

# RedirectMimic: Local Motion Content Redaction for Humanoid Whole-Body Control

Zepeng Wang and Xiaochuan Shi<sup>†</sup>

Wuhan University

<sup>†</sup>Corresponding author.

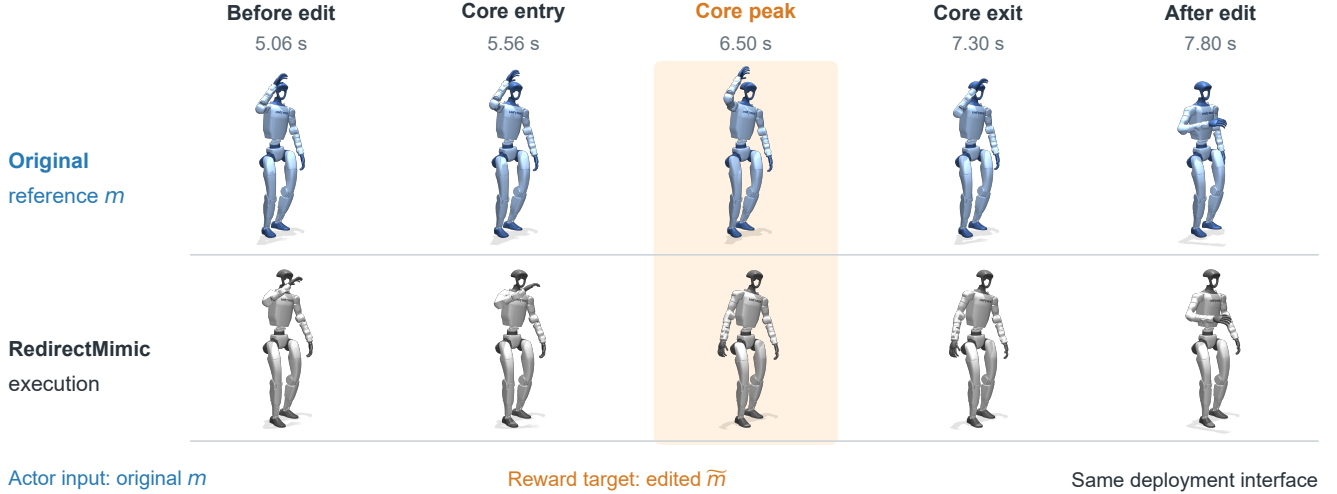


Fig. 1. Local redaction in a selected G1 simulation rollout. Top: original reference  $m$ . Bottom: recorded RedirectMimic execution at matching times. The pre-edit column shows ordinary hand-to-mouth motion. At core entry, execution lags the edited target.

**Abstract**—Humanoid motion interfaces should accommodate application-specific movement constraints while retaining useful reference behavior. Local content redaction requires a controller to replace selected movements even when it continues to receive the original reference. Removing the tracking incentive for those movements leaves their replacement unspecified. We introduce RedirectMimic, which separates the reference observed by the actor from the target used to train its execution. A rule-generated edited reference guides tracking rewards during proximal policy optimization, while the deployed observation and action interfaces remain unchanged. We evaluate pose- and speed-defined arm editing in Unitree G1 simulation using a tracker pretrained on 7,509 motion clips. On the overhead-arm task, RedirectMimic reduces core-window content occurrence to 2.7%, compared with 58.9% for reward zeroing and 32.8% for anti-tracking. It achieves 94.2% core-window joint success, combining content suppression, fall-free execution, and local retention. Across both tasks, it achieves the lowest edited-arm error among the matched local methods. Trajectory analysis shows why replacement fidelity adds information beyond binary content suppression. These results establish target-directed local policy editing as a way to change selected behavior while preserving the original control interface. Project page: <https://xiaopeng-whu.github.io/RedirectMimic/>.

## I. INTRODUCTION

Motion tracking provides a practical interface between human demonstrations and humanoid control, including whole-body teleoperation and data collection [1], [2]. Physics-based

imitation methods learn diverse motor skills from reference trajectories [3], [4], [5], and large motion collections broaden the behaviors available to a controller [6]. Faithful reproduction, however, is not always the desired deployment behavior. A reference may contain an arm movement that a particular application excludes, embedded within otherwise acceptable locomotion or gestures. The objective is then selective: replace the specified movement while retaining the rest of the motion.

Figure 1 illustrates this local objective. The controller must execute a suitable replacement while maintaining balance and the surrounding movement. Arm motion and balance are dynamically coupled: a local joint edit can affect torso and leg execution even when their reference coordinates remain unchanged. Useful redaction therefore combines content suppression, replacement fidelity, and physical retention.

Existing work establishes several routes to modifying learned behavior. ForgetMimic suppresses designated motions in humanoid policies while retaining other motions [7]. Human Motion Unlearning and SafeMo modify motion generators, including selective removal within composite animations [8], [9]. Behavior Uncloning distills mode redirection into policy weights [10]. Local humanoid editing brings these ideas to a reference-conditioned control problem: how can a tracker learn a chosen replacement while its input continues

to request the original movement?

We address this question with RedirectMimic. An offline rule constructs a paired reference  $(m, \tilde{m})$ , preserving the root trajectory and all unedited joint coordinates. During fine-tuning, the actor continues to observe  $m$  and the robot state, while tracking rewards use  $\tilde{m}$ . Ordinary proximal policy optimization (PPO) [11] updates the existing actor and critic. An edit-window start distribution makes the sparse training objective accessible without supplying the edit mask to the policy. At deployment, the actor retains the original observation and action interfaces.

The central distinction is between suppressing a movement and specifying its replacement. Reward zeroing removes the incentive to track the selected arm, whereas anti-tracking rewards deviation from it. Neither objective specifies where the arm should go. For the tested overhead rule, RedirectMimic reaches 94.2% core-window joint success, compared with 40.8% for reward zeroing and 65.0% for anti-tracking. The speed-defined task complements this result by separating threshold suppression from replacement fidelity: local methods suppress the event similarly, while explicit targets improve agreement with the edited motion.

Our contributions are threefold:

- **Local redaction under original-reference inputs.** We formulate selective motion replacement as policy editing, requiring the controller to retain surrounding motion while its input continues to specify the original behavior.
- **Paired-reference policy fine-tuning.** We introduce RedirectMimic, which trains against rule-generated replacement targets while keeping actor observations tied to the original reference. Edit-window sampling exposes sparse replacements without adding deployment inputs.
- **Evidence for explicit replacement targets.** Matched comparisons in G1 simulation show the highest overhead-arm joint success and lowest replacement error across both tasks among local methods. Trajectory analysis distinguishes faithful replacement from threshold suppression.

## II. RELATED WORK

### A. Humanoid Tracking and Physical Feasibility

DeepMimic establishes reinforcement-learning imitation of reference skills [3], while AMP learns motion-style rewards from demonstrations [12]. Perpetual Humanoid Control extends imitation and recovery across large motion libraries [13]. Recent robot systems pursue expressive upper-body imitation with locomotion in ExBody [14], and more general whole-body tracking in ExBody2 [15]. GMT combines adaptive sampling with motion-specialized experts [16]. SONIC studies scaling motion data, model capacity, and compute [17]. HOVER distills multiple control modes into one policy [18], while TWIST combines reinforcement learning and behavior cloning for coordinated imitation [19]. A recent empirical study examines design choices for general motion tracking [20]. BeyondMimic

provides the tracking formulation underlying our mjlabs implementation [4], [21]. Our study changes the desired local execution of an existing tracker rather than enlarging its motion repertoire.

Reference fidelity can conflict with physical feasibility. ASAP adapts policies to simulation–reality discrepancies using a learned residual action model [22]. AMO combines trajectory optimization and reinforcement learning for adaptive control [23], while OmniTrack generates physically consistent references before training a general tracker [24]. PGMT relaxes tracking to accommodate terrain [25]. These approaches motivate validating a replacement dynamically. Here the reason to deviate is a fixed content rule, and the actor must continue observing the unedited reference.

### B. Motion Completion and Local Editing

MoMask supports text-conditioned temporal inpainting through masked motion modeling [26]. CondMDI supports sparse and partial keyframe constraints [27], while Dart-Control generates motion autoregressively with text and spatial control [28]. MaskedMimic formulates physically simulated character control as motion inpainting from partial descriptions [29]. CLoSD closes the loop between a diffusion planner and a tracking controller [30]. Our input–target split brings local replacement into policy fine-tuning: the actor executes the edited behavior while receiving the original reference, without an inference-time edit mask or replacement trajectory.

### C. Behavior Suppression and Policy Redirection

ForgetMimic suppresses designated humanoid motions while retaining other motions [7]. Local redaction additionally requires preservation within the targeted motion. Behavior Uncloning encodes redirection in policy weights without inference-time steering [10]. We study a paired-reference control objective that directs selected body parts toward a replacement while preserving the original-reference interface.

Human Motion Unlearning suppresses unwanted generated motions through latent code replacement, including edits within composite behaviors [8]. SafeMo studies continuous motion unlearning with refined targets [9]. Building on target-based behavior modification, our experiments isolate explicit replacement from local reward suppression in humanoid control. We also evaluate direct edited-reference tracking as RefEdit<sup>†</sup>, separating deployment with an edited reference from deployment through the original input interface.

## III. LOCAL REDACTION WITH PAIRED REFERENCES

Figure 2 summarizes the three stages: offline pair construction, dual-reference fine-tuning, and deployment through the original policy interface.

### A. Problem and Information Constraint

Let  $m = \{q_t, \dot{q}_t, p_t, R_t, \dots\}_{t=1}^T$  denote a reference motion and  $\pi_0$  a pretrained tracker. A fixed content rule identifies

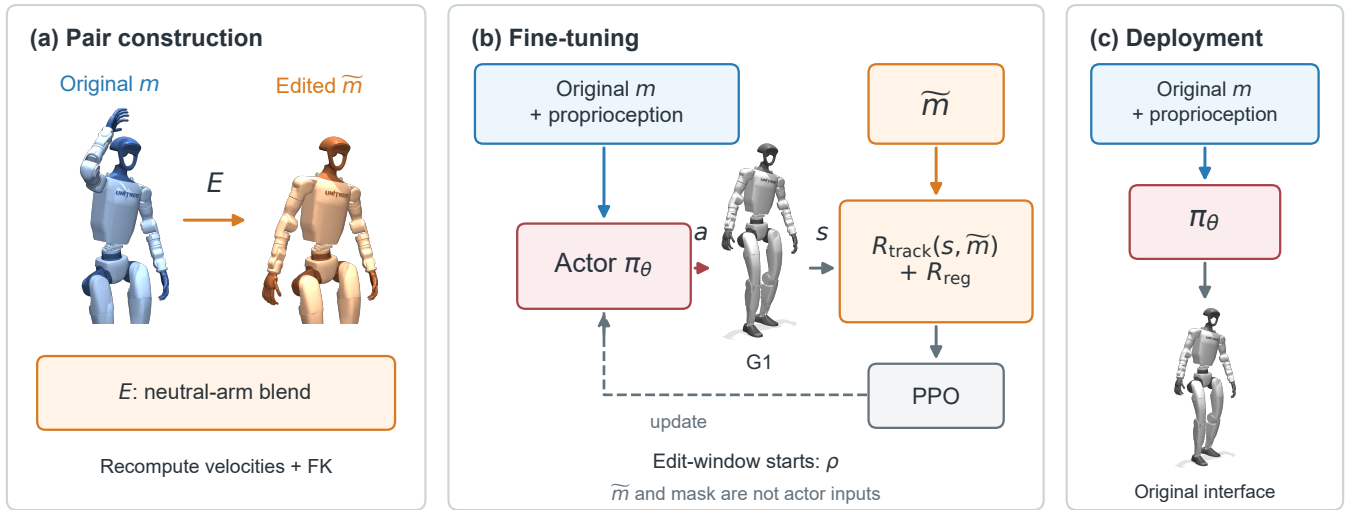


Fig. 2. RedirectMimic overview. (a) Construct the local replacement. (b) Fine-tune with original-reference observations and edited-target rewards. (c) Retain the deployment interface. G1 images show reference poses and recorded simulation execution.

arm-specific intervals, and an editor constructs  $\tilde{m} = E(m)$ . The spatiotemporal edit support  $B$  includes the selected joints, temporal transitions, and neighboring frames affected by differentiation. Normal clips satisfy  $\tilde{m} = m$ . The policy objective is to execute the replacement while preserving other body parts, other times, and normal motions.

The actor receives the original reference and the actual robot state:

$$o_t = \text{Obs}(s_t, m_t), \quad a_t \sim \pi_\theta(\cdot | o_t). \quad (1)$$

The deployed actor is a feedforward network with current-frame reference inputs and the previous action. It receives no future reference frames, clip identifier, edit mask, or edited-reference error. All reference-derived observations and control paths use  $m$ , and the content rule is fixed throughout training and deployment.

The offline editor uses the complete clip to place smooth transitions around each event. The actor learns the resulting replacement from its deployed observations, without access to the future frames used by the editor.

### B. Constructing Consistent Local Targets

For each arm, the speed rule uses the wrist position relative to the shoulder,  $r_t$ , and their relative velocity,  $v_t$ , in torso coordinates. Its signal is

$$g_t^{\text{speed}} = \frac{\langle v_t, r_t \rangle}{\|r_t\|}, \quad g_t^{\text{pose}} = z_t^{\text{wrist}} - z_t^{\text{shoulder}}. \quad (2)$$

The speed and pose rules flag outward speed above 1.5m/s and world-height difference above 0.2m, respectively. Gaps up to 0.6s and 0.3s, respectively, are merged during event construction. Events shorter than 0.06s are discarded. These kinematic rules define application-specific editing requests.

Each event is expanded by 0.3s on both sides to define a core interval  $C$ . An additional 0.4s transition on each

side defines the extended interval  $I$ . The seven joints of the content arm are blended toward the G1 home pose:

$$\tilde{q}_{t,j} = (1 - w_{t,j})q_{t,j} + w_{t,j}q_j^{\text{home}}. \quad (3)$$

Here  $w = 1$  in the core,  $w = 0$  outside the extended interval, and each transition uses the smoothstep polynomial  $3u^2 - 2u^3$ . Nearby intervals are merged to avoid repeatedly returning to the original arm pose. The root, waist, legs, and unselected arm remain unchanged in joint-reference space.

We recompute joint velocities and forward kinematics after editing, including body linear and angular velocities. Both references share timestamps and root alignment. The edit support includes frames affected by differentiation and bodies descending from the edited joints. Paired targets satisfy the content rule, while normal clips retain their original references.

### C. Training Against the Edited Reference

The fine-tuning reward is

$$r_t = \Delta t [R_{\text{track}}(s_t, \tilde{m}_t) + R_{\text{reg}}(s_t, a_t)]. \quad (4)$$

The six tracking terms cover anchor position and orientation, relative body position and orientation, and body linear and angular velocities. All use the edited target with the same coordinate alignment as the original reference. Action-rate, joint-limit, and self-collision regularizers retain their original definitions and weights.

PPO updates both actor and critic using the paired-reference reward. Observation normalizers continue updating across all fine-tuned methods. Local methods share anchor- and ankle-based tracking termination, allowing the arm to depart from its original reference. Pretrained and Vanilla-FT retain wrist tracking checks during training. Fixed-window evaluation removes reference-error termination for every method.

#### D. Sampling Sparse Edit Intervals

Adaptive motion sampling is established in general tracking [16]. Here, failure-driven sampling need not emphasize intervals where following  $m$  remains physically feasible but produces the wrong content. We therefore mix the base start distribution with an edit pool. The pool contains frames where the paired arm references differ and a 1 s lead-in, clipped at motion boundaries. If its fraction of the concatenated time axis is  $p_{\text{pool}}$ , the mixture weight is

$$\rho = \min(\rho_{\max}, \kappa p_{\text{pool}}), \quad \rho_{\max} = 0.3, \quad \kappa = 8. \quad (5)$$

Edit-pool starts are sampled uniformly, while remaining starts use the base adaptive sampler. All episodes initialize from  $m$ , including those that begin inside an edit interval. Pool membership is not an observation. Edit-pool episodes do not update the base sampler, keeping the two sampling components separate.

The base sampler weights starts by failure rate with a normalized 10% uniform mixture.

Algorithm 1 summarizes the complete procedure. Pair construction defines the desired response, while start sampling controls how often the policy encounters it. During a rollout, both references advance on the same motion timeline. The actor follows the original-reference observation path, and the reward evaluates the edited target at the corresponding frame. This separation makes the local replacement part of the learned policy without changing the deployment command.

Here  $\phi$  denotes critic parameters,  $K$  the update count, and  $H$  the rollout length.  $E$  includes blending and kinematic recomputation, and  $\mathcal{U}$  denotes uniform sampling. Step evaluates Eq. (4), returns termination flag  $d_t$ , and maintains the base sampler using base-sampled episodes only. The loop runs in parallel environments initialized with  $\text{reset} = 1$ . Generalized advantage estimation (GAE) supplies advantages  $\hat{A}$  and return targets  $\hat{V}$ .

#### E. Local Reward Controls and Oracle Reference

We construct two local reward controls, ZeroTrack and AntiTrack, to isolate reward suppression from explicit replacement. Both operate on four body-level tracking terms. For a term with squared error  $e_{t,b}$ , scale  $\sigma$ , and  $N = 14$  tracked bodies, the original reward is  $\exp[-\sum_b e_{t,b}/(N\sigma^2)]$ . Let  $A_t$  contain tracked descendants of the edited arm joints. Inside the edit support, ZeroTrack uses

$$r_t^{\text{zero}} = \exp\left[-\frac{\sum_{b \notin A_t} e_{t,b}}{N\sigma^2}\right]. \quad (6)$$

The original denominator  $N$  is retained, so “zeroing” removes selected error contributions from the coupled exponential reward. AntiTrack adds

$$r_t^{\text{anti}} = r_t^{\text{zero}} - \exp\left[-\frac{\sum_{b \in A_t} e_{t,b}}{|A_t|\sigma^2}\right]. \quad (7)$$

Both use  $m$  for these errors and equal the original tracking terms outside the edit support. Anchor rewards and physical regularizers remain unchanged. AntiTrack uses unit anti-reward weight.

---

#### Algorithm 1 RedirectMimic

---

**Require:** Motions  $\mathcal{M}$ , editor  $E$ , pretrained  $(\theta_0, \phi_0)$ , sampler  $Q_{\text{base}}$ , update/rollout lengths  $K, H$

**Ensure:** Edited policy  $\pi_\theta$

```

1: Initialize  $(\theta, \phi) \leftarrow (\theta_0, \phi_0)$ 
2: Pair targets  $\mathcal{M} \leftarrow \{m \mapsto E(m) : m \in \mathcal{M}\}$ 
3: Build edit-start pool  $\mathcal{P}$  and set  $\rho$  (Eq. (5))
4: for PPO update  $k = 1, \dots, K$  do
5:   Clear rollout buffer  $\mathcal{D} \leftarrow \emptyset$ 
6:   for rollout step  $h = 1, \dots, H$  do
7:     if reset required then
8:       Mix starts  $Q \leftarrow \rho \mathcal{U}(\mathcal{P}) + (1 - \rho) Q_{\text{base}}$ 
9:       Sample  $(m, t) \sim Q$ , retrieve  $\tilde{m} \leftarrow \mathcal{M}[m]$ 
10:      Reset from original  $s_t \leftarrow \text{Init}(m_t)$ 
11:    end if
12:    Observe original  $o_t \leftarrow \text{Obs}(s_t, m_t)$ 
13:    Act  $a_t \sim \pi_\theta(\cdot | o_t)$ 
14:    Simulate  $(s_{t+1}, r_t, d_t) \leftarrow \text{Step}(s_t, a_t, m, \tilde{m})$ 
15:    Collect  $(o_t, a_t, r_t, s_{t+1}, d_t)$  in  $\mathcal{D}$ 
16:    Set reset  $\leftarrow d_t, \quad t \leftarrow t + 1$ 
17:  end for
18:  Estimate  $(\hat{A}, \hat{V}) \leftarrow \text{GAE}(\mathcal{D}, V_\phi)$ 
19:  Update  $(\theta, \phi) \leftarrow \text{PPO}(\theta, \phi, \mathcal{D}, \hat{A}, \hat{V})$ 
20: end for
21: return  $\pi_\theta(\cdot | \text{Obs}(s, m))$ 

```

---

AntiTrack, ZeroTrack, and RedirectMimic share the base checkpoint, edited datasets, start distribution, termination, and update budget. RefEdit<sup>†</sup> supplies  $\tilde{m}$  directly to  $\pi_0$  after initialization from the common original-reference state. It measures tracking performance with deployment-time access to the edited reference, providing a target-feasibility reference under a different information condition from Eq. (1).

## IV. EXPERIMENTS

### A. Experiment Setup

We evaluate the 29-actuated-joint Unitree G1 in mlab using Retargeted AMASS for Robotics [31], [6]. The two tasks request complementary edits: S1 suppresses fast outward arm movements, and S2 replaces overhead-arm poses. Their training sets contain 7,509 and 6,650 motion clips, respectively.

All local methods use the same pretrained tracker, task-specific data, architecture, sampling, termination conditions, and 15,000-iteration fine-tuning budget. Their tracking objectives differ as described in Sec. III-E. Pretrained and Vanilla-FT assess the original behavior and continued training on its objective. The RefEdit<sup>†</sup> reference tracks  $\tilde{m}$  directly, with access to the replacement at deployment.

Table I distinguishes editing objectives and input conditions. RedirectMimic combines an explicit local replacement target with policy editing under the original input. AntiTrack and ZeroTrack learn local suppression without specifying the replacement, while RefEdit<sup>†</sup> supplies that replacement directly at deployment.

S1 evaluates all 173 eligible windows from 56 clips. S2 uses a source-stratified sample and a prospective feasibility filter, excluding windows where original- or edited-reference tracking falls in at least half the rollouts. This yields 149

TABLE I  
EDITING MECHANISMS AND DEPLOYMENT INPUTS.

| Method               | Local edit   | Explicit target | Policy edit  | Original input |
|----------------------|--------------|-----------------|--------------|----------------|
| Pretrained           | $\times$     | $\times$        | $\times$     | $\checkmark$   |
| Vanilla-FT           | $\times$     | $\times$        | $\times$     | $\checkmark$   |
| AntiTrack            | $\checkmark$ | $\times$        | $\checkmark$ | $\checkmark$   |
| ZeroTrack            | $\checkmark$ | $\times$        | $\checkmark$ | $\checkmark$   |
| <b>RedirectMimic</b> | $\checkmark$ | $\checkmark$    | $\checkmark$ | $\checkmark$   |
| RefEdit <sup>†</sup> | $\checkmark$ | $\checkmark$    | $\times$     | $\times$       |

Local edit: a body- and time-specific intervention. Explicit target: a specified replacement trajectory. Policy edit: the editing objective is learned in policy weights. Original input: deployment uses  $m$ . Marks describe method design, not measured success.

windows from 96 clips. Each window is evaluated with 16 randomized rollouts, covering the edit and 3s of context on each side. All controllers start from the same original-reference state and run without reference-error termination or automatic resets.

We measure content occurrence, replacement error, preserved-body error, and joint success. Content occurrence  $D_C$  requires at least 0.06s above the task threshold in core interval  $C$ . A fall  $F_H$  occurs when pelvis height drops below 0.35m anywhere in horizon  $H$ . With preserved-body error  $e_C^{\text{keep}}$  measured relative to  $m$  in the torso yaw frame, joint success is

$$J_C = \mathbf{1}[-D_C] \cdot \mathbf{1}[-F_H] \cdot \mathbf{1}[e_C^{\text{keep}} \leq 0.10\text{m}]. \quad (8)$$

Arm error to  $\tilde{m}$  measures replacement fidelity. Preserved-body errors within and outside the edit assess local retention. Tables report rollout means. Bold and underlining mark the best and second-best non-oracle values, with ties at the displayed precision. Curve intervals use clip-cluster bootstrap resampling.

### B. Local Editing Performance

RedirectMimic achieved the strongest overhead-content suppression and highest joint success among the matched local methods (Table II). ZeroTrack removes the selected arm’s tracking objective, leaving other reward terms to favor motion close to the original reference. AntiTrack rewards departure from that reference without specifying a destination. In contrast, tracking  $\tilde{m}$  rewards both leaving the overhead pose and approaching the chosen replacement. The selected rollout in Fig. 3 illustrates the resulting arm responses.

The lower replacement error shows that the learned response follows the intended alternative. Its joint-success advantage persists despite a higher fall rate than AntiTrack and remains under unfiltered S2 evaluation.

S1 separates replacement fidelity from threshold suppression. The local methods suppressed fast outward movement similarly, with ZeroTrack attaining the highest joint-success point estimate. Yet its arm trajectory remained close to the original reference, whereas RedirectMimic more closely followed  $\tilde{m}$ . Slowing a movement can satisfy the speed rule without changing its spatial path. Explicit targets therefore

TABLE II  
LOCAL REPLACEMENT PERFORMANCE. ROLLOUT-MEAN RATES AND TARGET ERRORS (M).

| Method                               | $J_C \uparrow$ | Content $\downarrow$ | Error $\downarrow$ | Fall $\downarrow$ |
|--------------------------------------|----------------|----------------------|--------------------|-------------------|
| <i>S2: overhead-arm replacement</i>  |                |                      |                    |                   |
| Pretrained                           | 0.245          | 0.737                | 0.238              | 0.014             |
| Vanilla-FT                           | 0.218          | 0.774                | 0.237              | <u>0.011</u>      |
| AntiTrack                            | <u>0.650</u>   | <u>0.328</u>         | <u>0.171</u>       | <b>0.009</b>      |
| ZeroTrack                            | 0.408          | 0.589                | 0.224              | <u>0.011</u>      |
| <b>RedirectMimic</b>                 | <b>0.942</b>   | <b>0.027</b>         | <b>0.079</b>       | 0.022             |
| RefEdit <sup>†</sup>                 | 0.972          | 0.003                | 0.046              | 0.013             |
| <i>S1: speed-defined replacement</i> |                |                      |                    |                   |
| Pretrained                           | 0.725          | 0.105                | 0.177              | 0.089             |
| Vanilla-FT                           | 0.745          | 0.120                | 0.177              | 0.079             |
| AntiTrack                            | 0.857          | <u>0.023</u>         | 0.174              | <u>0.061</u>      |
| ZeroTrack                            | <b>0.897</b>   | <b>0.020</b>         | <u>0.158</u>       | <b>0.049</b>      |
| <b>RedirectMimic</b>                 | <u>0.874</u>   | <u>0.023</u>         | <b>0.143</b>       | 0.064             |
| RefEdit <sup>†</sup>                 | 0.803          | 0.000                | 0.079              | 0.103             |

<sup>†</sup>Oracle receives  $\tilde{m}$ . Vanilla-FT retains the original training protocol and uses its S1-trained policy in both tasks.

improve the requested replacement even when binary suppression is similar.

The RefEdit<sup>†</sup> reference closely tracked  $\tilde{m}$  and nearly eliminated content in both tasks, supporting the feasibility of the replacement targets. RedirectMimic approached its overhead-arm joint performance while observing the original reference. In S1, accurate oracle arm tracking coexisted with more falls, showing why target fidelity and physical execution must be assessed together.

### C. Normal-Motion Retention

Normal-motion tracking remained close to pretrained behavior on 200 clips per task (Table III). Normal clips retain their original targets during fine-tuning, so the objective continues to favor the existing motion repertoire. The observed tracking errors and falls support retention beyond the edited clips.

TABLE III  
NORMAL-MOTION RETENTION. ERROR IS RETAINED-BODY DISTANCE TO  $m$  (M).

| Method               | S1                |                    | S2                |                    |
|----------------------|-------------------|--------------------|-------------------|--------------------|
|                      | Fall $\downarrow$ | Error $\downarrow$ | Fall $\downarrow$ | Error $\downarrow$ |
| Pretrained           | 0.098             | 0.070              | 0.095             | 0.070              |
| Vanilla-FT           | <b>0.090</b>      | <b>0.067</b>       | 0.098             | <u>0.070</u>       |
| AntiTrack            | <b>0.090</b>      | <b>0.067</b>       | 0.103             | 0.073              |
| ZeroTrack            | <u>0.093</u>      | <u>0.068</u>       | <b>0.093</b>      | <b>0.067</b>       |
| <b>RedirectMimic</b> | <u>0.093</u>      | <b>0.067</b>       | 0.098             | <u>0.070</u>       |

Vanilla-FT improved aspects of normal tracking while continuing to reproduce the selected content. Continued optimization of the original objective did not produce the desired replacement. The matched local comparisons attribute the editing benefit to the changed training objective rather than additional optimization alone.

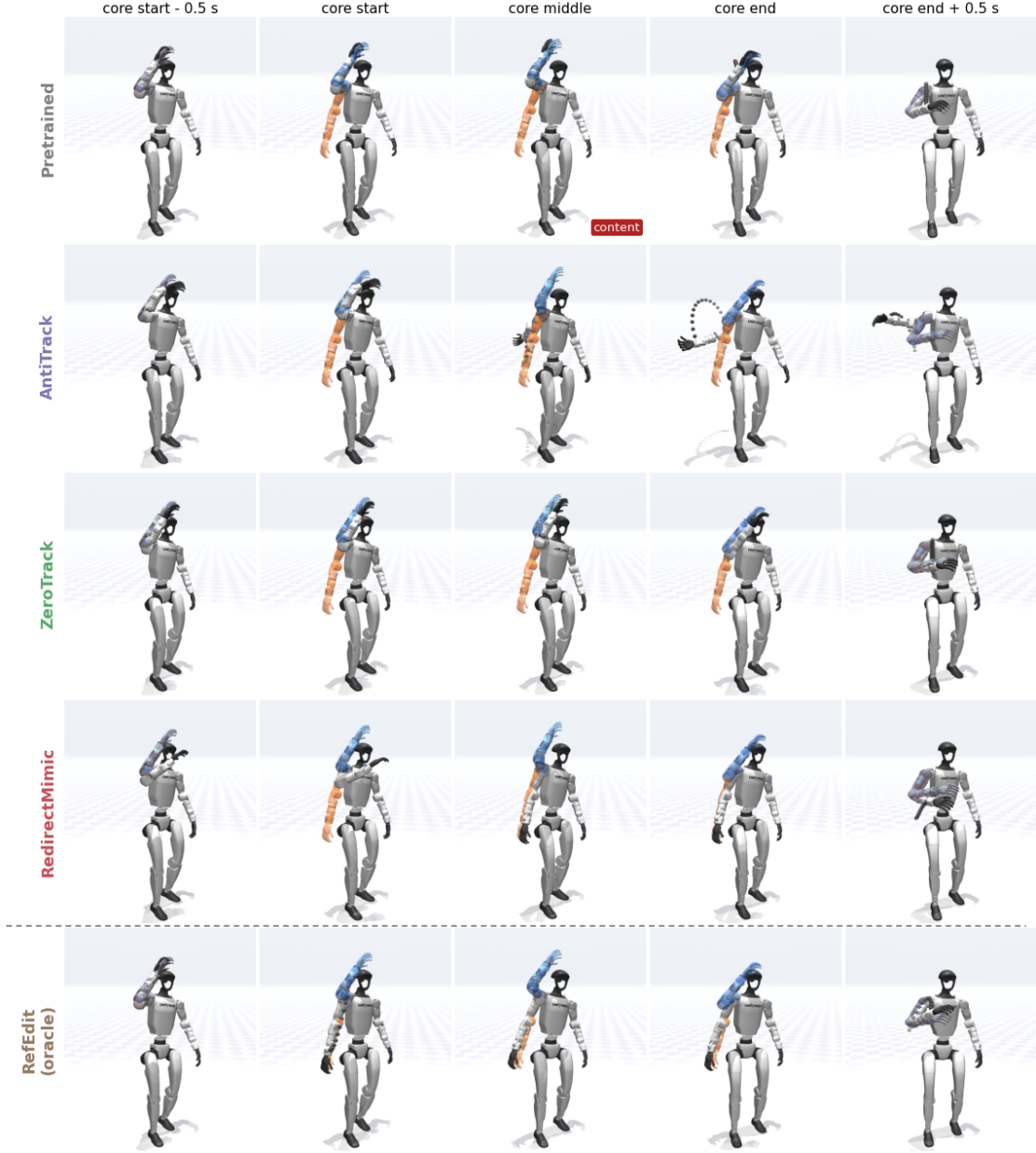


Fig. 3. Overhead-arm replacement in a selected successful GRAB cup-drinking example. Blue and orange ghosts show  $m$  and  $\tilde{m}$ , aligned to the torso. Dots trace recent wrist motion. Columns span the edit interval, and the last row is the oracle. Aggregate performance is reported in Table II.

#### D. Trajectory Redirection

To distinguish directed replacement from deviation, we project executed elbow and wrist positions  $x$  onto the intended displacement from  $m$  to  $\tilde{m}$ :

$$\alpha = \frac{\sum_{t,b} \langle x_{t,b} - m_{t,b}, \tilde{m}_{t,b} - m_{t,b} \rangle}{\sum_{t,b} \|\tilde{m}_{t,b} - m_{t,b}\|^2}. \quad (9)$$

Positions use the torso yaw frame, and sums pool each phase across nonfallen rollouts. The value  $\alpha = 1$  denotes the intended displacement along the edit direction. All values in Table IV are below one, so larger values indicate closer agreement along this direction. Target distance captures deviations that the projection does not resolve.

In S2, RedirectMimic followed the intended displacement while staying close to the replacement trajectory (Table IV, Fig. 4). ZeroTrack showed little directed redirection. AntiTrack moved farther from the original motion but had a larger target error.

S1 makes the complementary point. AntiTrack produced the larger displacement projection, while RedirectMimic stayed closer to the edited trajectory. A larger projection need not imply faithful replacement because it omits error perpendicular to the edit direction. Taken together, the two measures identify whether a controller reaches the requested alternative.

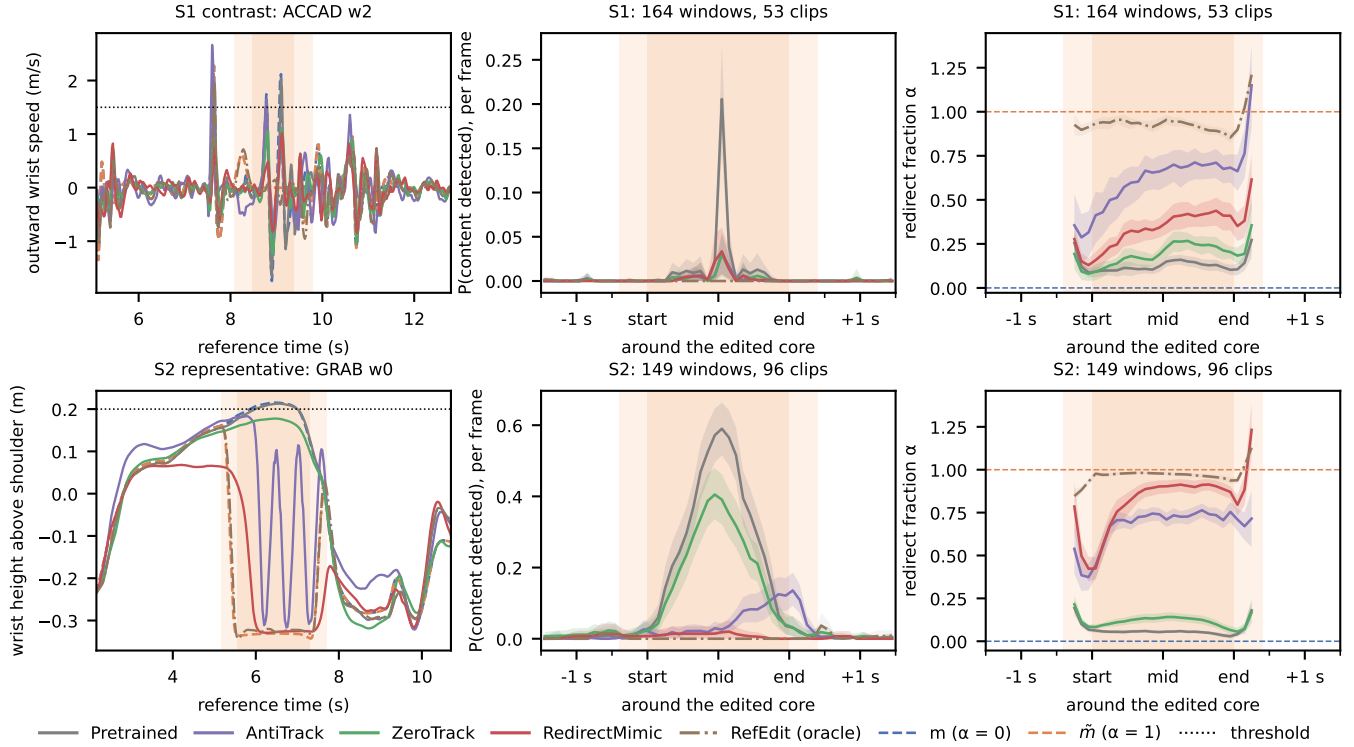


Fig. 4. Content suppression and trajectory redirection. Left: selected successful examples. Aggregate curves use feasibility-filtered pools (S1: 164 windows, S2: 149). S1 filtering is post hoc and does not affect the primary results. Core time is phase-normalized, and shading denotes clip-bootstrap intervals. The dash-dotted curve is the oracle.

TABLE IV  
CORE REDIRECTION ON NONFALLEN ROLLOUTS. ERROR IS  
ELBOW-WRIST DISTANCE TO  $\tilde{m}$  (M).

| Method               | S1                |                    | S2                |                    |
|----------------------|-------------------|--------------------|-------------------|--------------------|
|                      | $\alpha \uparrow$ | Error $\downarrow$ | $\alpha \uparrow$ | Error $\downarrow$ |
| Pretrained           | 0.12              | 0.283              | 0.05              | 0.389              |
| AntiTrack            | <b>0.64</b>       | 0.280              | <u>0.71</u>       | <u>0.273</u>       |
| ZeroTrack            | 0.20              | <u>0.255</u>       | 0.12              | 0.370              |
| <b>RedirectMimic</b> | <u>0.35</u>       | <b>0.218</b>       | <b>0.84</b>       | <b>0.110</b>       |
| RefEdit <sup>†</sup> | 0.92              | 0.080              | 0.97              | 0.056              |

### E. Ablation Studies

A controlled arm-mirroring experiment on one LAFANI motion isolates edit-window sampling. Without targeted sampling, both scratch training and fine-tuning retained the original content. Adding edit-pool starts enabled both to learn the replacement. Repeated exposure to the edited interval was therefore necessary in this single-pair experiment.

The S1 initialization comparison tests reuse of an existing tracker. Both scratch training and fine-tuning learned local replacements, with fine-tuning yielding closer target agreement and similar joint success relative to clip-level uncertainty. This supports fine-tuning as a route for modifying an available repertoire. The comparison treats the pretrained tracker as an existing resource, so total training costs differ.

## V. DISCUSSION

Local motion redaction depends on both the desired replacement and the information available to the controller. The paired-reference formulation connects these two aspects by making the alternative motion part of the learning signal. Reward suppression leaves the replacement unspecified, whereas tracking  $\tilde{m}$  rewards a particular local response. The overhead-arm results demonstrate the value of this distinction under matched training conditions. The speed-defined task shows why replacement fidelity remains informative even when binary suppression rates are similar.

This separation allows the policy’s local response to change while preserving the reference interface. Applications with edited commands can use direct reference tracking, as measured by RefEdit<sup>†</sup>. When the interface supplies the original motion, RedirectMimic learns the replacement through fine-tuning without adding an edit mask or replacement trajectory to the actor. The normal-motion results support retaining the existing repertoire while modifying the selected behavior on the evaluated clips.

The same interface constraint also shapes the timing of a replacement. The offline editor uses full-clip context to construct smooth transitions, while the actor receives current-frame reference inputs. Our experiments establish the learned response on known clips in simulation. Extending this setting to streaming inputs calls for evaluating transition timing and content throughout the rollout, including whether events shift beyond the intended edit interval.

## VI. CONCLUSION

RedirectMimic learns local motion replacements by separating the actor’s original reference from its edited tracking-reward target. Under matched local fine-tuning conditions, explicit targets give the best overhead-arm joint success and the lowest replacement error across both tested content rules. The resulting controller retains its original observation and action interfaces. The central lesson is to train toward the desired local response: a replacement target turns content suppression into directed motion editing.

## REFERENCES

- [1] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, “OmniH2O: Universal and Dexterous Human-to-Humanoid Whole-Body Teleoperation and Learning,” arXiv:2406.08858, 2024. [Online]. Available: <https://arxiv.org/abs/2406.08858>
- [2] Y. Ze, S. Zhao, W. Wang, A. Kanazawa, R. Duan, P. Abbeel, G. Shi, J. Wu, and C. K. Liu, “TWIST2: Scalable, Portable, and Holistic Humanoid Data Collection System,” arXiv:2511.02832, 2025. [Online]. Available: <https://arxiv.org/abs/2511.02832>
- [3] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, “DeepMimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 143:1–143:14, 2018. [Online]. Available: <https://arxiv.org/abs/1804.02717>
- [4] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, and C. K. Liu, “BeyondMimic: From motion tracking to versatile humanoid control via guided diffusion,” arXiv:2508.08241, 2025. [Online]. Available: <https://arxiv.org/abs/2508.08241>
- [5] Z. Wang, J. Wang, S. Yao, Y. Zhang, Z. Ding, M. Yang, Y. Wang, H. Jiang, C. Ma, X. Shi, and Z. Lu, “General humanoid whole-body control via pretraining and fast adaptation,” *arXiv preprint arXiv:2602.11929*, 2026. [Online]. Available: <https://arxiv.org/abs/2602.11929>
- [6] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, “AMASS: Archive of motion capture as surface shapes,” arXiv:1904.03278, 2019. [Online]. Available: <https://arxiv.org/abs/1904.03278>
- [7] X. Luan, Z. Lei, C. Gong, S. Li, Y. Bi, and J. Liu, “ForgetMimic: Motion unlearning for reinforcement learning humanoid control,” arXiv:2609.28378, 2026. [Online]. Available: <https://arxiv.org/abs/2609.28378>
- [8] E. D. Matteis, M. Migliarini, A. Sampieri, I. Spinelli, and F. Galasso, “Human motion unlearning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 5, 2026, pp. 3533–3541. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/37351>
- [9] Y. Wang, Z. Zhang, Y. Wang, and H. Tang, “SafeMo: Linguistically grounded unlearning for trustworthy text-to-motion generation,” arXiv:2601.00590, 2026. [Online]. Available: <https://arxiv.org/abs/2601.00590>
- [10] H. Wang, J. Lei, D. Li, B. Liu, M. Zheng, M. Li, R. Zhang, and Z. Fan, “Behavior uncloning: Distilling mode redirection into policy weights without inference-time steering,” arXiv:2606.29201, 2026. [Online]. Available: <https://arxiv.org/abs/2606.29201>
- [11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” arXiv:1707.06347, 2017. [Online]. Available: <https://arxiv.org/abs/1707.06347>
- [12] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, “AMP: Adversarial motion priors for stylized physics-based character control,” *ACM Transactions on Graphics*, vol. 40, no. 4, 2021, article 144. [Online]. Available: <https://arxiv.org/abs/2104.02180>
- [13] Z. Luo, J. Cao, A. Winkler, K. Kitani, and W. Xu, “Perpetual humanoid control for real-time simulated avatars,” arXiv:2305.06456, 2023. [Online]. Available: <https://arxiv.org/abs/2305.06456>
- [14] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, “Expressive Whole-Body Control for Humanoid Robots,” arXiv:2402.16796, 2024. [Online]. Available: <https://arxiv.org/abs/2402.16796>
- [15] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang, “ExBody2: Advanced Expressive Humanoid Whole-Body Control,” arXiv:2412.13196, 2024. [Online]. Available: <https://arxiv.org/abs/2412.13196>
- [16] Z. Chen, M. Ji, X. Cheng, X. Peng, X. B. Peng, and X. Wang, “GMT: General Motion Tracking for Humanoid Whole-Body Control,” arXiv:2506.14770, 2025. [Online]. Available: <https://arxiv.org/abs/2506.14770>
- [17] Z. Luo, Y. Yuan, T. Wang, C. Li, F. Castañeda, S. Chen, Z.-A. Cao, J. Li, D. Minor, Q. Ben, J. Park, D. Sami, Z. Wang, X. Da, R. Ding, C. Hogg, L. Song, E. Lim, E. Jeong, T. He, H. Xue, W. Xiao, S. Yuen, J. Kautz, Y. Chang, U. Iqbal, L. J. Fan, and Y. Zhu, “SONIC: Supersizing Motion Tracking for Natural Humanoid Whole-Body Control,” *Science Robotics*, vol. 11, no. 117, p. ead4592, 2026. [Online]. Available: <https://arxiv.org/abs/2511.07820>
- [18] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, L. Fan, and Y. Zhu, “HOVER: Versatile Neural Whole-Body Controller for Humanoid Robots,” arXiv:2410.21229, 2024. [Online]. Available: <https://arxiv.org/abs/2410.21229>
- [19] Y. Ze, Z. Chen, J. P. Araújo, Z.-a. Cao, X. B. Peng, J. Wu, and C. K. Liu, “TWIST: Teleoperated Whole-Body Imitation System,” arXiv:2505.02833, 2025. [Online]. Available: <https://arxiv.org/abs/2505.02833>
- [20] F. Amadio and E. M. Hoffman, “What Matters in Humanoid General Motion Tracking? An Empirical Study,” arXiv:2607.19903, 2026. [Online]. Available: <https://arxiv.org/abs/2607.19903>
- [21] K. Zakka, Q. Liao, B. Yi, L. L. Lay, K. Sreenath, and P. Abbeel, “mjlab: A lightweight framework for GPU-accelerated robot learning,” arXiv:2601.22074, 2026. [Online]. Available: <https://arxiv.org/abs/2601.22074>
- [22] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbab, C. Pan, Z. Yi, G. Qu, K. Kitani, J. Hodgins, L. J. Fan, Y. Zhu, C. Liu, and G. Shi, “ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills,” arXiv:2502.01143, 2025. [Online]. Available: <https://arxiv.org/abs/2502.01143>
- [23] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang, “AMO: Adaptive Motion Optimization for Hyper-Dexterous Humanoid Whole-Body Control,” arXiv:2505.03738, 2025. [Online]. Available: <https://arxiv.org/abs/2505.03738>
- [24] Y. Li, P. Zhi, Y. Wang, T. Liu, S. Yan, W. Liu, X. Wang, B. Jia, and S. Huang, “OmniTrack: General Motion Tracking via Physics-Consistent Reference,” arXiv:2602.23832, 2026. [Online]. Available: <https://arxiv.org/abs/2602.23832>
- [25] H. Li, P. Li, Y. Tao, Z. Wang, F. Xu, J. Chen, Y. Yuan, D. Jia, Y. Jin, M. Fan, G. Sartoretti, and H. Wang, “PGMT: Perceptive General Motion Tracking for Humanoid Robots,” arXiv:2609.08511, 2026. [Online]. Available: <https://arxiv.org/abs/2609.08511>
- [26] C. Guo, Y. Mu, M. G. Javed, S. Wang, and L. Cheng, “MoMask: Generative Masked Modeling of 3D Human Motions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. [Online]. Available: <https://arxiv.org/abs/2312.00063>
- [27] S. Cohan, G. Tevet, D. Reda, X. B. Peng, and M. van de Panne, “Flexible Motion In-betweening with Diffusion Models,” in *ACM SIGGRAPH Conference Papers*, 2024. [Online]. Available: <https://arxiv.org/abs/2405.11126>
- [28] K. Zhao, G. Li, and S. Tang, “DartControl: A Diffusion-Based Autoregressive Motion Model for Real-Time Text-Driven Motion Control,” in *International Conference on Learning Representations*, 2025. [Online]. Available: <https://arxiv.org/abs/2410.05260>
- [29] C. Tessler, Y. Guo, O. Nabati, G. Chechik, and X. B. Peng, “MaskedMimic: Unified Physics-Based Character Control Through Masked Motion Inpainting,” arXiv:2409.14393, 2024. [Online]. Available: <https://arxiv.org/abs/2409.14393>
- [30] G. Tevet, S. Raab, S. Cohan, D. Reda, Z. Luo, X. B. Peng, A. H. Bermanno, and M. van de Panne, “CLOSD: Closing the Loop between Simulation and Diffusion for multi-task character control,” in *International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=pZISppZSTv>
- [31] K. Zhao, “Retargeted AMASS for robotics,” Hugging Face dataset, revision 275c4e4, accessed: 2026-10-26. [Online]. Available: [https://huggingface.co/datasets/fleaven/Retargeted\\_AMASS\\_for\\_robotics](https://huggingface.co/datasets/fleaven/Retargeted_AMASS_for_robotics)