sparse and noisy demonstrations. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[36] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023.

[37] Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions On Graphics (TOG)*, 43(6):1–21, 2024.

[38] Yifei Yao, Chengyuan Luo, Jiaheng Du, Wentao He, and Jun-Guo Lu. Gbc: Generalized behavior-cloning framework for whole-body humanoid imitation. *arXiv preprint arXiv:2508.09960*, 2025.

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

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

[41] Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582*, 2023.

[42] Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. Controlvae: Model-based learning of generative controllers for physics-based characters. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022.

[43] Heyuan Yao, Zhenhua Song, Yuyang Zhou, Tenglong Ao, Baoquan Chen, and Libin Liu. Moconvq: Unified physics-based motion control via scalable discrete representations. *ACM Transactions on Graphics (TOG)*, 43(4):1–21, 2024.

[44] Yan Wu, Korrawe Karunratanakul, Zhengyi Luo, and Siyu Tang. Uniphys: Unified planner and controller with diffusion for flexible physics-based character control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13214–13224, 2025.

[45] Xiaoyu Huang, Takara Truong, Yunbo Zhang, Fangzhou Yu, Jean Pierre Sleiman, Jessica Hodgins, Koushil Sreenath, and Farbod Farshidian. Diffuse-cloc: Guided diffusion for physics-based character look-ahead control. *ACM Transactions on Graphics (TOG)*, 44(4):1–12, 2025.

[46] Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. Superpadl: Scaling language-directed physics-based control with progressive supervised distillation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[47] Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–9, 2023.

[48] Takara Everest Truong, Michael Piseno, Zhaoming Xie, and Karen Liu. Pdp: Physics-based character animation via diffusion policy. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–10, 2024.

[49] Nicklas Hansen, Jyothir SV, Vlad Sobal, Yann LeCun, Xiaolong Wang, and Hao Su. Hierarchical world models as visual whole-body humanoid controllers. *arXiv preprint arXiv:2405.18418*, 2024.

[50] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13756–13766, 2023.

[51] Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. *arXiv preprint arXiv:2302.01877*, 2023.

[52] Zixuan Wu, Sean Ye, Manisha Natarajan, and Matthew C Gombolay. Diffusion-reinforcement learning hierarchical motion planning in adversarial multi-agent games. *arXiv e-prints*, pages arXiv–2403, 2024.

[53] Liangliang Chen, Yutian Lei, Shiyu Jin, Ying Zhang, and Liangjun Zhang. Rlingua: Improving reinforcement learning sample efficiency in robotic manipulations with large language models. *IEEE Robotics and Automation Letters*, 9(7):6075–6082, 2024.

[54] Yifei Yao, Wentao He, Chenyu Gu, Jiaheng Du, Fuwei Tan, Zhen Zhu, and Jun-Guo Lu. Anybipe: An automated end-to-end framework for training and deploying bipedal robots powered by large language models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 14790–14797. IEEE, 2025.

[55] Allen Ren, Justin Lidard, Lars Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, volume 2025, pages 77288–77329, 2025.

[56] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *European conference on computer vision*, pages 581–600. Springer, 2020.

[57] Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15935–15946, 2022.

[58] Jiaxin Lu, Chun-Hao Paul Huang, Uttaran Bhattacharya, Qixing Huang, and Yi Zhou. Humoto: A 4d dataset of mocap human object interactions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10886–10897, 2025.

[59] Jiawei Gao, Ziqin Wang, Zeqi Xiao, Jingbo Wang, Tai Wang, Jinkun Cao, Xiaolin Hu, Si Liu, Jifeng Dai, and Jiangmiao Pang. Coohoi: Learning cooperative human-object interaction with manipulated object dynamics. *Advances in Neural Information Processing Systems*, 37:79741–79763, 2024.

[60] Liang Pan, Zeshi Yang, Zhiyang Dou, Wenjia Wang, Buzhen Huang, Bo Dai, Taku Komura, and Jingbo Wang. Tokenhsi: Unified synthesis of physical human-scene interactions through task tokenization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5379–5391, 2025.

[61] Boyu Mi, Mengchen Ma, Yifei Yao, Xing Gao, Junting Chen, Yangzi Li, Zihou Zhu, Guohao Li, Zhenfei Yin, Tai Wang, et al. Exploratory, communicative, and deployable: Vision-driven embodied agents for open-world mobile manipulation. In *European Conference on Computer Vision*, pages 627–645. Springer, 2026.

[62] Jisong Cai, Yao Mu, Ganlin Yang, Zhe Cao, Zhangzheng Tu, Xing Gao, Kailin Li, Xinyu Zhan, Lixin Yang, Yangkun Zhu, et al. Internw0: A foundational physical world model for efficient real-world interactions. *arXiv preprint arXiv:2609.27656*, 2026.

[63] Xingyu Miao, Zizun Li, Baole Fang, Kaiwen Song, Tenghui Wang, Hanxue Zhang, Yating Wang, Xudong Li, Yuping He, Xueyuan Wei, et al. Internw0-Δ: A world action model bridging predictive dynamics and actions with 20k+ hours of open data. *arXiv preprint arXiv:2609.31394*, 2026.

[64] Yude Zou, Yifei Yao, Guanbo Jiao, Jianming Ma, Guanjie Zheng, and Xing Gao. PICA-WAM: Visual perturbation invariance via cross-appearance self-alignment in world action models. *arXiv preprint arXiv:2610.26813*, 2026.

[65] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). In *European Conference on Computer Vision*, pages 264–273. Springer, 2024.

[66] Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27649–27660. IEEE, 2025.

[67] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[68] Ning Gao, Jinliang Zheng, Xing Gao, Haoxiang Ma, Hanqing Wang, Yukai Wang, Jiantong Chen, Zanxin Chen, Shujie Zhang, Mingda Jia, et al. Ebench: Elemental diagnosis of generalist mobile manipulation policies. *arXiv preprint arXiv:2606.18239*, 2026.

[69] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International conference on robotics and automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[70] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as I can, not as I say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[71] Zhuo Lin, Sirui Xu, Liuyu Bian, Yu-Xiong Wang, and Liang-Yan Gui. InterEvolve: Test-time evolution of reward programs for humanoid loco-manipulation. *arXiv preprint arXiv:2610.02196*, 2026.

[72] Yifei Yao, Yude Zou, Guanbo Jiao, Xing Gao, Ning Gao, Yukai Wang, Jiantong Chen, Zihou Zhu, Xinyu Li, Ningyu Yan, et al. Profiling a frontier agent with EBench: From direct control to harness with policy. *arXiv preprint arXiv:2610.35832*, 2026.

[73] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018.

[74] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. A study of BFLOAT16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

[75] Hanqing Liu, Jianjun Cao, Yuanze Li, and Zijian Zhou. Grokking or glitching? How low-precision drives Slingshot loss spikes. In *ICML 2026 Workshop on High-Dimensional Learning Dynamics*, 2026.

[76] Penghui Qi, Zichen Liu, Xiangxin Zhou, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Defeating the training-inference mismatch via FP16. *arXiv preprint arXiv:2510.26788*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions and scope are specifically claimed in the abstract and introduction. Our key contributions are summarized in the last paragraph of Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present theorems. The derivation of the SFPO transition likelihood is given in Section 2.2.1 and Appendix E.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully describe our architecture settings, simulation and training parameters in Section 3 and Appendices A, C, and D, and the provided information is sufficient to reproduce our main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide sufficient material in supplemental material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify our training and test details in Section 3.1 and Appendices A, D, and E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We define the evaluation metrics in Section 3.1. And then we compare our method with other baselines, showing statistical results in Sections 3.2 and 3.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see our experimental details in Section 3 and Appendices C and D. Training InfCoiL takes about one week on four NVIDIA GeForce RTX 4090 GPUs (Appendix D).

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We read the NeurIPS Code of Ethics, and make sure that our research conducted in the paper conform with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impacts in Appendix F.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite works of related assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets are introduced.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: NA

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: NA

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: NA

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Experimental Protocol and Metrics

This section defines the evaluation protocol, baseline variants, and metrics used in the main experiments.

A.1 Evaluation Protocol and Baseline Variants

All main comparisons are evaluated in the same Isaac Gym simulation environment with the same humanoid assets, object assets, and object physical parameters. We re-split OMOMO by partitioning the sequences of each morphology with the same train/test ratio, rather than using the original split that holds out two unseen morphologies for testing. We use a 30 Hz control rate by setting the simulation control-frequency divisor to two, and evaluate each method with deterministic controller actions. In the task-oriented protocol, each test sequence is evaluated three times and metrics are averaged. Planner training, controller training, and coordinated fine-tuning use only the training sequences. A rollout is successful at threshold d (SR@d) if both the human pelvis and the object end within d m of their target end positions. We use $d = 0.3$ as the primary threshold and additionally report $d \in \{0.2, 0.1\}$ where available. In the reported tables, success rate is shown in percent, while humanoid tracking error E_h, object tracking error E_o, pelvis-goal error, and object-goal error are reported in centimeters. Table 3 uses a separate InterMimic-style protocol, where success is termination-based rather than task-goal based.

Baselines and variants. The baselines and variants are described in Section 3.1. InterMimic student uses the same observation normalization, action space, PD control, and evaluation code as our controller. InterMimic student, SimGenHOI*, and InfCoiL open-loop generate the complete reference once before execution, and the controller tracks it until success, failure, or timeout without feeding simulated states back to the planner. The closed-loop variants w/o FT P & C, w/o FT C, and w/o FT P use the same receding-horizon execution as InfCoiL and differ only in which modules are fine-tuned.

InterMimic-style reference-tracking protocol. Table 3 follows the InterMimic evaluation protocol rather than the task-oriented protocol in Table 1: rollouts are uniformly sampled from the dataset under a fixed budget, results are reported by the best sampled rollout, and SR means not triggering termination under the 0.5 m deviation threshold instead of reaching task goals.

A.2 Metrics

Success rate SR@d is the fraction of evaluated episodes in which both the pelvis-goal and object-goal errors are below d m, and SR without a threshold denotes SR@0.3. Avg. SR is the mean of SR@0.3, SR@0.2, and SR@0.1. Table 4 reports all three thresholds for the closed-loop variants. For the open-loop methods, InterMimic student, SimGenHOI*, and InfCoiL open-loop reach 18.56%, 45.12%, and 43.58% SR@0.2 and 9.52%, 25.75%, and 24.13% SR@0.1 (Avg. SR 17.57%, 41.00%, and 38.98%), all below the frozen closed-loop system at every threshold. The pelvis-goal and object-goal errors are the final Euclidean distances to the corresponding target goals, reported in centimeters.

Humanoid tracking error E_h is the mean Euclidean distance between selected simulated body keypoints and their reference keypoints, accumulated over the episode and reported in centimeters. Object tracking error E_o is the mean Euclidean distance between sampled object surface points transformed by the simulated object pose and the same points transformed by the reference object pose, also reported in centimeters. These two metrics measure physical tracking quality rather than task completion alone: a method can reach the final goal but still have poor reference tracking, or track locally while failing to complete the object-centric task.

B Additional Experimental Results

This section reports supplementary quantitative and qualitative results that support the main experimental analysis.

B.1 InterMimic-Style Reference-Tracking Evaluation

Under the InterMimic-style protocol defined above, InfCoiL Teacher improves over InterMimic Teacher on both subsets, raising SR from 48.21%/50.38% to 92.86%/93.56% and reducing E_h/E_o from 9.58/9.04 and 11.18/14.56 cm to 6.03/6.13 and 6.83/9.39 cm. The w/o CSI rows show that contact-aware state initialization is important for robust execution: removing CSI lowers SR by 25.00/7.58 points and worsens object tracking, despite a slightly lower FID on subset-8.

Table 3: InterMimic-style reference-tracking evaluation. Rollouts are uniformly sampled from the dataset under a fixed sampling budget and reported by the best sampled rollout. SR denotes not triggering termination under the 0.5 m deviation threshold, not task-goal success.

Method	SR ↑	E_h ↓	E_o ↓	Rprec ↑			FID ↓
				Top-1	Top-2	Top-3	
subset-2 with ground-truth references							
InterMimic Teacher	48.21	9.58	9.04	12.50	31.25	34.38	10.36
InfCoiL Teacher	**92.86**	**6.03**	**6.13**	**18.75**	**43.75**	**53.12**	**3.29**
InfCoiL Teacher w/o CSI	67.86	7.01	13.34	9.38	28.12	43.75	3.98
subset-8 with ground-truth references							
InterMimic Teacher	50.38	11.18	14.56	19.53	35.16	46.88	5.88
InfCoiL Teacher	**93.56**	**6.83**	**9.39**	**23.83**	**42.19**	**55.08**	1.76
InfCoiL Teacher w/o CSI	85.98	7.97	14.92	21.48	38.67	53.12	**1.46**

B.2 Relation to the InterMimic Evaluation Protocol

The InterMimic student results in Table 1 are lower than those reported by InterMimic [4] because the two protocols measure different objectives, not because of a reproduction failure. We load the officially released student checkpoint with its official environment configuration and change only the references and the metric. InterMimic distills its student from successful teacher rollouts, omitting sequences that no teacher can track, and counts a rollout as successful if it tracks the reference without exceeding a 0.5 m deviation, reporting the best of repeated trials. Its success rate therefore measures reference survival on a teacher-supported subset. Moreover, each morphology-specific InterMimic teacher is trained on the full data of its subject before distillation, so the sequences in its reported OMOMO-Test evaluation are not held out from teacher training. Our task-oriented SR instead requires both the pelvis and the object to reach their final goals on held-out test sequences, with planner-generated references and three averaged trials. Teacher filtering is also inapplicable to closed-loop generation. Future references depend on the states reached during execution, so filtering after execution would remove planning failures and reduce the task to post-hoc reference imitation. In summary, Table 1 measures planner-conditioned task completion of the full system on held-out sequences, whereas the InterMimic-style evaluation in Table 3 isolates controller tracking under ground-truth references.

B.3 Coordinated Learning at Matched Controller Budget

Table 4: Coordinated learning at a matched controller budget under the closed-loop protocol of Table 1. C and P denote controller (PPO) and planner (SFPO) fine-tuning phases, with iteration counts in parentheses. Avg. SR averages the three thresholds, and errors are in cm.

Training path	SR@0.3↑	SR@0.2↑	SR@0.1↑	Avg. SR↑	E_h ↓	E_o ↓	Pelvis↓	Object↓
Frozen closed loop	56.09	48.91	30.59	45.20	**7.25**	16.10	46.89	69.83
C (10k)	73.37	67.96	47.54	62.96	8.44	14.67	35.72	49.45
C (10k) → P (500)	77.24	72.07	48.59	65.97	8.90	14.76	32.90	45.37
C (14k)	71.99	66.83	44.96	61.26	9.19	**14.09**	40.45	51.62
C (10k) → P (500) → C (+4k)	**81.19**	**76.84**	**58.03**	**72.02**	8.87	14.21	**31.31**	**41.51**

Table 4 separates the contribution of planner learning from the number of controller updates. Starting from the frozen closed-loop system, 10k controller iterations raise SR@0.3 to 73.37%, and a subsequent SFPO planner phase adds 3.87 points (77.24%). Continuing controller-only training to 14k iterations does not help and lowers SR@0.3 to 71.99%, whereas inserting the planner phase before the same additional 4k controller iterations reaches 81.19%. At a matched controller budget of 14k iterations, coordinated learning therefore improves SR@0.3 by 9.20 points, Avg. SR by 10.76 points, and SR@0.1 by 13.07 points, and reduces pelvis/object goal errors from 40.45/51.62 cm to 31.31/41.51 cm. Planner adaptation thus raises the ceiling that controller adaptation alone reaches, and the gain of InfCoiL (C→P→C) does not come from additional controller updates.

B.4 Online RL versus Supervised Fine-Tuning

Table 5 compares three planner update rules with the same controller-adapted checkpoint: no planner update, supervised fine-tuning (SFT) on trajectories executed by the controller, following the planner alignment of the SimGenHOI* baseline [6], and online RL with SFPO. For SFT, each training sequence is executed 10 times, and

Table 5: Planner update rules under closed- and open-loop execution. All rows use the controller after 10k closed-loop iterations (C (10k) in Table 4), and the planner is kept fixed, fine-tuned by SFT on executed trajectories as in SimGenHOI*, or fine-tuned online by SFPO. Open-loop rows execute references generated once by the same three planners.

Execution	Planner update	SR@0.3↑	SR@0.2↑	SR@0.1↑	Avg. SR↑
Closed-loop	None	73.37	67.96	47.54	62.96
	SFT	71.75	62.07	34.62	56.15
	Online RL (SFPO)	**77.24**	**72.07**	**48.59**	**65.97**
Open-loop	None	52.87	44.39	**26.39**	41.22
	SFT	**54.40**	45.04	27.04	**42.16**
	Online RL (SFPO)	53.35	**45.68**	25.67	41.57

the successful execution with the lowest error is used as the fine-tuning target. The same three planners are also evaluated under open-loop execution, where each reference is generated once before execution. Under open-loop execution, SFT slightly improves over the fixed planner (52.87% to 54.40% SR@0.3 and 41.22% to 42.16% Avg. SR), and online RL changes the results only marginally (53.35% SR@0.3). Under closed-loop execution, however, SFT lowers SR@0.3 from 73.37% to 71.75% and Avg. SR from 62.96% to 56.15%, with the largest drop at the strictest threshold (47.54% to 34.62% SR@0.1). This is consistent with fitting near-distribution trajectories without a signal for recovering once replanning reaches states away from the demonstrations. Online RL instead assigns positive relative advantage to rollouts that recover and negative advantage to those that do not, and it is the only update that improves closed-loop performance (77.24% SR@0.3 and 65.97% Avg. SR).

B.5 Planner RL Ablation: Setup

This section provides the setup for the planner RL ablation in Sec. 3.4. We compare SFPO with FPO++ [19],[1] FPO++ with a behavior cloning anchor, ReinFlow [15] PPO with and without the anchor, and Flow-GRPO [16] with isotropic noise. All methods are trained on a single NVIDIA 4090 GPU with a fixed learning rate of 2×10^{-6}. To ensure fair comparison and maintain the stability of the effective sample size for group-relative optimization, ReinFlow and the GRPO-based methods, which share the same noise injection scheme, use a batch size of 1536 environments. In contrast, FPO++ consumes significantly more memory and is restricted to a batch size of 512. Since FPO++ and ReinFlow require a critic network for value estimation, they share the same critic network architecture. Detailed implementation and hyperparameter settings are provided in Appendix E.2.

Since FPO++ does not use denoising-path noise injection, its BC anchor is implemented as an output-space distance to a frozen base planner. Appendix E.2.2 defines this distance and the frozen baseline.

B.6 Planner Prediction and Loss Parameterization

We compare our planner, which predicts the clean state x_0 and is trained with an x-space loss (X-base), against a standard rectified-flow planner that predicts the velocity v and is trained with a velocity-space loss (V-base). The two variants differ only in this parameterization and share the same architecture, training data, and number of training epochs. We evaluate open-loop autoregressive generation on the official OMOMO test split against the ground-truth motions, following the metric definitions of OMOMO [8]: foot sliding (FS), mean per-joint position error (MPJPE), root and object translation errors (T_{root}, T_{obj}), object orientation error (O_{obj}), and mean per-joint velocity error (MPJVE).

Table 6: Planner prediction and loss parameterization, evaluated by open-loop autoregressive generation on the official OMOMO test split. V-base predicts v with a velocity-space loss, and X-base predicts x_0 with an x-space loss.

Variant	FS↓	MPJPE↓	T_{root} ↓	T_{obj} ↓	O_{obj} ↓	MPJVE↓
V-base	0.579	**11.71**	**7.40**	17.25	1.098	1.122
X-base (ours)	**0.350**	11.87	7.43	**16.06**	**1.046**	**0.686**

Table 6 shows that X-base reduces foot sliding by 40% and velocity error by 39% relative to V-base and lowers object translation and orientation errors, while V-base is marginally better on MPJPE (11.71 vs. 11.87) and root translation (7.40 vs. 7.43). Because the controller must track these references under contact-rich dynamics, we adopt X-base for InfCoiL.

[1] Our maximum sample size is limited to 8 instead of 16 or 32 because of memory constraints.

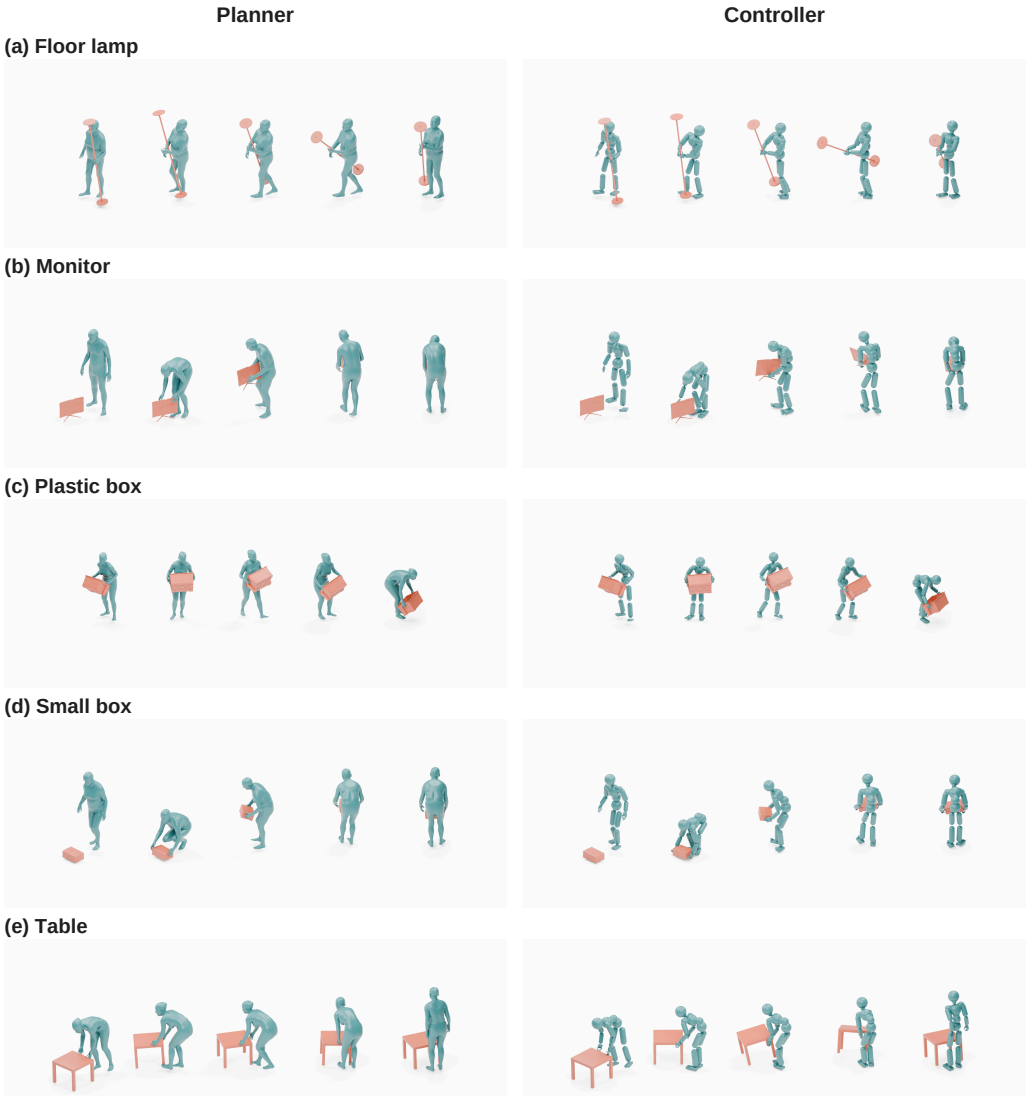

Figure 3: Planner references (left) and controller motions (right) for floorlamp, monitor, plasticbox, smallbox, and table. Each row shows matching frames from one episode, ordered from left to right.

B.7 Generalization to Random Morphology–Object Combinations

Figures 3 and 4 compare planner references and controller motions from the same episodes across different body shapes and objects. Across diverse objects such as floorlamp, plasticbox, smalltable, and trashcan, the planner produces semantically meaningful SMPL-X interaction references, and the humanoid controller tracks these references in simulation. Although the controller embodiment differs from the planner morphology and may adjust poses, contact timing, and object trajectories during execution, it largely preserves the intended interaction semantics, such as bending toward the object, lifting, carrying, or repositioning it. These results demonstrate that the closed-loop planner-controller design improves robustness to morphology-object variation and supports generalization beyond fixed body-object pairs.

B.8 Qualitative Failure Analysis

Failure modes of open-loop execution. Figure 2 shows typical failures of the open-loop baselines. In the suitcase example, InterMimic student tips the suitcase over while approaching it and never establishes a lifting grasp, whereas SimGenHOI* lifts the suitcase but carries it in a tilted pose and does not complete the task. In the monitor example, both baselines lose control of the object after contact, and SimGenHOI* drops it to the ground. In the small-box example, InterMimic student briefly lifts the box but loses its grasp, and the rollout is recorded as a failure. These examples illustrate a limitation of open-loop execution. The reference is generated



Figure 4: Planner references (left) and controller motions (right) for smalltable, suitcase, trashcan, woodchair, and tripod. Each row shows matching frames from one episode, ordered from left to right.

once before execution; after a deviation in human-object contact, the controller continues tracking the fixed reference, and errors can accumulate until termination.

Effect of closed-loop replanning and coordinated learning. InfCoiL completes the same tasks because the planner replans every N_r steps from the realized human, object, and contact states, so each new reference window starts from the actual object configuration rather than the one assumed before execution. Table 1 quantifies this effect. With the same frozen planner and controller, closed-loop replanning raises SR from 49.23% to 56.09% and reduces pelvis/object goal errors from 57.15/75.51 cm to 46.89/69.83 cm. The remaining gap suggests that replanning alone cannot compensate for references that the controller fails to execute precisely around contact. Controller adaptation raises SR to 73.37%, and subsequent planner learning moves references toward behaviors that the adapted controller executes reliably, reaching 77.24% (C→P) and 81.19% (C→P→C).

Remaining failures. Inspecting failed InfCoiL rollouts, we observe two recurring regimes. First, objects with complex contact geometry or less reliable simulated dynamics, such as the monitor, remain harder to grasp and manipulate stably than simple boxes. Second, when an object is initialized in an airborne or otherwise strongly off-demonstration state, the planner may not react before termination, because the demonstrations contain little or no pre-contact object motion. Neither regime is resolved by replanning or coordinated learning on the current data, and addressing them would likely require broader coverage of object states in the training data and more accurate contact modeling.

C System Implementation Details

This section describes the concrete planner-controller interface, planner tokenization and conditioning, controller observation design, and action parameterization needed to reproduce InfCoiL.

C.1 Planner Representation and Conditioning

The high-level planner predicts structured HOI motion tokens over a short horizon. Each frame token has 490 channels:

$$490 = 72_{\text{body}} + 90_{\text{hand}} + 312_{\text{rot}} + 3_{\text{obj trans}} + 9_{\text{obj rot}} + 4_{\text{contact}}. \tag{11}$$

The first term contains 24 body joints in 3D, the second contains 30 hand joints in 3D, the rotation term stores 52 global joint rotations in 6D representation, the object state contains object translation and a 3×3 rotation matrix, and the final term stores coarse hand-object contact labels. During closed-loop execution, the planner receives the latest simulated humanoid, object, and contact history, converts it back into this token format, and clamps the prefix frames during sampling. This makes the generated future window continuous with the physical state actually reached by the controller.

The planner uses a Transformer encoder with model width 512, 16 attention heads, 8 layers, GELU activations, dropout 0.1, and sinusoidal positional encoding. To keep each token compact, the 490-dimensional frame feature is split into two motion sub-tokens of 220 and 270 dimensions, embedded separately, interleaved over time, and then decoded back to the original channel order. We train the model as a rectified-flow planner with predict-x and loss-x parameterization, using logit-normal time sampling, Gaussian noise scale 1, and 10 Euler sampling steps at inference. The supervised planner is trained with batch size 512, learning rate 10^{-4}, EMA decay 0.9998, and auxiliary losses on object keypoints, forward-kinematic hand/foot positions, and finger-tip local offsets.

Conditioning is encoded as prefix tokens before the motion tokens. Following the OMOMO-based language-conditioned HOI setting of CHOIS [2], the language instruction is encoded by a pretrained CLIP ViT-L/14@336px text encoder [35], which is kept frozen in evaluation mode rather than trained from scratch. Its L2-normalized 768-dimensional sentence embedding is projected to the planner width by a trainable two-layer MLP and fused with a dynamic progress embedding computed from the current progress index, the target end index, and the sequence length. The pelvis goal is encoded from the local horizontal (x, z) coordinates, while the object goal is encoded from its 3D local position. Object geometry is represented by BPS features using 1024 object surface points and projected by a two-layer MLP into the planner width. Morphology is encoded from the 16-dimensional SMPL-X shape vector β through a linear projection. During supervised training, the morphology token is randomly dropped with probability 0.1 for classifier-free-style regularization, and normal inference uses the conditional branch. Since the text embedding is part of the conditioning context c, it affects the predicted clean state and its induced velocity at every flow step. The interface accepts free-form sentences that the CLIP tokenizer can process, rather than a finite set of prompt IDs, without architectural changes. Its effective semantic range, however, follows the interaction types and phrasing of the OMOMO/CHOIS annotations, since no additional language corpus is used. Deterministic evaluation uses the selected annotation of each test sequence.

The planner operates in a receding-horizon manner. The HOI model samples a 16-token future window with 4 additional prefix frames, and the sampled trajectory is converted to a 30 Hz simulation reference by interpolation. In our default closed-loop setting, this yields an executable reference horizon of $N_f = 42$ frames, while replanning is triggered every $N_r = 24$ simulation steps. Since the horizon is longer than the replanning interval, consecutive plans overlap, allowing the controller to track a near-future reference while the planner periodically refreshes later motion from updated physical feedback.

C.2 Controller Observations, Architecture, and Actions

The controller uses a unified student policy for all SMPL-X body shapes. A single per-step tracking observation has 1599 dimensions and contains heading-normalized humanoid proprioception, contact indicators, object state, reference-state differences, future reference features, and interaction-geometry features between selected body keypoints and object surface points. The student policy receives four temporal versions of this tracking observation at offsets $\{0, 1, 4, 16\}$, concatenated with the 16-dimensional SMPL-X shape vector:

$$o_t^{\text{student}} = \left[\beta,\ o_t^0,\ o_t^1,\ o_t^4,\ o_t^{16}\right] \in \mathbb{R}^{6412}. \tag{12}$$

This observation design gives the controller both the current tracking error and near-future reference context, while the shape vector tells the shared policy which morphology is being controlled.

The student actor is a compact Transformer encoder. The shape vector is embedded into a 256-dimensional shape token, and each of the four 1599-dimensional observation vectors is embedded into a 256-dimensional observation token. The five tokens are summed with sinusoidal positional encoding and processed by a 3-layer Transformer encoder with 4 attention heads, 512 feed-forward width, GELU activations, and zero dropout. The

output corresponding to the shape token is used as a CLS-style summary and projected to the Gaussian action mean. The critic follows the standard actor-critic implementation used by the underlying RL framework and predicts the value from the normalized full observation.

The action is a 153-dimensional vector corresponding to the actuated humanoid joint degrees of freedom. The simulated SMPL-X humanoid has 52 rigid bodies. Excluding the floating root, the controller actuates 51 joints with a 3D local rotation target per joint. Actions are clipped to $[-1, 1]$ and converted to PD position targets by

$$q^{\text{target}} = q^{\text{offset}} + q^{\text{scale}} \odot a, \tag{13}$$

where q^{offset} and q^{scale} are computed from joint limits. The simulator then applies position-control drives to track these targets. Thus, the policy does not directly output torques but normalized joint-position targets for a low-level PD controller.

D Training Details

This section collects controller distillation, contact-aware initialization, closed-loop fine-tuning, and rollout settings for coordinated planner-controller learning. Training InfCoiL takes about one week on four NVIDIA GeForce RTX 4090 GPUs.

D.1 Controller Distillation and Fine-Tuning

The unified controller is initialized by distilling subject-specific InterMimic-style teachers into one multi-morphology student with the distillation-to-RL curriculum of InterMimic [4]. During rollout, each environment is assigned a subject-object pair. At every simulation step and independently for each environment, the executed action is the corresponding teacher's sampled and clipped action with probability $p_{\text{tea}}(e)$ and the student's action otherwise, where $p_{\text{tea}}(e) = \text{clip}\,(1 - (e - 500)/5000,\ 0,\ 1)$ and e is the training epoch. Rollouts are thus fully teacher-driven for the first 500 epochs and fully student-driven after epoch 5500. Regardless of which policy acts, the teacher's action mean on the visited state is recorded as the imitation target for the student's action mean, providing a DAgger-style signal on states increasingly induced by the student.

The training objective follows a gated schedule. For $e \leq 5000$, only the imitation loss is used. For $5000 < e \leq 6000$, the critic loss is added with a weight increasing linearly to its full value, after which it remains at full weight. From $e > 7000$, we track an exponential moving average of the critic's explained variance, updated as $0.99\,\text{EV} + 0.01\,\text{ev}$ at every minibatch. The PPO actor loss is applied to a minibatch only when this average has reached 0.6 for three consecutive minibatch checks. Otherwise, the minibatch falls back to the critic and full-weight imitation losses. Letting n be the cumulative number of gated minibatch updates, the actor and imitation weights are $\min(n/4000, 1)$ and $\max(1 - n/4000, 0.1)$, respectively.

We use PPO with discount factor 0.99, GAE parameter 0.95, learning rate 2×10^{-5}, clipping threshold 0.2, horizon length 16, minibatch size 16384, 6 mini-epochs, critic coefficient 5, gradient-norm limit 1, and action-bound loss coefficient 10. The Gaussian exploration parameter is fixed during student training. The base imitation coefficient is 1.0 and is annealed to 0.1 within the same pretraining run as described above. In closed-loop controller fine-tuning it is set to 0 so that the controller is optimized directly for the closed-loop reward induced by planner-generated references.

For contact-rich phases, we use contact-aware state initialization during controller pretraining. Demonstration frames near hand-object contact onset and release receive higher reset probability, controlled by a contact window of 15 frames and sampling weight scale 3.0. This exposes the student to difficult grasping and release states more often than uniform random initialization, reducing the chance that rare but important contact transitions are under-trained.

D.2 Closed-Loop Training Details

Closed-loop controller fine-tuning uses the same simulator, action parameterization, and student architecture as pretraining, but the reference trajectory is generated online by the planner. In the default all-subject configuration, the environment uses 4096 parallel simulations, start-state initialization, 500-step episode length, 24-step replanning interval, and 42-frame reference horizon. The reward keeps humanoid and object tracking terms active, adds terminal success and termination rewards, and disables the interaction-graph and contact-consistency terms during closed-loop coordinated learning.

For planner fine-tuning, SFPO evaluates a sampled plan only after execution by the controller and physics simulator. Rollouts are grouped by identical task semantics, including subject, object, sequence, reset state, start time, and goal conditions, so samples in the same group differ only by the stochastic flow transitions used for planner exploration. We compute a sequence-level score from the time-averaged transition reward plus terminal task reward, normalize it within the group, clip the resulting advantage, and assign the same sequence advantage

to all planner transitions produced during that rollout. This avoids comparing absolute rewards across tasks with different objects, body shapes, or reset difficulties.

E Planner RL and Algorithmic Derivations

This section provides the planner-side derivations and optimization details omitted from the main paper, including SFPO, FPO baselines, group-relative advantages, and the coordinated learning schedule.

E.1 Derivations for SFPO

This section derives the SFPO transition likelihood of Section 2.2.1. The planner update follows the ReinFlow/Flow-GRPO stochastic-denoising view: selected denoising transitions are treated as Gaussian policy steps, the old transition log-probabilities are stored during rollout, and the current log-probabilities are recomputed on the same denoising path during PPO-style updates. Our modification is semantic: the transition variance is diagonal and channel-dependent, reflecting that body translations, rotations, hand joints, object states, and contact labels tolerate different perturbation scales. This keeps the online RL path aligned with Flow-GRPO's stochastic flow policy while avoiding an explicit density-score estimator for HOI motion tokens.

Consider a K-step Euler denoising sampler. Let

$$z_k = x_{\tau_k}, \quad k = 1, \dots, K+1, \tag{14}$$

where $\tau_k \in [0, 1]$ denotes the flow matching time at the k-th denoising step, with $\tau_1 = 1$ and $\tau_{K+1} = 0$. Stochastic transitions are applied to a fixed subset of S denoising steps,

$$\mathcal{S} \subseteq \{1, \dots, K\}, \quad |\mathcal{S}| = S. \tag{15}$$

For steps outside this subset ($k \notin \mathcal{S}$), the planner retains the original deterministic updates. For the selected steps ($k \in \mathcal{S}$), we model the transition as a Gaussian policy. Selecting only a subset of denoising steps sacrifices the accuracy of the full-path log-probability but significantly reduces computation and memory overhead. When S equals K, this formulation degenerates to a full-chain stochastic flow policy. In all experiments, we use $K = 10$ and make only the final step stochastic, *i.e.*, $\mathcal{S} = \{K\}$ and $S = 1$, corresponding to the transition from $\tau_K = 0.1$ to $\tau_{K+1} = 0$. This subset is fixed across all rollouts rather than resampled, and the preceding $K - 1$ steps use deterministic Euler updates.

For the selected denoising steps, we start from the marginal-preserving reverse-time SDE used by Flow-GRPO:

$$dx_\tau = \left[v_\theta(x_\tau, \tau, c_t) - \frac{1}{2}\sigma_\tau^2 \nabla_{x_\tau} \log p_\tau(x_\tau \mid c_t) \right] d\tau + \sigma_\tau dW_\tau, \tag{16}$$

where v_θ denotes the velocity field of the planner, c_t is the condition at simulation step t, and σ_τ controls the stochasticity. For rectified flow with linear interpolation, the score can be rewritten as

$$\nabla_{x_\tau} \log p_\tau(x_\tau \mid c_t) = -\frac{x_\tau + (1-\tau) v_\theta(x_\tau, \tau, c_t)}{\tau}. \tag{17}$$

Substitution yields a score-free drift expressed only through the current state and velocity:

$$\bar{v}_\theta(x_\tau, \tau, c_t) = v_\theta(x_\tau, \tau, c_t) + \frac{\sigma_\tau^2}{2\tau} \left[x_\tau + (1-\tau) v_\theta(x_\tau, \tau, c_t) \right]. \tag{18}$$

We use this score-free rectified-flow form rather than estimating $\nabla \log p_\tau$ with an additional model. For HOI tokens, an explicit score estimator would need to model a mixed representation containing normalized continuous body/object coordinates, rotations, hand articulation, autoregressive fixed prefixes, and contact-like semantic channels. Errors in such a score are particularly harmful because they enter the drift at every stochastic denoising step and can move samples into contact-inconsistent states before the physics controller evaluates them. The velocity-substituted form retains Flow-GRPO's Gaussian transition likelihood while keeping the planner architecture unchanged.

We further replace the isotropic diffusion coefficient with a semantic diagonal coefficient:

$$\Gamma_\tau = \lambda_{\mathrm{n}} \sqrt{\frac{\tau}{1-\tau}} \, \mathrm{Diag}(\kappa), \tag{19}$$

where λ_{n} denotes the global noise level and $\kappa \in \mathbb{R}_+^D$ assigns a fixed scale to each channel, so that channel j has standard deviation proportional to κ_j. The scales are set by hand, constant within each of six channel groups, and not updated during RL. Table 7 lists the scale of each channel group. Object translation receives the largest perturbation and rotations the smallest. The noise levels compared in Table 2 ($\lambda_{\mathrm{n}} \in \{0.05, 0.1, 0.2\}$) change only the global level. In the discrete sampler, the time factor is computed as $\sqrt{\tau_k / \max(1 - \tilde{\tau}_k, 10^{-6})}$, with $\tilde{\tau}_k = \tau_{k+1}$ at the endpoint $\tau_k = 1$ and $\tilde{\tau}_k = \tau_k$ otherwise, which does not affect our experiments, where the

Table 7: Semantic noise scales κ used by SFPO, constant within each channel group of the 490-dimensional planner frame.

Channel group	Dim.	κ
Body joint positions	72	0.35
Hand joints	90	0.175
Human global rotations	312	0.07
Object translation	3	0.7
Object rotation	9	0.07
Contact labels	4	0.175

only stochastic step has $\tau_K = 0.1$. The isotropic Flow-GRPO baseline in Table 2 uses the same time schedule with identical entries of κ across all perturbable channels. With this diagonal covariance, the score-free drift in Eq. (18) is applied elementwise as

$$\bar{v}_\theta(z_k, \tau_k, c_t) = v_\theta(z_k, \tau_k, c_t) + \frac{1}{2\tau_k}\Gamma_{\tau_k}\Gamma_{\tau_k}^\top \left[z_k + (1-\tau_k)v_\theta(z_k, \tau_k, c_t)\right]. \tag{20}$$

Through Euler-Maruyama discretization, the transition for the selected steps becomes

$$z_{k+1} = \mu_{\theta,k} + \Gamma_{\tau_k}\sqrt{-\Delta\tau_k}\epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, I), \tag{21}$$

where the mean is given by

$$\mu_{\theta,k} = z_k + \bar{v}_\theta(z_k, \tau_k, c_t)\Delta\tau_k, \quad \Delta\tau_k = \tau_{k+1} - \tau_k < 0. \tag{22}$$

Therefore, the policy probability is expressed as

$$\pi_\theta(z_{k+1} \mid z_k, c_t) = \mathcal{N}\left(\mu_{\theta,k}, (-\Delta\tau_k)\Gamma_{\tau_k}\Gamma_{\tau_k}^\top\right). \tag{23}$$

During the planner rollout phase, the frozen snapshot θ_{old} generates plans and we record their transition log-probabilities. In the update phase, we recompute them on the same denoising path with the current planner. Each replanning event t of the g-th rollout under task condition c_i forms one training sample, whose context c_t combines c_i with the executed history. Let Ω denote the generated elements (u, d) of a planned window, *i.e.*, its non-prefix frames u times the $D = 490$ channels d. We use a normalized log-probability score, in which the initial-noise term $z_1 \sim \mathcal{N}(0, I)$ and each selected transition term are computed as the mean element-wise log-density over the same set Ω, and these $S + 1$ terms are then averaged:

$$\bar{\ell}_\theta^{i,g,t} = \text{clip}\left(\frac{1}{S+1}\left[\frac{1}{|\Omega|}\sum_{(u,d)\in\Omega}\log\mathcal{N}(z_{1,u,d}; 0, 1) + \sum_{k\in\mathcal{S}}\frac{1}{|\Omega|}\sum_{(u,d)\in\Omega}\log\pi_{\theta,u,d}(z_{k+1,u,d} \mid z_k, c_t)\right], -10, 10\right), \tag{24}$$

where $\pi_{\theta,u,d}$ is the univariate marginal of Eq. (5) for element (u, d), and all states are those of sample (i, g, t). The importance ratio is

$$\rho_\theta^{i,g,t} = \exp\left(\bar{\ell}_\theta^{i,g,t} - \bar{\ell}_{\theta_{\text{old}}}^{i,g,t}\right). \tag{25}$$

The initial-noise term does not depend on θ and cancels in the difference when clipping is inactive, but it halves the scale of the log-ratio for $S = 1$. Together with the element-wise averaging, this makes ρ a normalized surrogate ratio rather than a ratio of joint Gaussian densities over the planned window, and the ratio clipping range ϵ_c in Table 8 is set on this scale.

Sampling G rollouts under the same task condition c_i yields sequence-level rewards $R_i^1, \ldots, R_i^G$ after execution by the controller and the physics engine, and every replanning event of rollout g receives the clipped group-relative advantage A_i^g of Eq. (37). Appendix E.3 describes how groups are formed and how invalid rollouts and zero-variance groups are excluded. The planner objective is

$$\begin{aligned} \mathcal{J}_{\text{clip}} &= \mathbb{E}_{i,g,t}\left[\min\left(\rho_\theta^{i,g,t}A_i^g,\ \text{clip}(\rho_\theta^{i,g,t}, 1-\epsilon_c, 1+\epsilon_c)A_i^g\right)\right], \\ \mathcal{J}_{\text{SFPO}} &= \mathcal{J}_{\text{clip}} - \lambda_{\text{KL}}\,\mathbb{E}_{i,g,t,\,k\in\mathcal{S}}\left[D_{\text{KL}}\big(\pi_\theta(\cdot \mid z_k, c_t) \,\|\, \pi_{\text{fix}}(\cdot \mid z_k, c_t)\big)\right]. \end{aligned} \tag{26}$$

The expectation runs over all valid replanning samples, so longer rollouts contribute more samples. Ratios are neither multiplied across the replanning events of a rollout nor normalized by their number. In practice, the loss is first averaged over valid samples within each log-probability microbatch and then weighted by the microbatch size, which differs slightly from the global valid-sample mean when a microbatch contains invalid samples. No fixed-reference KL or BC penalty is applied in the reported SFPO configuration ($\lambda_{\text{KL}} = 0$). Table 8 lists the SFPO hyperparameters.

Table 8: SFPO hyperparameters used in all reported planner fine-tuning phases.

Hyperparameter	Value
Group size G	16
Denoising steps K / stochastic steps S	10 / 1 (final step, τ: $0.1 \to 0$)
Global noise level λ_{n}	0.1
Ratio clipping ϵ_{c}	0.03
Advantage clipping $A_{\max}$	5
Group skipped if reward std $\leq$	10^{-6}
Std floor in advantage normalization	10^{-12}
Log-probability clipping	$[-10, 10]$
Reference regularizer weight λ_{KL}	0
PPO epochs per iteration	up to 5, early stop when approx. KL exceeds 0.01
Optimizer minibatch	50,000 replanning samples
Log-probability microbatch	1,024 samples
Planner sampling chunk	128
Learning rate	2×10^{-6}

E.2 FPO Baseline, BC Anchor, and Critic Details

The FPO ablations reported in the experiments are distinct from the SFPO planner update above. They reuse the same closed-loop planner-controller rollout and the same reward and scoring pipeline, but the actor ratio is not computed from ReinFlow denoising-path log-probabilities. Instead, FPO constructs a conditional-flow-matching surrogate ratio on the final denoised plan.

E.2.1 FPO Ratio

During an FPO rollout, the planner first generates a final denoised motion window x_0 using the planner's flow-matching sampler. For each generated window, FPO samples auxiliary pairs (ϵ_s, τ_s), $s = 1, \ldots, N_{\mathrm{aux}}$, with $\epsilon_s \sim \mathcal{N}(0, I)$ and $\tau_s \in (0, 1)$, and interpolates

$$x_{\tau_s} = \tau_s \epsilon_s + (1 - \tau_s) x_0. \tag{27}$$

The autoregressive fixed-prefix frames are clamped before the model forward pass. Since our planner predicts x_0, FPO uses the induced velocity of Eq. (4),

$$v_\theta(x_{\tau_s}, \tau_s, c) = \frac{x_{\tau_s} - f_\theta(x_{\tau_s}, \tau_s, c)}{\tau_s}, \tag{28}$$

and the target velocity is $v^\star = \epsilon_s - x_0$. FPO computes the masked CFM loss

$$\mathcal{L}_\theta^{\mathrm{CFM}}(x_0, \epsilon_s, \tau_s, c) = \frac{1}{|\Omega|} \sum_{(u,d) \in \Omega} \left(v_{\theta,u,d} - v^\star_{u,d} \right)^2, \tag{29}$$

where Ω is the generated-element set of Eq. (24), which excludes the fixed autoregressive prefix. The rollout stores $\mathcal{L}_{\mathrm{old}}^{\mathrm{CFM}}$ from the rollout model θ_{old}. During training, the current model recomputes $\mathcal{L}_\theta^{\mathrm{CFM}}$ on the same $(x_0, \epsilon_s, \tau_s, c)$ samples and forms

$$\rho_\theta^{\mathrm{FPO}} = \exp\left(\mathcal{L}_{\mathrm{old}}^{\mathrm{CFM}} - \mathcal{L}_\theta^{\mathrm{CFM}} \right). \tag{30}$$

The actor loss then applies the same proximal trust-region principle as PPO, with clipping to prevent extreme surrogate-ratio updates. This FPO ratio should therefore be interpreted as a CFM-loss surrogate baseline, not as the denoising-transition likelihood used by SFPO.

E.2.2 BC Anchor

The BC Anchor constrains FPO updates to remain close to a frozen motion prior in output space. The baseline model is initialized from an explicit checkpoint if provided. Otherwise, it is copied from the fixed base reference when that exists, or from the initial planner parameters. Its parameters are frozen throughout FPO training.

For the main anchored FPO setting, we use an output-distribution distance over the same auxiliary (ϵ_s, τ_s) samples used by the FPO ratio. Let $f_\theta^s = f_\theta(x_{\tau_s}, \tau_s, c)$ and f_{fix}^s denote the current and frozen-baseline

clean-state predictions for sample $s = 1, \ldots, N_{\text{aux}}$. The implementation supports three anchor forms:

$$\mathcal{L}_{\text{BC}}^{\text{MSE}} = \frac{1}{|\Omega| N_{\text{aux}}} \sum_{s=1}^{N_{\text{aux}}} \sum_{(u,d)\in\Omega} \left(f_{\theta,u,d}^{s} - f_{\text{fix},u,d}^{s}\right)^2, \tag{31}$$

$$\mathcal{L}_{\text{BC}}^{\text{W2}} = \frac{1}{|\Omega|} \sum_{(u,d)\in\Omega} \left[\left(\bar{f}_{\theta,u,d} - \bar{f}_{\text{fix},u,d}\right)^2 + \left(\varsigma_{\theta,u,d} - \varsigma_{\text{fix},u,d}\right)^2\right], \tag{32}$$

$$\mathcal{L}_{\text{BC}}^{\text{KL}} = \frac{1}{2|\Omega|} \sum_{(u,d)\in\Omega} \left[\log \frac{\varsigma_{\text{fix},u,d}^2}{\varsigma_{\theta,u,d}^2} + \frac{\varsigma_{\theta,u,d}^2 + \left(\bar{f}_{\theta,u,d} - \bar{f}_{\text{fix},u,d}\right)^2}{\varsigma_{\text{fix},u,d}^2} - 1\right], \tag{33}$$

where $\bar{f}$ and ς are the mean and standard deviation of f^s across the auxiliary samples. The reported *FPO w/o BC Anchor* ablation keeps the FPO surrogate-ratio machinery but removes this frozen-baseline output regularizer by setting the BC loss coefficient to zero. This isolates the effect of anchoring the planner to its supervised motion prior from the effect of the FPO update itself.

E.2.3 Planner Critic Architecture

The FPO baselines use a planner-level critic that estimates the state value before each reference window is executed. It is an MLP with LayerNorm and SiLU activations whose base input concatenates the planner's fixed prefix states, the current simulator state (root height, local body positions, rotations, and velocities, and contact indicators), and the task goals. Optional branches encode privileged object and goal features, the controller observation, the CLIP text embedding, the object BPS features, and the flow-matching conditions. The critic never observes planner samples or controller actions, and its value estimates serve as the GAE baseline of the surrogate-ratio update. SFPO does not use a critic, since group-relative normalization of the sequence scores replaces this learned baseline.

E.3 Reward Terms and Group-Relative Advantage

The reward terms of Sec. 2.2.2 take the forms

$$\begin{aligned} r_{\text{humanoid}}^{t} &= \exp\left(-\alpha_{\text{h}} e_{\text{h}}^{t}\right), \qquad r_{\text{object}}^{t} = \exp\left(-\alpha_{\text{o}} e_{\text{o}}^{t}\right), \\ R_{\text{success}} &= \mathbb{1}[\text{task success}], \qquad R_{\text{term}} = -\mathbb{1}[\text{episode ends without success}], \end{aligned} \tag{34}$$

where e_{h}^{t} is the mean per-joint position error of the humanoid and e_{o}^{t} is the object center position error at step t, and $\alpha_{\text{h}}, \alpha_{\text{o}} > 0$ are scale coefficients.

Although the controller and the planner share the same reward components, they utilize these rewards differently. The controller operates as a stepwise policy. Consequently, its PPO reward at each simulation step t is formulated as

$$r_t^{\text{ctrl}} = R_{\text{trans}}^{t} + \mathbb{1}[t = T] R_{\text{task}}, \tag{35}$$

where T denotes the termination time of the current episode. The controller receives the tracking-based transition reward at every step and an additional task reward at the end of the episode.

Each planner sample is scored by the time-averaged sequence reward R^{seq} of Eq. (10), which reflects both the tracking feasibility of its references during closed-loop execution and whether the task is accomplished.

In planner group-relative policy optimization, we evaluate sample quality via closed-loop controller and physics engine execution, rather than designing intermediate rewards for individual denoised reference windows $\hat{x}_{t:t+N_f-1}$. An optimization group forms a semantically consistent reset cohort. Sharing identical subject-object pairs, reference indices, start times, contact reset states, and task conditions, group rollouts ensure uniform task semantics. Members differ solely through independent SFPO denoising noise, yielding multiple stochastic samples under identical interaction conditions. Given the i-th cohort condition c_i, we sample G rollouts per group. The g-th rollout generates reference motion $\hat{x}_{t:t+N_f-1}^{i,g}$. The controller executes the initial N_r steps, continuing closed-loop replanning until success, failure, or timeout. The rollout then receives the sequence reward R_i^g, *i.e.*, R^{seq} of Eq. (10) evaluated on this rollout with its own length T.

Excluding invalid or incomplete rollouts, we define the valid sample set as $\mathcal{V}_i = \{g \in \{1, \ldots, G\} \mid$ rollout g has terminated and R_i^g is finite$\}$. This yields the group mean and standard deviation:

$$\bar{R}_i = \frac{1}{|\mathcal{V}_i|} \sum_{g\in\mathcal{V}_i} R_i^g, \quad \varsigma_i = \text{std}\left(\{R_i^g : g \in \mathcal{V}_i\}\right), \tag{36}$$

where ς_i is the unbiased sample standard deviation. Groups with $\varsigma_i \leq 10^{-6}$ carry no relative preference and are skipped. For the remaining groups, we assign a clipped group-relative advantage to each valid rollout:

$$A_i^g = \text{clip}\left(\frac{R_i^g - \bar{R}_i}{\max(\varsigma_i, 10^{-12})}, -A_{\max}, A_{\max}\right). \tag{37}$$

Algorithm 1 Coordinated planner-controller learning in InfCoiL.

Require: planner P_θ pretrained by flow matching, controller π_ϕ from distillation and open-loop RL, minimum phase lengths $N_{\mathrm{c}}^{\min}$, $N_{\mathrm{p}}^{\min}$

while training budget not exhausted **do**
Controller phase (planner frozen)
  repeat
    Roll out in closed loop: replan with P_θ every N_r steps and act with π_ϕ
    Update π_ϕ with PPO on the HOI-centric reward (Sec. 2.2.2)
  until at least $N_{\mathrm{c}}^{\min}$ iterations and tracking quality or success rate stops improving
Planner phase (controller frozen)
  repeat
    $\theta_{\mathrm{old}} \leftarrow \theta$
    for all task conditions c_i in the batch **do**
      for $g = 1, \ldots, G$ **do**
        Execute with π_ϕ, replanning with $\pi_{\theta_{\mathrm{old}}}$ (Eq. (5)), until termination
        Store $\bar{\ell}_{\theta_{\mathrm{old}}}^{i,g,t}$ for every replanning event t and compute R_i^g
      end for
      Compute A_i^g by Eq. (37), skipping the group if its reward variance is near zero
    end for
    for up to 5 epochs, with KL early stopping **do**
      Recompute $\bar{\ell}_\theta^{i,g,t}$ on the stored paths and update θ by ascending $\mathcal{J}_{\mathrm{SFPO}}$ (Eq. (7))
    end for
  until at least $N_{\mathrm{p}}^{\min}$ iterations and sequence score or success rate stops improving
end while

Groups with fewer than $V_{\min}$ valid rollouts, which can occur when rollouts terminate at different times or batches are small, are also masked out. When a rollout contains several replanning events, its advantage A_i^g is assigned to every valid replanning event, each forming one training sample, so that every generated reference window, not only the last one, receives credit for the episode outcome.

E.4 Details of Coordinated Learning Paradigm

InfCoiL adopts a coordinated learning paradigm alternating between controller and planner adaptation. Before joint training, we pretrain the planner via supervised flow matching and derive a student controller via behavioral cloning distillation and open-loop reinforcement learning. Subsequently, we alternately optimize both components in the closed-loop environment. The system switches to planner reinforcement learning when controller tracking performance or success rate plateaus, and reverts to controller reinforcement learning when planner sequence-level success or rollout scores plateau. This alternation lets the controller adapt to the evolving planner distribution, while the planner can exploit the larger set of references that the improved controller can execute.

During the controller adaptation phase, we fix the planner and collect closed-loop rollouts. The controller is updated with proximal policy optimization using the stepwise reward r_t^{ctrl}. We track moving averages of the tracking quality and the success rate together with their best values in the current phase. To prevent frequent switching, each controller phase lasts at least $N_{\mathrm{c}}^{\min}$ iterations (Algorithm 1). After that, the system switches to the planner phase once either moving average fails to improve on its phase best by more than a tolerance δ_{c}.

During the planner adaptation phase, the controller is fixed. The system collects grouped closed-loop rollouts by sampling references using SFPO noise. We compute group-relative advantages and update the planner accordingly. Analogously, we track moving averages of the sequence reward and the success rate, and after at least $N_{\mathrm{p}}^{\min}$ iterations the system returns to the controller phase once either fails to improve on its phase best by more than a tolerance δ_{p}.

Algorithm 1 summarizes the procedure. In the reported experiments, the phase lengths are fixed rather than triggered by plateaus: InfCoiL (C→P) uses 10k controller iterations followed by 500 planner iterations, and InfCoiL (C→P→C) adds 4k further controller iterations, so the tolerances δ_{c} and δ_{p} play no role in the reported results.

F Additional Discussion

This section provides extended literature context, outlines future directions, and discusses broader impacts beyond the main paper.

F.1 Extended Related Work

Physics-based Humanoid Control. Physics-based humanoid control aims to produce physically plausible full-body motions by controlling simulated humanoids [13]. A common paradigm is to train RL policies to track motion capture datasets at scale, resulting in general-purpose motion trackers [36, 37]. Generalized behavior-cloning frameworks further retarget mocap data to diverse humanoid morphologies and learn whole-body imitation policies with DAgger-style distillation and PPO [38]. Subsequent methods distill reusable motion priors from such trackers through adversarial imitation [39, 40] or latent skill representations including VAEs and vector-quantized models [37, 41, 42, 43]. Recent diffusion-based controllers further model state-action trajectories for flexible task guidance [44, 45]. Furthermore, text- and task-conditioned control has also been explored through behavior cloning and multi-stage reinforcement learning [46, 47, 48], but these policies often provide limited motion diversity compared with kinematic generative models. To exploit the strengths of both paradigms, planner-controller frameworks combine diffusion-based kinematic planners with physics-based trackers [49, 50, 7]. However, these methods largely consider the humanoid alone, without explicitly modeling object dynamics or sustained contact-rich human-object coupling, which limits their applicability to physically grounded HOI tasks.

Generative Planning with Closed-Loop Control and RL. Among planner-controller frameworks for humanoids, CLoSD [7] closes the loop by replanning from simulated states, but adapts only the tracking controller while the diffusion planner stays fixed. In robotics, AdaptDiffuser [51] self-evolves a diffusion planner by generating synthetic trajectories, filtering them, and fine-tuning the planner on the retained ones. Hierarchical diffusion-RL planning [52] trains a low-level RL policy beneath a high-level diffusion path planner, and RLingua [53] and AnyBipe [54] use large language models to guide RL through generated controllers or reward design rather than training a generative motion planner. A separate line fine-tunes diffusion and flow policies with online RL, via stochastic denoising chains [55, 15, 16] or flow-matching surrogate likelihoods [17, 19]. In these settings, generated samples are scored directly by the environment or a reward model. InfCoiL differs in which component is optimized by execution feedback: it updates the flow-matching planner on-policy from complete closed-loop rollouts, where each plan is scored only after being executed by a separately learned controller under contact-rich dynamics, and alternates this with controller adaptation so that both sides of the planner-controller interface move together.

Generative Models for Human-Object Interaction. Large-scale 4D human-object interaction (HOI) datasets provide paired human-object motion and geometry annotations for learning interaction generators [56, 57, 8, 58]. A common direction is to enrich diffusion models with object motion, contact, language, geometry, and waypoint conditions, enabling physics-informed prediction, object-guided manipulation, and text-controllable interaction synthesis [1, 8, 20, 2]. Later methods further strengthen the conditioning structure with semantic-geometric context, object relations, kinematic-chain representations, unseen-object generalization, and interaction dynamics [21, 22, 23, 24, 25]. Recent studies also revisit the guidance and denoising mechanism itself: [3] obtains interaction guidance without auxiliary classifiers by asynchronously denoising body, hand, and object components, and introduces human shape as conditions. Some methods have explored consistency models for motion generation to accelerate inference, which is crucial for online planning [26, 27, 28]. However, flow-based HOI generation remains underexplored. Moreover, existing generators are typically trained and sampled without physical execution feedback, limiting their ability to adapt plans to the states actually reached by the controller.

Physics-based Human-Object Interaction. Physics-based HOI methods aim to execute interactions under contact-rich dynamics rather than only synthesize kinematic trajectories. A direct paradigm is to learn task policies with reinforcement learning from manually designed rewards, which can produce executable behaviors for sports, games, and object manipulation but often depends on task-specific engineering and lacks demonstration-level realism [29, 30, 31, 59, 60]. Imitation-based methods reduce this burden by tracking human-object demonstrations in simulation, enabling contact-rich and reusable interaction skills [32, 33, 34]. However, these skills remain bounded by the objects, motions, and interaction modes covered by demonstrations, limiting their ability to synthesize diverse long-horizon behaviors from fine-grained textual and spatial instructions. Recent large-scale methods improve coverage by learning whole-body controllers from broader and imperfect mocap interaction data, including unified controllers for heterogeneous objects and sparse-intention-conditioned control [4, 5]. These methods ground HOI in simulation, but they remain less expressive than text-conditioned generative planners in semantic controllability and motion diversity. Hierarchical systems further combine interaction generators with imitation policies to connect planning and physical tracking [6], but they still generate static references before tracking, leaving the planner essentially open-loop. Our work addresses this gap with a closed-loop planner-controller framework, where the planner repeatedly conditions on language, goals, morphology, object states, and recent physical feedback, and coordinated reinforcement learning adapts both modules to each other.

F.2 Future Directions

Beyond the limitations discussed in Sec. 5, several directions could extend InfCoiL. First, the planner currently consumes structured object states and goal specifications. Replacing them with perception-driven inputs would connect InfCoiL to vision-driven embodied agents that explore and act in open-world scenes [61], and to physical world models and world action models that jointly predict visual dynamics and actions [62, 63, 64], which could also provide learned dynamics priors beyond the kinematic mocap prior used by our planner. Second, the closed-loop planner-controller coordination could be evaluated beyond whole-body HOI, for example on bimanual and mobile robotic manipulation benchmarks with diverse objects and tasks [65, 66, 67, 68], where coordinated learning would carry over through domain-specific execution rewards and planner-controller interface definitions. Third, transferring the multi-morphology controller to physical humanoids would require accurate real-time tracking of the object states consumed by the planner, cross-embodiment motion retargeting and imitation [38] as well as automated sim-to-real training and evaluation pipelines [54], together with adaptation to hardware constraints, sensing noise, and real contact dynamics. Robot platforms such as Unitree G1 lie far outside the SMPL-X shape variation used in our experiments, so preserving the same interaction diversity would also require dynamics-aware adjustment of proportions, joints, contacts, balance, and actuation. Fourth, InfCoiL could serve as an executable skill for language-model agents that write control programs [69], invoke learned skills [70], or evolve reward programs for humanoid loco-manipulation [71]. A recent EBench study finds that such agents benefit from invoking post-trained policies but are limited by fixed-instruction execution without subgoals [72], a gap that the instruction- and goal-conditioned replanning of InfCoiL could fill. Fifth, the injected semantic noise is small, with final-step standard deviations below 10^{-2} in normalized units, so numerical discrepancies between rollout and update passes, e.g., from reduced-precision arithmetic [73, 74], can be amplified in the log-probability ratio. Since low precision has been linked to loss spikes [75] and to training-inference mismatch in RL fine-tuning [76], the numerical precision and the hand-set noise scales κ deserve a joint study. Finally, the stochastic-flow construction of SFPO does not depend on the motion representation and may extend to other flow-matching diffusion transformer (DiT) planners, in particular Mixture-of-Transformers variants with heterogeneous modality streams, for which the semantic covariance can be allocated per stream.

F.3 Broader Impact

InfCoiL studies text- and goal-conditioned physics-based human-object interaction for controllable character animation and simulated embodied agents. The proposed framework may benefit applications such as animation authoring, virtual character control, robotics simulation, assistive embodied agents, and synthetic data generation, by reducing the manual effort required to design physically plausible interaction behaviors. At the same time, improving the controllability and realism of simulated human-object interactions may raise potential risks if such techniques are used to generate misleading visual content, imitate human behaviors without proper disclosure, or transfer unsafe interaction policies to real robotic systems. Our work is conducted in simulation and focuses on physically grounded interaction modeling rather than human identity replication. We encourage responsible use of the released code and models, including clear disclosure when generated motions are used in media applications, careful safety validation before any real-world robotic deployment, and respect for the licenses and consent conditions of the motion and object datasets used for training and evaluation.