Beyond Mimicry: Learning Whole-Body Human-Humanoid Interaction from Human-Human Demonstrations

Supplementary Material

Overview

This supplementary material is organized into six sections (A–F) for clarity and reproducibility.

Terminology. We refer to our retargeting pipeline as **PAIR** (*Physics-Aware Interaction Retargeting*) and our policy as **D-STAR** (*Decoupled Spatio-Temporal Action Reasoner*). Within D-STAR, we decouple policy reasoning into Phase Attention (**PA**) and Multi-Scale Spatial (**MSS**), which are fused by the diffusion head.

- **A. Method Implementation Details** (Sec. A): Retargeting pipeline (Sec. A.1), hierarchical policy (Sec. A.2), training (Sec. A.3), and low-level controllers/execution (Sec. A.4; root interface conversion Sec. A.4.3; controller modularity Sec. A.4.5).
- **B. Dataset Details and Analysis** (Sec. B; built from the Inter-X dataset [6]).
- **C. Experimental Protocols, Metrics & Additional Diagnostics** (Sec. C): Unified notation & symbol table (Sec. C.1, Table S7), retargeting evaluation and diagnostics (Sec. C.2), policy evaluation and additional deployment diagnostics (Sec. C.3).
- **D. Dataset Retargeting for Sim-to-Real** (Sec. D).
- **E. Observation & Action Specification** (Sec. E; Table S1).
- **F. Scheduler, Densification & Diffusion Details** (Sec. F; includes densification and fusion Algorithms 8 and 9).

A. Method Implementation Details

This section provides a comprehensive breakdown of our proposed framework. We first detail **PAIR** (*Physics-Aware Interaction Retargeting*; Sec. A.1), explaining the principled design of our objective function and optimization strategy used to generate physically consistent HHoI data.

We then dissect **D-STAR** (*Decoupled Spatio-Temporal Action Reasoner*; Sec. A.2), elucidating the core principle of decoupled reasoning and its realization through specialized modules. In D-STAR, we decouple policy reasoning into **PA** and **MSS**. Finally, we specify the exact **Training Procedures and Loss Functions** (Sec. A.3) for both the high-level policy and the low-level controller.

A.1. PAIR: Physics-Aware Interaction Retargeting

This section provides a comprehensive breakdown of our implementation. We architect our solution as a sequence of targeted fixes, where each component of our objective function addresses a specific failure mode inherent in conventional retargeting.

A.1.1. Objective Function Components

We define the retargeting objective as a weighted sum of four terms that jointly encourage kinematic similarity and interaction semantics:

$$\mathcal{L}_{\text{retarget}} = w_{\text{kin}}\mathcal{L}_{\text{kin}} + w_{\text{con}}\mathcal{L}_{\text{con}} + w_{\text{hum}}\mathcal{L}_{\text{hum}} + w_{\text{reg}}\mathcal{L}_{\text{reg}},$$
$$\mathcal{L}_{\text{reg}} = \alpha\,\mathcal{L}_{\text{temp}} + \beta\,\mathcal{L}_{\text{pose}}.$$

Unless otherwise stated, distances are measured in meters and joint angles in radians. The frame rate is 50 Hz.

Kinematic similarity ($\mathcal{L}_{\text{kin}}$). We encourage the robot's joint-space configuration to match a morphologically reshaped source human skeleton, Reshaped(H_s). The reshaping solves for SMPL [4] shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$ and a global scale s that best fit the robot's bone lengths:

$$\boldsymbol{\beta}^*,\, s^* \;=\; \arg\min_{\boldsymbol{\beta},\, s} \sum_{i \in \mathcal{M}} \left\| \mathbf{j}_i^R - s \cdot \left(\mathbf{j}_i^{\text{SMPL}}(\boldsymbol{\beta}) - \mathbf{j}_0^{\text{SMPL}}(\boldsymbol{\beta})\right) - \mathbf{j}_0^R \right\|_2^2,$$

where $\mathcal{M}$ maps robot joints to SMPL joints (correspondence listed below). We optimize this objective for 500 iterations using Adam (learning rate 0.1).

Robot (G1)		SMPL
pelvis	→	Pelvis
left_hip_pitch_link	→	L_Hip
left_knee_link	→	L_Knee
left_ankle_roll_link	→	L_Ankle
right_hip_pitch_link	→	R_Hip
right_knee_link	→	R_Knee
right_ankle_roll_link	→	R_Ankle
left_shoulder_roll_link	→	L_Shoulder
left_elbow_link	→	L_Elbow
left_hand_link	→	L_Hand
right_shoulder_roll_link	→	R_Shoulder
right_elbow_link	→	R_Elbow
right_hand_link	→	R_Hand
head_link	→	Head
left_toe_link	→	L_Toe
right_toe_link	→	R_Toe

Interaction-contact consistency ($\mathcal{L}_{\text{con}}$). To maintain the interaction geometry, we enforce consistency between the original and optimized holistic spatial relationships. Let $\mathcal{K}_H$ and $\mathcal{K}_R$ denote selected human/robot keypoint sets (e.g., head, shoulders, elbows, wrists, hands); define the

joint set $\mathcal{K} = \mathcal{K}_H \cup \mathcal{K}_R$ with $N = |\mathcal{K}|$. For time t, let $\mathbf{x}_t(i) \in \mathbb{R}^3$ be the position of keypoint $i \in \mathcal{K}$, and define the pairwise distance matrix

$$[D_t]_{ij} \;=\; \|\mathbf{x}_t(i) - \mathbf{x}_t(j)\|_2, \quad D_t \in \mathbb{R}^{N\times N}.$$

We compute D_t^{orig} from the original interaction and D_t^{opt} after optimization, and minimize

$$\mathcal{L}_{\text{con}} \;=\; \frac{1}{T}\sum_{t=1}^{T} \left\| D_t^{\text{opt}} - D_t^{\text{orig}} \right\|_F^2 .$$

This construction measures configuration-level geometry rather than a single contact, improving robustness to minor local deviations.

Human motion fidelity ($\mathcal{L}_{\text{hum}}$). To avoid unrealistic adjustments on the human partner, we penalize deviations from the original human pose while allowing selective upper-body adaptation. Let $\tilde{\mathbf{p}}_{H,t}$ be the optimized human joint positions and $\mathbf{p}_{H,t}$ the original ones. Define the upper-body index set $\mathcal{J}_{\text{UA}} = \{\text{shoulder}, \text{elbow}, \text{wrist}\}$ bilaterally. Then

$$\mathcal{L}_{\text{hum}} \;=\; \frac{1}{T}\sum_{t=1}^{T} \sum_{j\in\mathcal{J}_{\text{UA}}} \|\tilde{\mathbf{p}}_{H,t}(j) - \mathbf{p}_{H,t}(j)\|_2^2 ,$$

which enables plausible upper-limb adjustments (e.g., a slight arm lift) while preserving core posture.

Physical plausibility regularizers ($\mathcal{L}_{\text{reg}}$). We use standard temporal and pose regularization. Let $\mathbf{q}_t \in \mathbb{R}^D$ be the robot joint angles at time t (D DoF). We use discrete derivatives

$$\mathbf{v}_t = \mathbf{q}_{t+1} - \mathbf{q}_t,$$
$$\mathbf{a}_t = \mathbf{v}_{t+1} - \mathbf{v}_t = \mathbf{q}_{t+1} - 2\mathbf{q}_t + \mathbf{q}_{t-1}.$$

The losses are

$$\mathcal{L}_{\text{temp}} = \frac{1}{T-1}\sum_{t=1}^{T-1} \|\mathbf{v}_t\|_2^2 + \frac{w_a}{T-2}\sum_{t=1}^{T-2} \|\mathbf{a}_t\|_2^2,$$

$$\mathcal{L}_{\text{pose}} = \frac{1}{T\cdot D}\sum_{t=1}^{T}\sum_{d=1}^{D} q_{t,d}^2,$$

where w_a balances velocity and acceleration penalties (numeric values and stage-wise schedules are given in Sec. A.1.2).

A.1.2. Two-Stage Optimization Strategy and Hyperparameters

Optimizing all terms from scratch can converge to poor local minima (e.g., "near-miss" contacts). We therefore adopt a coarse-to-fine, two-stage schedule.

Optimization process. Sequence-level retargeting is optimized for **200** iterations with Adam. The above **500** iterations refer solely to the one-time SMPL shape fitting (Sec. A.1.1); they are not part of the per-sequence optimization.

- **Stage 1 (coarse initialization)** (iterations 1–150, LR= 0.02): optimize the full objective with a moderate contact weight ($w_{\text{con}} = 0.25$) to obtain a kinematically feasible trajectory.
- **Stage 2 (contact refinement)** (iterations 151–200, LR= 0.005): increase the contact weight by $10\times$ to emphasize interaction consistency ($w_{\text{con}} = 2.5$), turning near-miss cases into consistent contact while keeping other weights unchanged.

Loss weights (Stage 1).

- Kinematic similarity ($\mathcal{L}_{\text{kin}}$): $w_{\text{kin}} = 1.0$
- Contact consistency ($\mathcal{L}_{\text{con}}$): $w_{\text{con}} = 0.25$
- Human motion fidelity ($\mathcal{L}_{\text{hum}}$): $w_{\text{hum}} = 0.25$
- Temporal coherence ($\mathcal{L}_{\text{temp}}$): $w_{\text{temp}} = 5.0$
- Pose regularization ($\mathcal{L}_{\text{pose}}$): $w_{\text{pose}} = 0.02$

In the notation of Sec. A.1.1 where $\mathcal{L}_{\text{reg}} = \alpha\,\mathcal{L}_{\text{temp}} + \beta\,\mathcal{L}_{\text{pose}}$, these correspond to $w_{\text{reg}}\alpha = w_{\text{temp}}$ and $w_{\text{reg}}\beta = w_{\text{pose}}$.

Constraints and post-processing. During optimization, joint angles are clamped to hardware limits $[\mathbf{q}_{\min}, \mathbf{q}_{\max}]$. After optimization, we apply a Gaussian smoothing filter (kernel size $= 5$, $\sigma = 0.75$) to the joint trajectories for improved temporal smoothness.

A.1.3. Experimental Environment

The pipeline is implemented in PyTorch. All optimizations are performed on a server with dual AMD EPYC 7B13 CPUs, without GPU acceleration.

A.2. D-STAR: Decoupled Spatio-Temporal Action Reasoner

D-STAR is architected around a core principle: **decoupling spatio-temporal reasoning**. To achieve robust and responsive interaction, an agent must solve two distinct sub-problems: inferring the temporal phase of the interaction (*when* to act) and understanding the precise geometric relationship with the partner (*where* to act). In D-STAR, we explicitly address this with parallel, specialized modules—Phase Attention (**PA**) and Multi-Scale Spatial (**MSS**)—which are then integrated to produce a final, context-aware action. This section details the components of this architecture.

A.2.1. Input Representation: Observation Space and Temporal Encoding

Observation Space Design. Each observation frame $\mathbf{O}_t$ concatenates human SMPL features and robot proprioception:

- **Human SMPL features:** root translation $s_t^H \in \mathbb{R}^3$ and joint positions $j_t^H \in \mathbb{R}^{72}$ (24 joints $\times$ 3D);

- **Robot proprioception:** $s_t^R \in \mathbb{R}^{74}$ including base linear & angular velocities (3+3), gravity projection (3), joint positions/velocities (29+29), root translation offset (3), and root orientation quaternion (4).

Thus $\mathbf{O}_t = \mathrm{Concat}(s_t^H, j_t^H, s_t^R) \in \mathbb{R}^{149}$.

Long–Short Temporal Encoder (LSTE). We use a dual-stream encoder to capture both long-term context and short-term reactivity:

$$\mathbf{f}_t^{long} = E_{long}(\mathbf{O}_{t-h+1:t}) = \mathrm{TranS}_{long}(\{\mathbf{O}_i\}_{i=t-h+1}^{t}),$$
$$\mathbf{f}_t^{short} = E_{short}(\mathbf{O}_{t-h'+1:t}) = \mathrm{TranS}_{short}(\{\mathbf{O}_i\}_{i=t-h'+1}^{t}).$$

We set $h = 12$ and $h' = 6$ *(short stream observes only the most recent 6 frames for reactive precision, while the long stream attends to the full 12-frame buffer for phase context).* Each stream produces a **768-d** feature; their concatenation (**1536-d**) is linearly projected to a **256-d** temporal feature $\mathbf{f}_t^{temp}$ shared by both reasoning streams.

A.2.2. Temporal Reasoning: Phase Attention (PA)

To answer **"when to act"**, PA dynamically infers the current interaction phase and emphasizes phase-relevant cues. A phase classifier predicts a distribution $\mathbf{p}_t$ over $k = 3$ phases from $\mathbf{f}_t^{temp}$:

$$\mathbf{p}_t = \mathrm{Softmax}(\mathrm{MLP}_{phase}(\mathbf{f}_t^{temp})) \in \mathbb{R}^3.$$

Concurrently, $k = 3$ phase-specialized self-attention experts process $\mathbf{f}_t^{temp}$. The phase-aware temporal feature is a probability-weighted mixture:

$$\mathbf{f}_t^{phase} = \sum_{i=1}^{k} p_{t,i} \cdot \mathrm{SelfAttn}_i(\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t),$$
$$[\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t] = \mathrm{Proj}(\mathbf{f}_t^{temp})$$

and is projected to 256-d to match the temporal feature scale.

A.2.3. Spatial Reasoning: Multi-Scale Spatial (MSS)

To answer **"where to act"**, MSS builds a scale-aware geometric understanding. We segment space into three zones (values fixed throughout):

- **Near Field (0–0.3 m)**: precise contact operations,
- **Mid Field (0.3–0.8 m)**: coordinated gesture exchange,
- **Far Field (0.8–3.0 m)**: approach and planning.

Encoders. A **3D positional encoder** applies sinusoidal encodings to the relative position $\mathbf{p}_{rel}$. A **multi-scale distance encoder** processes Euclidean distance d via zone-specific MLPs gated by indicator functions. A **hierarchical orientation encoder** uses heading, projections, and rotational harmonics for approach-aware orientation.

Fusion. We first concatenate encoder outputs into a spatial vector $\mathbf{f}_{spatial}$, transform it into a dynamic interaction field $\mathbf{f}_{field}$ via an MLP, and then fuse it with temporal context through attention:

$$\mathbf{f}_t^{MSS} = \mathrm{FusionAttn}\big(\mathrm{Concat}(\mathbf{f}_{field}, \mathrm{MLP}_{proj}(\mathbf{f}_t^{temp}))\big),$$

followed by a projection to 128-d to obtain the final spatial feature.

A.2.4. Feature Fusion and Hierarchical Action Generation

With decoupled streams providing $\mathbf{f}_t^{phase}$ and $\mathbf{f}_t^{MSS}$, we integrate them with the shared temporal context and the language command embedding to form a global conditioning vector:

$$\mathbf{c}_t = \mathrm{Concat}(\mathbf{f}_t^{temp}, \mathbf{f}_t^{phase}, \mathbf{f}_t^{MSS}, \mathbf{l}), \quad \mathbf{l} \in \mathbb{R}^{768}.$$

Dimensionalities. $\mathbf{f}_t^{temp} \in \mathbb{R}^{256}$, $\mathbf{f}_t^{phase} \in \mathbb{R}^{256}$, $\mathbf{f}_t^{MSS} \in \mathbb{R}^{128}$, $\mathbf{l} \in \mathbb{R}^{768}$, thus $\mathbf{c}_t \in \mathbb{R}^{\mathbf{1408}}$.

A conditional diffusion policy $\mathcal{D}$ generates a high-level target:

$$\mathbf{a}_t^{target} = \mathcal{D}(\boldsymbol{\epsilon} \mid \mathbf{c}_t).$$

This target is executed by a pre-trained whole-body controller (WBC-Sim) to produce physically-consistent motor commands $\mathbf{a}_t$ that maintain balance and respect joint limits. *Implementation note.* In simulation we command the articulated joints (29-DoF) while using the root as an internal reference; on hardware the root component is converted to base commands via the standard base-velocity interface.

A.3. Training Procedures and Loss Functions

Our framework's hierarchical nature is mirrored in our training methodology. We employ a **decoupled training strategy** that optimizes the high-level interaction policy and the low-level whole-body controller independently. This separation of concerns allows each component to be trained with the most suitable objectives and data distributions, leading to a system that is both intelligent in its interaction planning and robust in its physical execution. This section details the training procedures and objectives for each component of our hierarchy.

A.3.1. High-Level Policy Training (Supervised Learning)

The high-level policy, which includes the temporal encoders and our decoupled spatio-temporal reasoning modules, is trained end-to-end via supervised learning on our curated HHoI dataset.

Objective Function. The policy is trained to optimize a composite loss function $\mathcal{L}_{total}$ that combines a primary action prediction objective with several auxiliary terms for regularization and phase supervision.

$$\mathcal{L}_{total} = \lambda_{act}\mathcal{L}_{action} + \lambda_{ph}\mathcal{L}_{phase} + \lambda_{aux}\mathcal{L}_{aux}$$

The high-level components are defined as follows:

- **Action Prediction Loss ($\mathcal{L}_{action}$):** The core objective for the diffusion model is a dimension-weighted MSE loss on the predicted action $\mathbf{a}^{target}$.
- **Phase Supervision Loss ($\mathcal{L}_{phase}$):** This supervises the Phase Attention module using a cross-entropy loss for classification ($\mathcal{L}_{cls}$) and a KL divergence loss for enforcing logical phase transitions ($\mathcal{L}_{trans}$).
- **Auxiliary Geometric Loss ($\mathcal{L}_{aux}$):** This term encourages plausible interactive behaviors by regularizing the geometric and relational aspects of the generated motion. It includes losses for facing orientation, keypoint positioning, spatial consistency, and more.

The detailed mathematical formulations for the primary and auxiliary losses are provided below.

Action Prediction Loss. The primary action loss is a weighted MSE over the predicted change in root translation $\Delta\mathbf{p}$, root orientation (quaternion) $\mathbf{q}$, and joint angles $\boldsymbol{\theta}$.

$$\begin{aligned}\mathcal{L}_{action} = \quad & w_{trans}\|\Delta\mathbf{p}_{pred} - \Delta\mathbf{p}_{gt}\|_2^2 \\ & + w_{rot}\|\mathbf{q}_{pred} - \mathbf{q}_{gt}\|_2^2 \\ & + w_{dof}\|\boldsymbol{\theta}_{pred} - \boldsymbol{\theta}_{gt}\|_2^2\end{aligned}$$

We use weights $w_{trans} = 10.0$, $w_{rot} = 10.0$, and $w_{dof} = 1.0$.

Auxiliary Loss Formulations. The components of $\mathcal{L}_{phase}$ and $\mathcal{L}_{aux}$ are defined as:

- **Human-Facing Orientation Loss ($\mathcal{L}_{face}$):** To ensure the robot maintains a natural orientation towards its partner, we penalize the deviation from a direct facing angle using the cosine distance between the robot's forward vector $\mathbf{v}_{fwd}^R$ and the vector to the human $\mathbf{v}_{R\to H}$.

$$\mathcal{L}_{face} = 1 - \frac{\mathbf{v}_{fwd}^R \cdot \mathbf{v}_{R\to H}}{\|\mathbf{v}_{fwd}^R\|_2 \|\mathbf{v}_{R\to H}\|_2}$$

- **Keypoint Position Loss ($\mathcal{L}_{pos}$):** To guide the high-level geometric structure, we apply an L2 loss on the predicted 3D positions of critical keypoints $\mathcal{K}$ (e.g., hands, head).

$$\mathcal{L}_{pos} = \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \|\mathbf{p}_{pred}^k - \mathbf{p}_{gt}^k\|_2^2$$

- **Logical Phase Transition Loss ($\mathcal{L}_{trans}$):** To enforce logical temporal progression, we use a KL divergence loss between the predicted phase transition probabilities $\mathbf{P}_{t\to t+1}$ and a predefined valid transition matrix $\mathbf{T}_{valid}$.

$$\mathcal{L}_{trans} = D_{KL}(\mathbf{T}_{valid} \parallel \mathbf{P}_{t\to t+1})$$

- **Spatial Consistency Loss ($\mathcal{L}_{spatial}$):** To maintain the holistic geometric relationship, we penalize the difference between the predicted and ground-truth inter-keypoint distance matrices (D_{pred}, D_{gt}) using the Frobenius norm.

$$\mathcal{L}_{spatial} = \|D_{pred} - D_{gt}\|_F^2$$

The weights for these auxiliary terms are set as follows: $w_{cls} = 0.0005$ (for the phase classifier), $w_{\text{phase\_KL}} = 0.005$, $w_{face} = 0.2$, $w_{pos} = 5.0$, and $w_{spatial} = 0.02$.

Table S1 enumerates the oracle/student observation channels and how they are consumed during training and evaluation.

Implementation Parameters. The high-level policy is trained with the following configuration:

- **Temporal Window:** 12-frame observation history including the current frame (t) at 50 Hz; predict **5 anchors** at $t + \{0, 0.5, 1.0, 1.5, 2.0\}$ s; anchors are densified to 50 Hz and fused across consecutive 5 Hz calls via bilinear interpolation.
- **Network Dimensions:** LSTElong/short streams each 768-d; concatenation 1536-d with linear projection to 256-d. Phase Attention (PA) uses 256-d features; Multi-Scale Spatial (MSS) uses 128-d. Global conditioning for the diffusion head is 640-d from concatenation (256, 256, 128), with 768-d language features.
- **Training Scale:** 1000 epochs, Adam (1×10^{-4}), batch size 2048 (stride=25), seed 42.
- **Advanced Features:** Ego-centric processing for consistent spatial reasoning; auxiliary observations include robot proprioception and 768-d language features.

A.3.2. Low-Level Whole-Body Controller (WBC-Sim): Oracle→Student Distillation

Overview. We train the simulation controller in two steps: an oracle teacher on clean observations and a student distilled on noisy observations with 10-frame history. Both stages share the same robot and control interface (29-D joint-angle targets at 50 Hz), the same environment and motion-resampling interval (500 steps (50 Hz)), and the same base domain randomization; noise and DR curricula are disabled. The distilled student is the only controller used for all policy evaluations. Table S2 summarizes the teacher vs student configuration.

Privileged→Noisy Distillation. We frame WBC-Sim training as privileged→noisy distillation: the *oracle* is optimized on clean, *privileged* observations (including base linear and angular velocities), while the *student* operates on noisy, realistic observations *without* base linear velocity and with a 10-frame history. All evaluations use the distilled student.

Noise protocol. All *current-frame sensing* channels are perturbed with fixed per-channel uniform noise during student training; *history stacks inherit* the perturbation from their constituent frames and receive no extra injection; the *previous action* is left clean to prevent compounding errors. Table S3 details the per-channel perturbations.

Table S1. **WBC-Sim observation inventory and usage.** At train time, the oracle uses *clean* observations without history; the student consumes *noisy* observations with 10-frame history and the policy's 36-D reference (u_t). At evaluation, the fixed student is used; we retain the 10-frame history but do not inject synthetic noise. Only the WBC output (**29-D** joint-angle targets at **50 Hz**) is actuated in simulation.

Channel (actor)	Symbol	Dim	Oracle train	Student train	Eval (policy exec.)	History (frames)
Joint positions (target space)	q	29	✓	✓	✓	10
Joint velocities	$\dot{q}$	29	✓	✓	✓	10
Base angular velocity (body)	ω_b	3	✓	✓	✓	10
Base linear velocity (body)	v_b	3	✓	—	—	0
Gravity vector (body frame)	g_b	3	✓	✓	✓	10
Previous action (joint target)	a_{t-1}	29	✓	✓	✓	10
Policy reference (from high-level)	(u_t)	**36**	✗	✓	✓	10
Actor total (clean vs noisy)		*163 / 1090*	✓	✓	✓	
Critic total (clean vs noisy)		*361 / 1291*	✓	✓	✓	

Evaluation uses the fixed **student** controller with **10-frame history** and **no synthetic observation noise**.
Note. The oracle has access to privileged base linear velocity; the student excludes it.

Table S1. (continued): Observation and action channels. "Used" indicates channels that are fed to the policy; others are logging-only when present in raw data.

Channel	Dim.	Used
Human root translation	3	✓
SMPL joints (human)	72	✓
Base linear velocity	3	✓
Base angular velocity	3	✓
Gravity projection	3	✓
Robot joint positions	29	✓
Robot joint velocities	29	✓
Root translation (offset)	3	✓
Root orientation (quaternion)	4	✓
Reference action (policy head)	**36**	**—**
Joint targets (executed)	29	—
Root translation (reference)	3	—
Root orientation (reference)	4	—

Oracle (Teacher) — Clean Observations. Actor/critic receive noise-free raw channels (163-D / 361-D). We optimize an Upper-Body-Emphasized Tracking Reward with PPO using an MLP(512,256,128), clip 0.2, γ 0.99, λ 0.95, entropy 0.01, lr $1e{-3}$. Base domain randomization (mass/friction/PD/push up to 1.0 m/s) is enabled; curricula are off. A frozen checkpoint is used for distillation.

Student — Pure Action-Matching Distillation. Inputs are Noisy Observations with 10-Frame History (1090-D / 1291-D). We train with a pure distillation loss

$$\mathcal{L}_{\text{distill}} = \left\| \mu_s\left(o_{\text{noisy}}\right) - \mu_t\left(o_{\text{clean}}\right) \right\|_2^2,$$

freezing the teacher and the policy std (init = 0, min = 0). Network/optimizer mirror the teacher (lr $1e{-3}$). **Actions and history channels are noise-free**, while other keys use fixed per-key uniform noise (no curriculum). DR settings match the teacher.

Reward (key terms). The tracking suite emphasizes upper-body targets (e.g., root_position_xy 3.0, body-position (upper-body) 4.0, SMPL-hands-ori 5.0) with standard stability/effort penalties and a termination cost of -100.

A.4. Low-Level Controllers & Execution

This section details execution controllers in simulation and on the real robot, and outlines the interface conversion and robustness adaptations.

A.4.1. Simulation Controller (WBC-Sim).

We execute 29-D joint-angle targets (rad) at 50 Hz, produced by WBC-Sim. Here, x_t denotes robot proprioception (joint states, base signals, gravity, previous action, etc.). At each step, WBC-Sim takes as input (i) robot proprioception and (ii) the policy's 36-D reference u_t (29 joint targets + root translation and unit quaternion), and tracks these references under joint-limit and rate constraints. Only the 29-D joint targets are actuated in simulation; the root translation/rotation are used internally by WBC-Sim as reference signals (not directly actuated). All quantitative policy results in the main paper use the same WBC-Sim for fairness across methods. This controller corresponds to the distilled student described in Sec. A.3.2. For clarity, the policy observation used by the learning agent differs from the controller observation summarized here; this distinction is intentional.

$$a_t = \text{WBC-Sim}(x_t, u_t),$$

where $u_t \in \mathbb{R}^{36}$ is the policy reference and $a_t \in \mathbb{R}^{29}$ are joint-angle targets executed at 50 Hz.

Table S2. Teacher vs Student configuration for WBC-Sim distillation.

Component	Obs. Set	Dim (act/crit)	Noise	History	Curric.	Objective
Oracle (Teacher)	Clean observations (+ base linear velocity)	163 / 361	Off	0	Off	PPO w/ tracking reward
Student (WBC-Sim)	Noisy observations with 10-frame history (– base linear velocity)	1090 / 1291	On (per-key; actions & history off)	10	Off	Action-mean MSE to teacher

Table S3. **Noise model for student distillation.** We apply per-channel, zero-mean uniform noise during *student training* only; the oracle uses clean observations and evaluation does not inject synthetic noise. History and action channels are not perturbed.

Channel group	Distribution	Magnitude (symbolic)	Applied to
Joint positions q	$U(-\delta_q, \delta_q)$	δ_q (const.)	Student train
Joint velocities $\dot{q}$	$U(-\delta_{\dot{q}}, \delta_{\dot{q}})$	$\delta_{\dot{q}}$ (const.)	Student train
Base ang. vel. ω_b	$U(-\delta_\omega, \delta_\omega)$	δ_ω (const.)	Student train
Gravity vector g_b	$U(-\delta_g, \delta_g)$	δ_g (const.)	Student train
Prev. action a_{t-1}, History stacks		*no perturbation*	
Policy reference u_t (36-D)	$U(-\delta_u, \delta_u)$	δ_u (const.)	Student train
Curriculum		*disabled (fixed magnitudes throughout training)*	
Evaluation		*no synthetic noise injected*	

A.4.2. Real-World Controller (HOMIE-based, 27 DoF).

For real-robot demos we employ a HOMIE-based pre-trained controller [1], with the waist roll/pitch locked. A vertically mounted Logitech C1000e streams 1280×720@30 fps RGB to the control PC (Ryzen 9700X, RTX 4090), and 4D-Humans [3] provides single-RGB SMPL estimates after one-time camera-to-base calibration. HOMIE exposes a root-velocity interface $(v_x, v_y, \dot{\psi})$ (yaw rate); our planning model outputs root position/rotation, so we apply a deterministic conversion prior to execution (see Sec. A.4.3). Real-robot demonstrations are qualitative; quantitative comparisons are reported in simulation using WBC-Sim.

A.4.3. Root Interface Conversion (Position-to-Velocity Control)

Since the HOMIE[1] controller expects velocity commands $(v_x, v_y, \dot{\psi})$ while our policy predicts root position targets, we implement a closed-loop controller with active braking to ensure precise stopping. Instead of simple finite-difference conversion, we employ a distance-to-go logic based on onboard odometry.

We issue a constant forward reference velocity $v_{ref} = 0.2$ m/s to drive the robot. During execution, we continuously monitor the residual distance, defined as the difference between the policy's cumulative predicted displacement and the actual distance traversed via odometry. To mitigate inertial drift, we implement an **active braking mechanism**: when the residual distance drops below a proximity threshold ($\delta_{dist} \leq 0.15$ m) while the robot maintains significant momentum (current speed $v_{curr} \geq 0.1$ m/s), we issue a reverse velocity command ($v_{brake} = -0.2$ m/s) to rapidly decelerate and stabilize the robot at the target location.

A.4.4. Phase-Specific Trajectory Adaptation for Stability

To enhance execution robustness in simulation, we perform a deterministic adaptation of the reference trajectory based on the interaction phase. The **Act** phase is strictly preserved to maintain the fidelity of the learned interaction intent. For the **Preparation** and **Follow-Up** phases, we employ smoothed primitives to ensure stable transitions and prevent transient oscillations:

- **Neutral Upper-Body Pose ($q_{neutral}$):** We define a stable, task-agnostic upper-body configuration used as the anchor for non-interaction phases. The joint angles (in radians) are set as follows:
  - *Left Arm:* Shoulder (pitch/roll/yaw) = $[0.0, 0.3, 0.0]$, Elbow = 1.0, Wrist = $[0.0, 0.0, 0.0]$.
  - *Right Arm:* Shoulder (pitch/roll/yaw) = $[0.0, -0.3, 0.0]$, Elbow = 1.0, Wrist = $[0.0, 0.0, 0.0]$.
- **Preparation Phase:** To simulate diverse approach scenarios, we initialize the robot's root position randomly within a 1.0 m radius of the original sequence's starting point, while maintaining the original orientation. The robot's root pose (position and rotation) and upper-body joints are then generated via *bilinear interpolation* from this randomized state to the first frame of the Act phase, ensuring a smooth approach trajectory.
- **Follow-Up Phase:** Upon completion of the interaction (end of Act phase), the robot's root is kept stationary. The upper-body joints are bilinearly interpolated from the last frame of the Act phase back to $q_{neutral}$, ensuring a stable disengagement.

A.4.5. Controller Modularity.

Our framework is controller-agnostic: WBC-Sim and the HOMIE-based controller can be replaced by alternative whole-body controllers. In particular, GMT [2] is compatible with our action interface; we leave a systematic comparison for future work.

A.4.6. Controller-Space Jitter Diagnostics

Occasional oscillations in simulation mainly stem from aggressive high-bandwidth root tracking in WBC-Sim rather

than high-frequency artifacts in the policy outputs. Table S4 compares pre-WBC trajectories and controller-tracked executions for both D-STAR and Diffusion Policy across WBC-Sim, HOMIE, and TWIST2. Pre-WBC HF ratio remains low and similar between D-STAR and Diffusion Policy, while controller-level root tracking amplifies high-frequency components in WBC-Sim.

B. Dataset Details and Analysis

In this section, we detail the construction of our Human-Humanoid Interaction (HHoI) dataset. We emphasize that this dataset is not merely a collection of motions, but a **strategically engineered resource**. The principled task selection, rigorous quality control, and, most critically, the detailed phase annotations were all designed with the explicit goal of creating a benchmark that could validate our core hypothesis: that genuine interaction requires decoupling spatial reasoning. The structure of this dataset is therefore intrinsically linked to the architecture of our policy.

B.1. Data Source and Task Selection Rationale

Our data foundation is the Inter-X dataset [6], a large-scale repository of human-human interactions. To construct a comprehensive benchmark for HHoI, our task selection was not arbitrary but driven by a principled strategy to cover a taxonomy of interaction types crucial for humanoid robotics:

- **Contact-Rich Interactions** (Hug, Handshake): Tasks demanding precise physical contact, stability, and management of close-proximity geometry.
- **Dynamic Gestural Interactions** (High-Five, Wave): Tasks requiring temporal synchronization and spatial accuracy without sustained contact.
- **Socially-Encoded Interactions** (Bend, Fly-Kiss): Tasks that rely on conveying social intent through whole-body gestures, often at a distance.

This curated selection of six tasks ensures our policy is evaluated across a diverse and representative range of physical and social challenges inherent to HHoI.

B.2. Curation Pipeline and Quality Control

Following task selection, each candidate sequence from Inter-X passed through a rigorous, multi-stage curation pipeline to ensure its suitability for training a robust HHoI policy. We enforced strict quality gates to filter out unsuitable data:

- **Motion Fidelity:** We excluded samples with significant tracking errors, occlusions, or unnatural motion artifacts that would introduce noise into the training process.
- **Interaction Clarity:** We selected samples exhibiting clear initiation, execution, and completion phases, ensuring the full arc of an interaction was captured.
- **Kinematic Feasibility:** We ensured that the captured interactions occur within a spatial volume and pose range that is reasonably achievable by our target humanoid robot.
- **Temporal Completeness:** We retained only samples containing the complete interaction sequence, from the initial approach to the final departure.

Only sequences that passed all four gates were admitted into our final dataset, guaranteeing a high-quality foundation for our experiments.

B.3. Filtering Outcomes and Failure Cases

The filtering stage removes infeasible outliers rather than hard but meaningful interactions. In total, 30.05% of candidate sequences are discarded. Table S5 compares aggregate statistics before and after filtering: contact richness slightly increases, while extreme dynamics decrease marginally, indicating that the retained set better preserves interaction structure without introducing implausible motions. Fig. S2 shows representative filtered cases.

B.4. Phase Annotation for Interaction Understanding

A cornerstone of our work, and a key enabler for our policy's temporal reasoning, is the detailed manual annotation of interaction phases. Recognizing that interactions are not monolithic but possess a distinct temporal structure, we manually segmented each curated sequence into three semantically meaningful phases:

- **Preparation Phase:** The initial approach and positioning, where agents navigate and adjust their orientation towards each other.
- **Act Phase:** The core execution of the interaction, including contact establishment (e.g., Hug, Handshake), gesture performance (e.g., Wave, High-Five), or expression delivery (e.g., Bend, Fly-Kiss).
- **Follow-up Phase:** The completion and disengagement, involving contact release, returning to a neutral posture, and spatial separation.

This annotation process represents a significant human effort and provides the critical supervision signal for our Phase-Aware Attention module (Main. Sec. 4.2). It directly enables our policy to learn *when* to act—a capability that conventional imitation learning approaches inherently lack.

B.5. Final Dataset Statistics and Properties

The output of this comprehensive pipeline—from principled selection to manual annotation—is the HHoI dataset used throughout our experiments. The key statistics, which reflect the dataset's scale and diversity across the selected tasks, are summarized in Table S6.

Temporal and Spatial Richness: The final dataset spans approximately 6.9 hours of interaction data (at 50 Hz), with

Table S4. **Controller-space jitter diagnostics.** HF ratio denotes the fraction of joint-velocity energy above 5 Hz; jerk is computed on the executed trajectories.

Method	Controller	HF ratio↓	DOF jerk↓	Root jerk↓	e_{root} (m/deg)↓	e_{dof} (rad/rad/s)↓
D-STAR	pre-WBC	0.162	286.2	115.6	–	–
D-STAR	WBC-Sim	0.614	2259.5	227.6	0.031 / 13.8	0.304 / 1.558
D-STAR	HOMIE	0.297	827.4	171.2	0.349 / 52.6	0.370 / 0.644
D-STAR	TWIST2	0.203	499.3	120.0	0.367 / 48.7	0.249 / 0.436
DP	pre-WBC	0.165	180.0	119.3	–	–
DP	WBC-Sim	0.591	2146.1	201.0	0.034 / 9.9	0.242 / 1.479
DP	HOMIE	0.285	833.5	175.8	0.391 / 50.0	0.348 / 0.634
DP	TWIST2	0.195	498.0	126.2	0.358 / 47.5	0.226 / 0.446

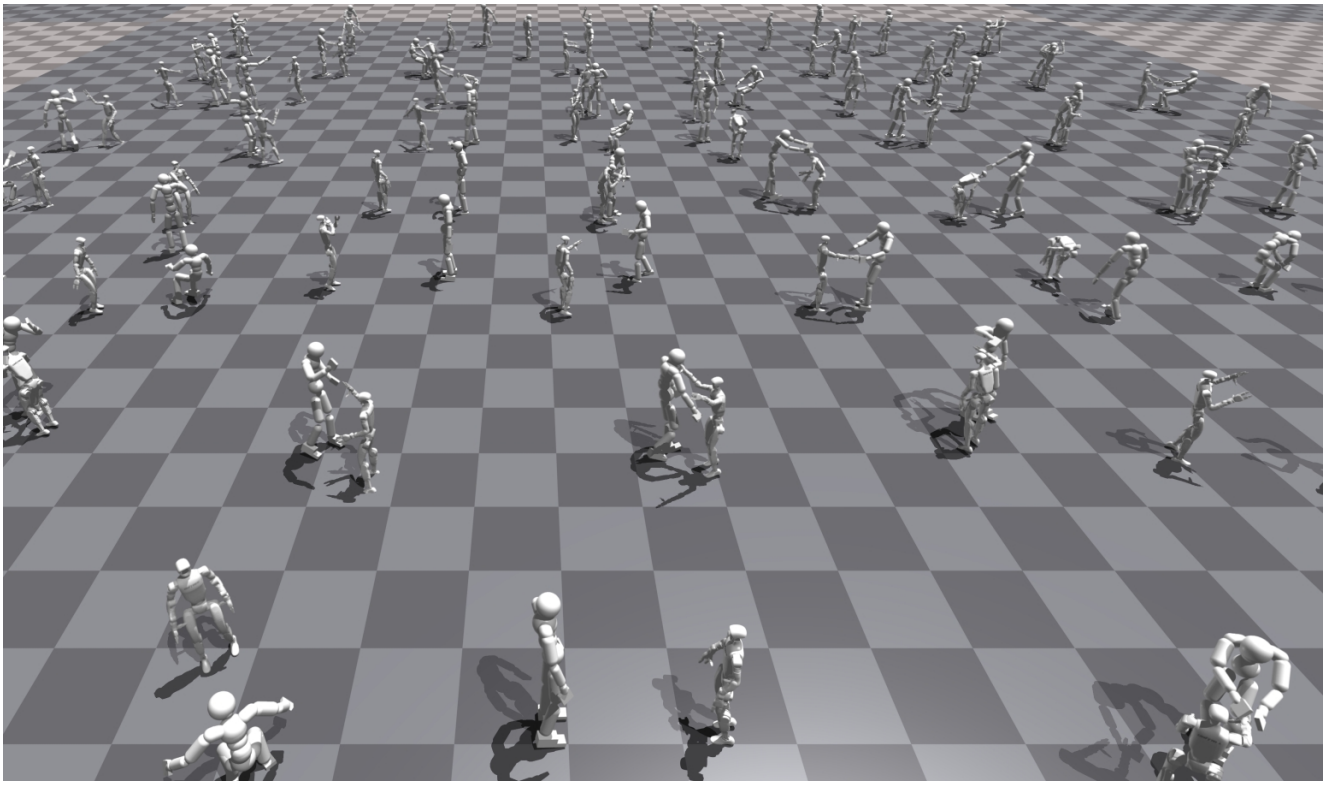

Figure S1. The foundation of our entire framework: a large-scale, high-fidelity Human-Humanoid Interaction (HHoI) dataset that we generated to solve the critical data scarcity problem. Visualized here are thousands of physically-consistent interaction pairs, the output of **PAIR**. The dataset's strategic diversity—spanning contact-rich scenarios (e.g., hugs, handshakes), dynamic gestures (e.g., high-fives), and social cues (e.g., bows)—provides the rich and complex spatio-temporal variations required to train a policy that learns to interact.

average sample durations ranging from 16.1 seconds for compact gestures to 25.9 seconds for complex, contact-rich tasks like hugs. This temporal depth, combined with the tracking of key body joints, provides a rich source of spatio-temporal data essential for training our multi-scale policy. The balanced distribution across tasks ensures robust training and evaluation across all targeted interaction categories.

C. Experimental Protocols, Metrics, and Additional Diagnostics

In this section, we provide a detailed account of the protocols and metrics used to validate our framework at both the data and policy levels. The descriptions are designed to be comprehensive, ensuring full reproducibility of our experimental results and the claims made in the main paper.

Table S5. **Aggregate statistics before and after filtering.** Filtering removes infeasible outliers while preserving or slightly improving contact richness.

Statistic	Before	After	Change
Contact events/s	0.2016	0.2136	+5.95%
Contact ratio	0.2361	0.2519	+6.67%
Mean nearest-joint dist (m)	0.9090	0.8822	-2.95%
AABB overlap IoU	0.0628	0.0655	+4.33%
Joint speed p95 (m/s)	1.6800	1.6764	-0.21%
Joint accel p95 (m/s^2)	16.9952	16.6538	-2.01%
COM outside support AABB	0.5408	0.5315	-1.72%

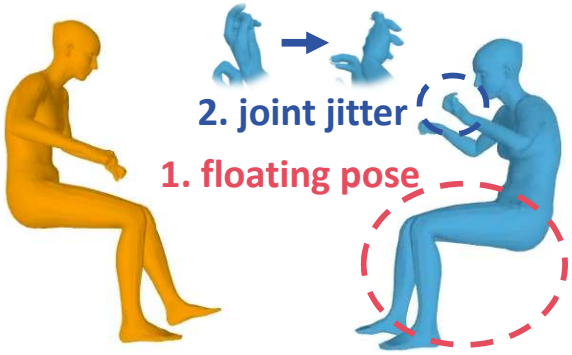

Figure S2. **Representative filtered cases.** The removed samples are infeasible outliers rather than meaningful but difficult interactions.

Table S6. Detailed HHoI Dataset Statistics by Interaction Task. The column abbreviations are as follows: **Sam.**: Samples; **Seq.**: Total sequences of frames; **Dura.**: Total duration in seconds; **Avg Fra.**: Average frames per sample; **Avg Sec.**: Average seconds per sample.

Task	**Sam.**	**Seq.**	**Dura.**	**Avg Fra.**	**Avg Sec.**
Hug	226	292,873	5,857.5	1,296	25.9
Handshake	259	296,414	5,928.3	1,144	22.9
High-Five	219	176,482	3,529.6	805	16.1
Bend	200	176,024	3,520.5	880	17.6
Fly-Kiss	177	158,412	3,168.2	895	17.9
Wave	160	149,118	2,982.4	932	18.6
Total	**1,241**	**1,249,323**	**24,986.5**	**1,007**	**20.1**

C.1. Unified Notation & Units

Sampling and units. All sequences are sampled at $\Delta t = 0.02\,\text{s}$ (50 Hz). Distances are in meters; angles in radians; jerk on positional trajectories is in m/s^3.

Motion variables. For agent $X \in \{H_s, H_p, R\}$ (source human, partner human, robot), a motion sequence is $\mathbf{M}_X = \{\mathbf{q}_t^X\}_{t=1}^T$ with generalized coordinates $\mathbf{q}_t^X \in \mathbb{R}^{D_X}$. Let $\mathcal{J}_X$ be the joint set of X. The 3D position of joint $j \in \mathcal{J}_X$ is $\mathcal{J}_t(X, j) \in \mathbb{R}^3$, obtained by forward kinematics from $\mathbf{q}_t^X$.

Table S7. **Symbol table (core).** In this table, (τ) denotes the **contact distance threshold** only; task-detection thresholds are denoted $(\tau_\cdot)$ and are summarized in Table S11.

Symbol	Meaning
$\mathbf{M}_X = \{\mathbf{q}_t^X\}_{t=1}^T$	Motion sequence of agent X
$\mathcal{J}_X$	Joint set of agent X
$\mathcal{J}_t(X, j) \in \mathbb{R}^3$	3D joint position via FK from $\mathbf{q}_t^X$
Reshaped(H_s)	Morphology-aligned skeleton of H_s
$\mathbf{P}_{\text{task}}$	Task-specific human–robot keypoint pairs
Δt	Sampling period (0.02 s, i.e., 50 Hz)
τ	Contact distance threshold $(0.2/0.35/0.5$ m)
$w_{\text{phase\_KL}}$	Weight for the phase/transition KL term ($\mathcal{L}_{\text{trans}}$)
PAIR	Physics-Aware Interaction Retargeting
D-STAR	Decoupled Spatio-Temporal Action Reasoner

Morphology alignment. Reshaped(H_s) denotes the morphology-aligned human skeleton constructed by pelvis-frame alignment, isotropic bone-length scaling, and a fixed joint correspondence derived from the robot–SMPL mapping. In practice, we obtain the reshaped SMPL parameters by optimizing SMPL shape $\boldsymbol{\beta}$ and a global scale s to fit robot bone lengths (see Sec. A.1); we refer to the resulting kinematic chain as Reshaped(H_s).

Task keypoint pairs. $\mathbf{P}_{\text{task}}$ denotes the task-specific set of human–robot keypoint pairs referenced by contact-related definitions (see Sec. C.2.1).

C.2. Motion Retargeting Evaluation

To rigorously assess the quality of **PAIR**, we employed a comprehensive suite of 18 metrics spanning physical consistency, contact preservation, plausibility, and smoothness.

C.2.1. Evaluation Metrics

Physical Consistency Metrics. These metrics evaluate the overall geometric and structural fidelity of the retargeted motion.

- **Joint Position Error (JPE)** $\downarrow$: The mean L2 distance between robot joints and the morphologically reshaped source human joints Reshaped(H_s) (Sec. A.1), measuring kinematic similarity.
- **Average Workspace Distance (AWD)** $\downarrow$: The mean absolute difference between the full $N \times N$ pairwise distance matrices built from a fixed set of interaction joints (head, shoulders, elbows, wrists). We compare the original human–human interaction (initiator vs. responder) with the retargeted human–robot interaction (initiator vs. robot), capturing holistic spatial-relationship preservation.

Multi-Threshold Contact Detection Metrics. To evaluate contact preservation across varying interaction types, we define three distance thresholds and compute standard classification metrics for each.

- **Thresholds**: 0.2 m, 0.35 m, and 0.5 m.
- **Protocol**: For each frame we classify contact for the two hand–hand pairs using optimal left/right matching; metrics are micro-averaged over frames×hands. Unless otherwise stated, we do not enforce a minimum contact duration.
- **Metrics**: For each threshold, we compute **Precision** ↑, **Recall** ↑, **F1-Score** ↑, and **Accuracy** ↑ to quantify how well the retargeted motion preserves the original contact semantics.

Physical Plausibility Metrics. These metrics assess whether the generated motion is natural and within realistic biomechanical limits.

- **Large Angle Ratio** ↓: The percentage of frames where joint-angle magnitudes (axis–angle) exceed 0.5 rad, flagging unnatural poses.
- **Angle Standard Deviation** ↓: The standard deviation of joint-angle magnitudes (axis–angle) aggregated over joints and frames, indicating pose distribution consistency.

Motion Smoothness Metrics. These metrics quantify the temporal coherence and fluidity of the motion.

- **Jerk Mean** ↓ **(m/s^3)**: The average magnitude of the third temporal derivative of 3D joint positions, computed via third-order finite differences at the dataset frame rate (50 Hz), penalizing high-frequency jitter.
- **Jerk Standard Deviation** ↓ **(m/s^3)**: The standard deviation of the same jerk magnitudes, indicating temporal stability.

C.2.2. Additional Retargeting Diagnostics

PAIR is lightweight and parallelizable in practice: generating the 6.9-hour HHI-to-HHoI dataset used in this work takes 421.26 s with 128 CPU processes (approximately 17 h for 1000 h at the same throughput). Table S8 shows that moderate loss-weight changes produce only small variations in JPE, AWD, contact F1, and Large-angle, indicating that the method is not brittle to hyperparameter choice. Beyond human-human interaction, the same contact-centric formulation can be extended by redefining task contact keypoints; Fig. S3 shows a qualitative transfer example on CORE4D.

C.3. Interaction Policy Evaluation

C.3.1. Task Success Criteria and Detection Algorithms

To quantitatively evaluate policy performance, we designed specific success criteria for each of the six interaction tasks. The core logic for each detection algorithm is outlined below (Algorithms 1–6), providing a transparent basis for our results.

Table S8. **Sensitivity to loss-weight scaling in PAIR.** Moderate changes to the main loss weights produce only small variations in retargeting quality.

Setting	JPE↓	AWD↓	Contact F1↑	Large-angle↓
base	0.1740	0.0950	0.8410	0.0930
$w_{\text{con}} \times 0.5$	0.1739	0.0944	0.8369	0.0930
$w_{\text{con}} \times 2.0$	0.1740	0.0973	0.8392	0.0931
$w_{\text{reg}} \times 0.5$	0.1736	0.0948	0.8377	0.1077
$w_{\text{reg}} \times 2.0$	0.1744	0.0998	0.8377	0.0769
$w_{\text{hum}} \times 0.5$	0.1740	0.0984	0.8392	0.0931
$w_{\text{hum}} \times 2.0$	0.1739	0.0940	0.8349	0.0930
$w_{\text{kin}} \times 0.5$	0.1745	0.0984	0.8384	0.0769
$w_{\text{kin}} \times 2.0$	0.1736	0.0954	0.8392	0.1077

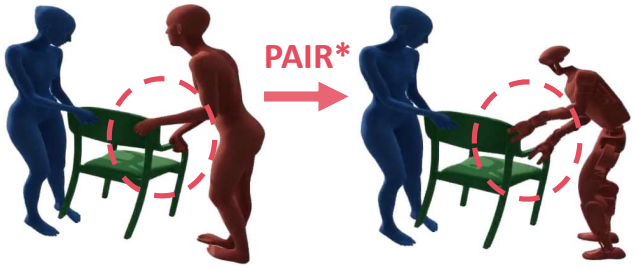

Figure S3. **Qualitative transfer beyond human-human interaction.** By redefining task contact keypoints, the same contact-centric formulation can be applied to human-object interaction data (illustrated here with CORE4D).

Hug

A hug is detected if the human's hands make sustained and appropriate contact with the robot's torso or shoulders, captured via three distinct modes. Algorithm 1 details the full detection logic.

Handshake

A successful handshake is verified through both sustained hand contact and appropriate interpersonal distance. Algorithm 2 provides the handshake detection procedure.

High-Five

Success requires a brief, decisive contact between human and robot hands at an appropriate height. Algorithm 3 outlines the high-five detector.

Wave

A wave is identified by characteristic hand motion patterns, including sufficient amplitude and clear direction changes. Algorithm 4 describes the full wave detection pipeline.

Bend

A bend is robustly measured by the forward inclination angle of the head-root vector relative to the vertical axis. Algorithm 5 formalizes the bend detection criteria.

Fly-Kiss

This gesture is detected via its characteristic two-phase sequence: hand-to-head proximity followed by a forward projection motion. Algorithm 6 summarizes the fly-kiss detector.

C.3.2. Hyperparameter Justification

The hyperparameters for these detection algorithms, detailed in Table S11, were empirically determined by tuning on a randomly sampled validation subset of 50 annotated interaction sequences drawn from the full dataset. The values were selected to maximize the alignment between our automated metrics and the consensus of three human evaluators, ensuring a robust and meaningful evaluation that reflects human judgment of interaction success.

C.3.3. Additional Policy Robustness and Deployment Diagnostics

Perception robustness and waist-locking effects. To assess robustness to noisy monocular perception, we inject test-time corruption into the human SMPL observations using joint-position jitter (1/2/5 cm) and random dropout ($p = 0.05/0.1/0.2$). Table S9 reports the resulting success rates together with an additional comparison between waist-free and waist-locked execution. Performance remains stable under substantial corruption, and the waist-locking ablation shows only a modest change in overall success.

Online reactivity counterfactuals. All real-robot results are generated online from perception-conditioned commands; no pre-recorded trajectories are used. To illustrate partner-conditioned behavior, we perform two counterfactual tests. First, we vary the spatial target location of the handshake and record the corresponding success rates (Table S10). Second, we vary the timing of the human hand raise; Fig. S4 shows that robot initiation shifts accordingly. These counterfactuals support that the deployed system is reactive rather than motion replay.

Table S10. **Spatial counterfactual for online handshake reactivity.** Success rates under left/right and high/low target variations.

Target variation	Left	Right
High	80%	100%
Low	40%	60%

D. Dataset Retargeting for Sim-to-Real

Protocol. Before deployment, we construct the adaptation set through a single, task-agnostic retargeting pass. We keep *Act* unchanged. *Preparation* is standardized to a neutral upper-body posture (both hands naturally placed in front of the root on the left/right sides), with identical lower-body motion; the root is retargeted to milder yaw and simpler walking. The last 16 frames bilinearly transition into the first frame of the *Act*. *Follow-up* becomes stationary while the upper body bilinearly returns over 16 frames to the same neutral posture. This retargeted dataset is used once to fine-tune the high-level policy for real-robot execution; no per-task tuning or prompt-specific calibration is applied.

Pseudo-code. Algorithm 7 summarizes the retargeting routine.

E. Observation & Action Specification

Observation and action specification. We use human root translation (3D) and SMPL joint positions (72D) as the human observation channels, combined with robot proprioception. The reference action is 36-D (29-DoF joint targets + root translation/orientation). Table S1 lists the exact channels and dimensionalities.

F. Temporal Scheduler: Densify and Fuse (within D-STAR)

Diffusion details. The diffusion-based planning head is trained with a DDIM scheduler [5] using 50 training time steps and 10 inference steps with a squared-cosine schedule. We use group normalization throughout the denoiser and adopt standard sinusoidal timestep embeddings.

Runtime pipeline. The policy emits 5 anchors per invocation, covering 0–2.0 s with a 0.5 s step; the policy runs at 5 Hz. Anchors are densified to 50 Hz and bilinearly fused across overlapping calls to ensure temporal continuity; zero-order hold is applied if updates are delayed and root-velocity bounds are temporarily tightened.

Densification (per call at t_k). Algorithm 8 interpolates each 5-anchor policy output to a 50 Hz reference trajectory.

Cross-call fusion (between t_k and t_{k+1}, 5 Hz). Algorithm 9 bilinearly fuses overlapping dense segments to ensure continuity.

Table S9. **Policy success under perception corruption and waist-locking settings.** We report overall success together with per-task success rates.

Setting	Overall	Hug	Handshake	High-Five	Wave	Bend	Fly-Kiss
Ours (clean)	84.9%	93.3%	77.4%	70.4%	96.3%	93.3%	90.9%
Jitter 1cm	80.2%	93.3%	64.5%	63.0%	96.3%	93.3%	90.9%
Jitter 2cm	79.4%	100.0%	61.3%	59.3%	96.3%	93.3%	90.9%
Jitter 5cm	77.8%	100.0%	58.1%	55.6%	96.3%	93.3%	90.9%
Dropout $p = 0.05$	78.6%	100.0%	61.3%	59.3%	92.6%	93.3%	90.9%
Dropout $p = 0.10$	78.6%	93.3%	67.7%	51.9%	96.3%	93.3%	90.9%
Dropout $p = 0.20$	75.4%	93.3%	58.1%	48.1%	96.3%	93.3%	90.9%
Ours (HOMIE, waist free)	67.8%	66.8%	46.7%	33.8%	93.8%	96.0%	86.4%
Ours (HOMIE, waist locked)	66.1%	56.2%	47.5%	32.9%	93.8%	97.5%	86.4%

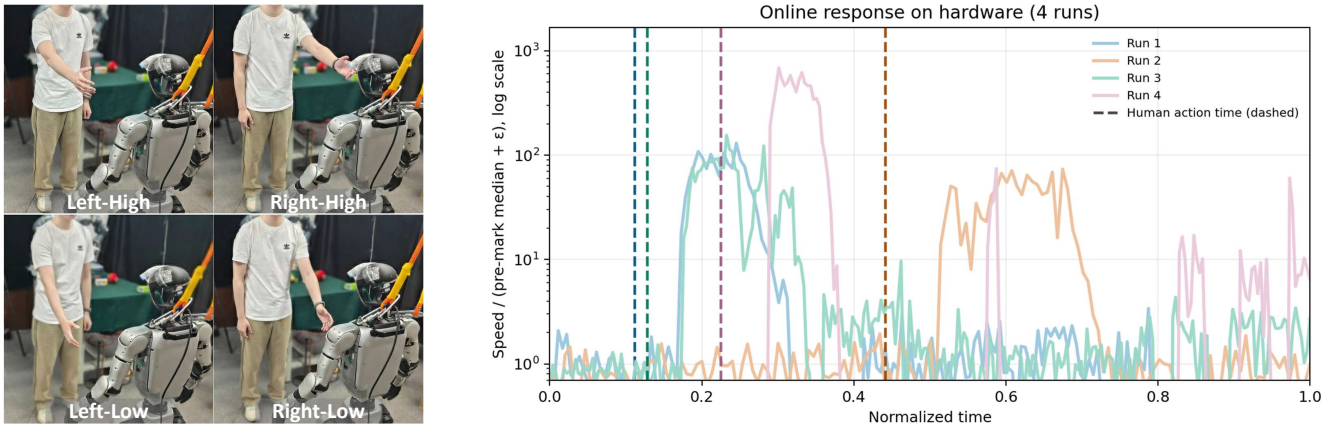

Figure S4. **Counterfactual online reactivity tests on hardware.** Left: handshake behavior under spatial target variations. Right: earlier or later human hand raises shift the timing of robot initiation.

Algorithm 1 Hug Detection with Multi-Modal Embrace Recognition

Require: Human joints $\mathbf{J}_H$, robot torso position $\mathbf{p}_R$
Ensure: Success flag $success_{hug}$

Extract hand positions: $\mathbf{h}_L, \mathbf{h}_R \leftarrow \mathbf{J}_H[joints_{hand}]$
$double\_embrace \leftarrow \text{DetectDoubleEmbrace}(\mathbf{h}_L, \mathbf{h}_R, \mathbf{p}_R)$
$single\_embrace \leftarrow \text{DetectSingleEmbrace}(\mathbf{h}_L, \mathbf{h}_R, \mathbf{p}_R)$
$shoulder\_embrace \leftarrow \text{DetectShoulderEmbrace}(\mathbf{h}_L, \mathbf{h}_R, \mathbf{p}_R)$
return $double\_embrace \vee single\_embrace \vee shoulder\_embrace$

procedure DETECTDOUBLEEMBRACE($\mathbf{h}_L, \mathbf{h}_R, \mathbf{p}_R$)
    $d_{hands} \leftarrow \|\mathbf{h}_L - \mathbf{h}_R\|_2$ ▷ Distance between hands
    $d_{L2robot} \leftarrow \|\mathbf{h}_L - \mathbf{p}_R\|_2, d_{R2robot} \leftarrow \|\mathbf{h}_R - \mathbf{p}_R\|_2$
    $hands\_close \leftarrow d_{hands} < \tau_{hand\_dist}$
    $hands\_near\_robot \leftarrow (d_{L2robot} < \tau_{body}) \wedge (d_{R2robot} < \tau_{body})$
    $embrace\_frames \leftarrow hands\_close \wedge hands\_near\_robot$
    return MaxConsecutiveFrames$(embrace_frames) \geq \tau_{min_frames}$
end procedure

Table S11. Detection Algorithm Hyperparameters

Task	Parameter	Value	Description
Hug	τ_{hand_dist}	0.5m	Maximum distance between hands for double embrace
	τ_{body}	0.45m	Hand to robot torso threshold
	$\tau_{shoulder}$	0.4m	Hand to shoulder distance for shoulder embrace
	τ_{min_frames}	3	Minimum consecutive frames for a successful embrace
High-Five	$\tau_{contact}$	0.4m	Maximum hand-to-hand distance for contact
	$\tau_{min\_approach}$	1	Minimum frames of valid contact approach
	$\tau_{max\_sustained}$	1000	Maximum frames of sustained contact (to rule out holding)
	τ_{height}	0.3m	Minimum hand height above human's root
Handshake	$\tau_{contact}$	0.3m	Maximum hand-to-hand contact distance
	$\tau_{min\_contact}$	10	Minimum consecutive frames of sustained contact
	τ_{min_dist}	0.4m	Minimum distance between agent roots
	τ_{max_dist}	1.5m	Maximum distance between agent roots
	$\tau_{std\_contact}$	0.15m	Upper bound of contact-distance standard deviation (stability)
	$\tau_{mean\_contact}$	0.45m	Upper bound of contact-distance mean (stability)
Wave	τ_{motion_dist}	0.5m	Minimum total distance traveled by the waving hand
	$\tau_{min\_changes}$	3	Minimum number of significant direction changes
	$\tau_{amplitude}$	0.3m	Minimum motion amplitude (peak-to-peak)
	τ_{angle}	0.785 rad	Angle threshold for detecting a direction change ($\pi/4$)
	τ_{height}	0.45m	Minimum hand height above human's root
Bend	τ_{min_angle}	0.349 rad	Minimum bend angle from vertical ($\pi/6$)
	τ_{min_frames}	5	Minimum consecutive frames angle is above threshold
Fly-Kiss	$\tau_{hand2head}$	0.3m	Maximum distance from hand to head for Phase 1
	$\tau_{forward\_thresh}$	0.1m	Minimum total forward motion projected towards partner
	$\tau_{min\_forward}$	5	Minimum consecutive frames of forward hand motion in Phase 2

Algorithm 2 Handshake Detection with Spatial Validation

Require: Human hands $\mathbf{h}_{L,H}, \mathbf{h}_{R,H}$, robot hands $\mathbf{h}_{L,R}, \mathbf{h}_{R,R}$, human root $\mathbf{r}_H$
Ensure: Success flag $success_{handshake}$
$d_{min} \leftarrow$ ComputeMinHandDist($\mathbf{h}_{\text{L,H}}, \mathbf{h}_{\text{R,H}}, \mathbf{h}_{\text{L,R}}, \mathbf{h}_{\text{R,R}}$)
$contact\_phase \leftarrow$ DetectContactPhase($d_{min}, \mathbf{r}_H$)
$stability\_phase \leftarrow$ DetectStabilityPhase($d_{min}, \mathbf{r}_H$)
return $contact\_phase \wedge stability\_phase$
procedure DETECTCONTACTPHASE($d_{min}, \mathbf{r}_H$)
    $contact\_frames \leftarrow d_{min} < \tau_{contact}$
    Validate each contact frame with root distance constraint:
    for $t \in contact\_frames$ **do**
        $\mathbf{h}_{active} \leftarrow$ GetActiveHand(t) ▷ Hand with min dist
        $d_{root} \leftarrow \|\mathbf{h}_{active} - \mathbf{r}_H[t]\|_2$
        $contact\_frames[t] \leftarrow (d_{root} \geq \tau_{min\_dist}) \wedge (d_{root} \leq \tau_{max\_dist})$
    end for
    $valid\_contact\_count \leftarrow \sum contact\_frames$
    $max\_consecutive \leftarrow$ MaxConsecutiveFrames($contact\_frames$)
    return $(valid\_contact\_count \geq \tau_{min\_contact}) \wedge (max\_consecutive \geq \tau_{min\_contact})$
end procedure
procedure DETECTSTABILITYPHASE($d_{min}, \mathbf{r}_H$)
    $contact\_frames \leftarrow d_{min} < \tau_{contact}$
    $d\_seq \leftarrow d_{min}[contact\_frames]$
    $stability\_std \leftarrow$ RollingStd(d_seq)
    $stability\_mean \leftarrow$ RollingMean(d_seq)
    return $(\text{Std}(d\_seq) < \tau_{std\_contact}) \wedge (\text{Mean}(d\_seq) < \tau_{mean\_contact})$
end procedure

Algorithm 3 High-Five Detection with Height Validation

Require: Human hands $\mathbf{h}_{L,H}, \mathbf{h}_{R,H}$, robot hands $\mathbf{h}_{L,R}, \mathbf{h}_{R,R}$, human root $\mathbf{r}_H$
Ensure: Success flag $success_{highfive}$

Compute all hand-to-hand distances:
$d_{LL} \leftarrow \|\mathbf{h}_{L,H} - \mathbf{h}_{L,R}\|_2$, $d_{LR} \leftarrow \|\mathbf{h}_{L,H} - \mathbf{h}_{R,R}\|_2$
$d_{RL} \leftarrow \|\mathbf{h}_{R,H} - \mathbf{h}_{L,R}\|_2$, $d_{RR} \leftarrow \|\mathbf{h}_{R,H} - \mathbf{h}_{R,R}\|_2$
$d_{min} \leftarrow \min(d_{LL}, d_{LR}, d_{RL}, d_{RR})$
$close\_frames \leftarrow d_{min} < \tau_{contact}$
if height_validation_enabled **then**
    $valid\_frames \leftarrow$ ValHandHeight($close\_frames, \mathbf{h}_{\text{L,H}}, \mathbf{h}_{\text{R,H}}, \mathbf{h}_{\text{L,R}}, \mathbf{h}_{\text{R,R}}, \mathbf{r}_{\text{H}}$)
    $close\_frames \leftarrow close\_frames \wedge valid\_frames$
end if
$contact\_count \leftarrow \sum close\_frames$
$consecutive\_count \leftarrow$ MaxConsecutiveFrames($close\_frames$)
return $(contact\_count \geq \tau_{min\_approach}) \wedge (consecutive\_count \leq \tau_{max\_sustained})$

Algorithm 4 Wave Detection with Motion Pattern Analysis

Require: Human hands $\mathbf{h}_{L,H}, \mathbf{h}_{R,H}$, human root $\mathbf{r}_H$
Ensure: Success flag $success_{wave}$

Select active hand based on motion intensity.
$motion_L \leftarrow$ CalculateMotionIntensity($\mathbf{h}_{L,H}$)
$motion_R \leftarrow$ CalculateMotionIntensity($\mathbf{h}_{R,H}$)
$\mathbf{h}_{active} \leftarrow \mathbf{h}_{L,H}$ if $motion_L.total > motion_R.total$ else $\mathbf{h}_{R,H}$
$motion\_sufficient \leftarrow motion.total > \tau_{motion\_dist}$
$direction\_changes \leftarrow$ DetectDirectionChanges($\mathbf{h}_{active}$)
$amplitude\_check \leftarrow$ DetectAmplitude($\mathbf{h}_{active}$)
$height\_check \leftarrow$ DetectHandHeight($\mathbf{h}_{L,H}, \mathbf{h}_{R,H}, \mathbf{r}_H$)
return $motion\_sufficient \wedge direction\_changes \wedge amplitude\_check \wedge height\_check$

procedure DETECTDIRECTIONCHANGES($\mathbf{h}_{active}$)
    Compute position changes: $\Delta\mathbf{p} \leftarrow$ diff($\mathbf{h}_{active}$)
    Compute motion angles: $\theta \leftarrow \arctan 2(\Delta\mathbf{p}[:,1], \Delta\mathbf{p}[:,0])$
    $direction\_changes \leftarrow 0$
    for $i = 1$ to len(θ) $- 1$ **do**
        $\Delta\theta \leftarrow |\theta[i] - \theta[i-1]|$
        if $\Delta\theta > \pi$ **then** $\Delta\theta \leftarrow 2\pi - \Delta\theta$
        end if
        if $\Delta\theta > \tau_{angle}$ and CheckDisplacement($\mathbf{h}_{active}, i$) **then**
            $direction\_changes \leftarrow direction\_changes + 1$
        end if
    end for
    return $direction\_changes \geq \tau_{min\_changes}$
end procedure

Algorithm 5 Bend Detection via Head-Root Angular Analysis

Require: Human head $\mathbf{h}_{head}$, human root $\mathbf{r}_H$
Ensure: Success flag $success_{bend}$
Compute head-root vectors: $\mathbf{v}_{hr} \leftarrow \mathbf{h}_{head} - \mathbf{r}_H$
Define vertical axis: $\mathbf{z}_{axis} \leftarrow [0, 0, 1]$
Compute bend angles: $\theta_{bend} \leftarrow \text{VectorAngle}(\mathbf{v}_{hr}, \mathbf{z}_{axis})$
$max\_angle \leftarrow \max(\theta_{bend})$
$frames\_above\_threshold \leftarrow \sum(\theta_{bend} \geq \tau_{min\_angle})$
$angle\_condition \leftarrow max\_angle \geq \tau_{min\_angle}$
$stability\_condition \leftarrow frames\_above\_threshold \geq \tau_{min\_frames}$
return $angle\_condition \wedge stability\_condition$

Algorithm 6 Fly-Kiss Detection with Motion Projection

Require: Human hands $\mathbf{h}_{L,H}, \mathbf{h}_{R,H}$, head $\mathbf{h}_{head}$, roots $\mathbf{r}_H, \mathbf{r}_R$
Ensure: Success flag $success_{flyingkiss}$
Select active hand closest to head.
$d_{L2head} \leftarrow \min(\|\mathbf{h}_{L,H} - \mathbf{h}_{head}\|_2)$
$d_{R2head} \leftarrow \min(\|\mathbf{h}_{R,H} - \mathbf{h}_{head}\|_2)$
$\mathbf{h}_{active}, d_{min} \leftarrow (\mathbf{h}_{L,H}, d_{L2head})$ if $d_{L2head} \leq d_{R2head}$ else $(\mathbf{h}_{R,H}, d_{R2head})$
$condition\_A \leftarrow d_{min} \leq \tau_{hand2head}$
Compute forward direction: $\mathbf{dir} \leftarrow \text{normalize}(\text{mean}(\mathbf{r}_R) - \text{mean}(\mathbf{r}_H))$
Compute hand motion: $\Delta\mathbf{h} \leftarrow \text{diff}(\mathbf{h}_{active})$
Forward projections: $proj \leftarrow \Delta\mathbf{h} \cdot \mathbf{dir}$
$forward\_frames \leftarrow proj > 0$
$consecutive\_forward \leftarrow \text{MaxConsecutiveFrames}(forward\_frames)$
$total\_forward\_motion \leftarrow \sum(proj[proj > 0])$
$condition\_B \leftarrow (consecutive\_forward \geq \tau_{min\_forward}) \wedge (total\_forward\_motion \geq \tau_{forward\_thresh})$
return $condition\_A \wedge condition\_B$

Algorithm 7 Retarget(Preparation, Act, Follow-up)

Prep ← neutralize_upper_body(Preparation)
Prep.root ← retarget_root(Preparation.root, mild_yaw, simple_walk)
Act ← Act
Follow ← make_stationary(Follow_up)
Follow.upper ← return_to_neutral(Follow_up.upper)
// 16-frame bilinear ramps
Prep$[-16:]$ ← bilinear_ramp(Prep$[-16:]$, Act[0])
Follow$[:16]$ ← bilinear_ramp(Act$[-1]$, Follow$[:16]$) **return** concatenation(Prep, Act, Follow)

Algorithm 8 DensifyAndTrack(5 anchors → 100-frame reference)

anchors $\{(t_k + 0.0, \mathbf{q}_0), (t_k + 0.5, \mathbf{q}_{0.5}), \ldots, (t_k + 2.0, \mathbf{q}_{2.0})\}$
grid $\mathcal{T} = \{t_k + i/50 \mid i = 0, \ldots, 99\}$
for $t \in \mathcal{T}$ **do**
    $(\tau_1, \tau_2) \leftarrow$ nearest anchor times s.t. $\tau_1 \leq t \leq \tau_2$
    $\hat{\mathbf{q}}(t) \leftarrow$ **bilinear_interp** in (time × joint/root) between $(\tau_1, \mathbf{q}_{\tau_1}), (\tau_2, \mathbf{q}_{\tau_2})$
end forreturn $\{\hat{\mathbf{q}}(t)\}_{t \in \mathcal{T}}$

Algorithm 9 FuseOverlaps($\hat{\mathbf{q}}_k$, $\hat{\mathbf{q}}_{k+1}$)

let $\mathcal{T}_{\text{overlap}} = \mathcal{T}_k \cap \mathcal{T}_{k+1}$ on the 50 Hz grid
for $t \in \mathcal{T}_{\text{overlap}}$ **do**
    $\tilde{\mathbf{q}}(t) \leftarrow$ **bilinear_interp_time**$(\hat{\mathbf{q}}_k(t), \hat{\mathbf{q}}_{k+1}(t), t)$
end for
if timeout: $\tilde{\mathbf{q}}(t) \leftarrow$ hold-last and tighten root-velocity bounds **return** fused dense segment $\tilde{\mathbf{q}}(t)$

References

[1] Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. HOMIE: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025. 6

[2] Zixuan Chen, Mazeyu Ji, Xuxin Cheng, Xuanbin Peng, Xue Bin Peng, and Xiaolong Wang. GMT: General motion tracking for humanoid whole-body control. *arXiv preprint arXiv:2506.14770*, 2025. 6

[3] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14737–14748, 2023. 6

[4] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248:1–248:16, 2015. 1

[5] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 11

[6] Liang Xu, Xintao Lv, Yichao Yan, Xin Jin, Shuwen Wu, Congsheng Xu, Yifan Liu, Yizhou Zhou, Fengyun Rao, Xingdong Sheng, et al. Inter-X: Towards versatile human-human interaction analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22260–22271, 2024. 1, 7