THE RESET-TO-ROLLOUT GAP IN ADAPTIVE SAMPLING FOR HUMANOID MOTION TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in humanoid motion tracking have enabled policies to imitate diverse human motions with high fidelity. Learning such repertoires requires allocating training samples across motions of varying difficulty. Probability-based adaptive sampling blends failure-based reset prioritization with uniform sampling. Weaker blending favors faster improvement in overall tracking quality, whereas stronger blending promotes difficult-motion acquisition but can still leave some motions slow to learn or unlearned. We find a *reset-to-rollout gap*: difficult regions can contribute a substantially smaller share of rollout samples than their reset probabilities suggest. Increasing reset priority alone therefore does not ensure sufficient local practice for difficult-motion acquisition. To address this gap while combining weaker blending's faster refinement with stronger blending's acquisition benefits, we introduce *Adaptive Rehearsal Allocation* (ARA). This plug-in reserves a bounded pool of parallel environments for repeated rehearsal of difficult regions, while the remaining environments retain ordinary adaptive sampling. Across the evaluated settings, ARA consistently acquires all target motions. In training from scratch, weaker adaptive sampling augmented with ARA acquires all target motions and achieves better tracking quality than stronger adaptive sampling alone under the same training budget.

1 INTRODUCTION

Humanoid motion tracking via reinforcement learning (RL) has advanced rapidly, with recent systems reproducing diverse human motions at high fidelity and scale (Peng et al., 2018; Liao et al., 2025; Chen et al., 2025; Ze et al., 2025b; Luo et al., 2025; Yin et al., 2025; He et al., 2025; Qi et al., 2026; Wang et al., 2026a). As a single policy learns a broader repertoire, allocating training samples across motions of different difficulty becomes increasingly important. A policy may track much of the corpus accurately while continuing to fail on a few challenging transitions. Training must therefore support both the acquisition of difficult motions and continued refinement of overall tracking quality.

Probability-based adaptive sampling addresses this uneven progress by blending a failure-rate-based distribution over temporal motion regions with uniform sampling (Liao et al., 2025; Luo et al., 2025). The resulting distribution determines where episodes begin, giving difficult regions higher reset priority while preserving broad coverage. The blending factor controls how strongly resets favor difficult regions. Weaker blending retains a larger uniform component and favors faster improvement in overall tracking quality, whereas stronger blending promotes difficult-motion acquisition. Yet even with stronger blending, some difficult motions require prolonged training or remain unlearned, including motions verified to be learnable through independent training.

We find a *reset-to-rollout gap*: difficult regions can contribute a substantially smaller share of collected training samples than their reset probabilities suggest. Reset probabilities govern where new attempts begin, whereas policy updates use transitions collected throughout each episode. Attempts at difficult transitions may terminate after only a few steps, contributing little local practice. Longer surviving episodes keep environments occupied and continue collecting samples elsewhere, delaying new opportunities to reset near difficult regions. Consequently, a region can receive high reset priority yet remain underexposed in the data used for learning. Increasing reset priority alone therefore does not ensure sufficient local practice for difficult-motion acquisition.

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

To address this gap while combining weaker blending's faster refinement with stronger blending's acquisition benefits, we introduce *Adaptive Rehearsal Allocation* (ARA), a plug-in mechanism for explicit rehearsal allocation. ARA reserves a bounded pool of parallel environments for repeated rehearsal of selected difficult regions. The remaining environments retain ordinary adaptive sampling to support broad coverage and tracking refinement. Allocating rehearsal capacity separately allows this background sampler to retain weaker blending while difficult regions receive additional practice. The regions receiving rehearsal change as learning progresses, and unused capacity returns to ordinary sampling.

Our evaluation varies adaptive sampling strength and environment count and includes training from scratch. Across the evaluated settings, ARA consistently acquires all target motions. In our from-scratch comparison, weaker adaptive sampling augmented with ARA reduces difficult-region underexposure, acquires all target motions, and achieves better tracking quality than stronger adaptive sampling alone under the same training budget.

Our contributions are:

- We identify and quantify the *reset-to-rollout gap*, showing that difficult regions can remain underexposed in rollout samples despite high reset priority.
- We introduce ARA, a plug-in mechanism for explicit rehearsal allocation that supports complete motion acquisition across the evaluated settings and enables weaker adaptive sampling to retain better tracking quality than stronger adaptive sampling alone.

2 Related Work

Humanoid Motion Tracking. Reinforcement learning for motion tracking, popularized by DeepMimic (Peng et al., 2018), has become a central approach to learning physics-based humanoid skills from reference motions. Recent work spans general trackers and teleoperation (He et al., 2024; Ji et al., 2024; Chen et al., 2025; Yin et al., 2025; Ze et al., 2025b;c; Li et al., 2025b), dynamic and long-horizon motion tracking (Xie et al., 2025; Han et al., 2025; Liao et al., 2025), and athletic control (Zhang et al., 2026). Broader humanoid-control research explores behavioral foundation models (Zeng et al., 2025; Li et al., 2025a; Tirinzoni et al., 2025), expert aggregation (Wang et al., 2025; 2026a), residual adaptation (Wang et al., 2026b; Sun et al., 2026), rapid sim-to-real training (Seo et al., 2025), and scaling datasets and model capacity (Luo et al., 2025; Qi et al., 2026). Complementary character-control work learns reusable adversarial motion priors (Peng et al., 2021).

Adaptive Curriculum and Sampling. Curriculum learning in RL prioritizes tasks, states, or environments using estimates of learning utility. Examples include TD-error-based level prioritization in Prioritized Level Replay (Jiang et al., 2021), absolute learning progress in ALP-GMM (Portelas et al., 2020), and regret-driven environment generation in PAIRED (Dennis et al., 2020).

Probability-based adaptive sampling similarly prioritizes difficult reference motions or temporal regions. MaskedMimic (Tessler et al., 2024) uses motion-level failure rates on flat terrain, with a minimum sampling weight to preserve coverage. GMT (Chen et al., 2025) combines randomized motion clipping with sampling probabilities based on completion history and tracking error, addressing the limited practice received by difficult portions of long motions. BeyondMimic and SONIC (Liao et al., 2025; Luo et al., 2025) operate at finer temporal-bin granularity, using online failure statistics to prioritize reset locations while retaining a uniform component for coverage. These reset probabilities govern where attempts begin, while episode survival and the regions visited determine the rollout samples they contribute. ARA complements these approaches by explicitly reserving a bounded pool of environments for repeated local rehearsal of persistently difficult regions, while the remaining environments retain ordinary adaptive sampling.

3 Method

We introduce the tracking setup and adaptive sampler in Section 3.1, analyze the reset-to-rollout gap in Section 3.2, and present ARA's rehearsal allocation and recycling mechanism in Section 3.3. Figure 1 summarizes the overall training pipeline.

3.1 PRELIMINARIES

3.1.1 HUMANOID MOTION TRACKING

The motion corpus $\mathcal{D} = \{m_1, \ldots, m_{|\mathcal{D}|}\}$ contains time-indexed reference trajectories. At timestep t, the observation o_t combines proprioceptive and reference-motion features. The policy with parameters θ samples an action $a_t \sim \pi_\theta(\cdot \mid o_t)$, specifying target joint positions for low-level PD controllers. The reward encourages reference tracking while regularizing control actions. An episode follows its reference motion until a tracking failure, the ordinary timeout of H_{long} environment steps, or motion completion, after which the environment resets to a sampled reference state.

We train with PPO (Schulman et al., 2017), collecting H_{roll} transitions from each of N parallel environments per update. The resulting buffer contains NH_{roll} transitions, regardless of how many episodes terminate during collection. An episode may span multiple buffers, and a buffer may contain several episodes from one environment. Thus, the collection horizon fixes the update batch size, while episode duration determines when an environment becomes available for another reset.

3.1.2 BIN-BASED ADAPTIVE SAMPLING

The reference corpus is divided into B temporal bins of duration Δ_{bin} seconds, indexed by $\mathcal{B}_{\text{bin}} = \{1, \ldots, B\}$. For bin b, the historical failure statistic f_b is the cumulative number of failures occurring in that bin divided by cumulative rollout exposure, measured in units of the bin's duration. Following probability-based curricula (Luo et al., 2025; Liao et al., 2025), we blend normalized, capped failure weights with a uniform component. For the analysis, we assume resets begin within the selected bin, without the backward sampling window used in our implementation (Appendix C). The reset probability for bin b is

$$p_b = \alpha \frac{\min(f_b, \beta \bar{f})}{\sum_{j=1}^{B} \min(f_j, \beta \bar{f})} + (1 - \alpha)\frac{1}{B}, \tag{1}$$

where $\bar{f} = B^{-1} \sum_{j=1}^{B} f_j$ and $\beta > 0$ caps failure weights relative to their mean. The blending factor $\alpha \in [0, 1]$ controls the emphasis on difficult bins: larger α strengthens failure-based prioritization, while smaller α retains more uniform sampling. If all failure weights vanish, the adaptive component defaults to uniform sampling.

At each reset, the sampler draws a bin according to p_b and a reference time within that bin.

3.2 THE RESET-TO-ROLLOUT GAP

3.2.1 FROM RESET PROBABILITY TO ROLLOUT SHARE

Consider episodes collected under ordinary adaptive sampling. Let q_b denote the fraction of collected rollout steps whose reference phase lies in bin b. For an episode initialized in bin i, let L_i be its length in environment steps and $\ell_{i \to b}$ the number of those steps spent in bin b, so that $\sum_b \ell_{i \to b} = L_i$. Over a sufficiently long collection interval with an approximately stationary policy and reset distribution,

$$q_b \approx \frac{\sum_{i=1}^{B} p_i \, \mathbb{E}[\ell_{i \to b}]}{\sum_{i=1}^{B} p_i \, \mathbb{E}[L_i]}. \tag{2}$$

The expectations are conditioned on initialization in bin i and include variation in the initial state and subsequent trajectory. The numerator is the expected number of local steps in b per reset, including visits from episodes initialized elsewhere; the denominator is the expected episode length. The approximation neglects episode fragments at the collection boundaries. Under the same approximation, bin b contributes about $NH_{\text{roll}}q_b$ transitions per PPO update. Equation (2) therefore connects reset allocation to the local samples available for learning.

We characterize the reset-to-rollout gap using q_b/p_b for $p_b > 0$. A ratio below one indicates that a bin's rollout share is smaller than its initialization share.

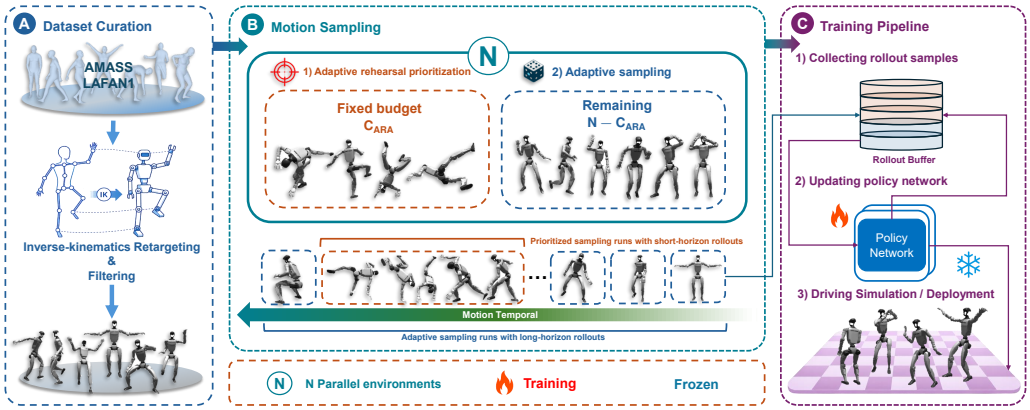


Figure 1: Overview of ARA. After the competence gate activates rehearsal, a bounded pool of environments is allocated to selected difficult regions. Complementary short-horizon recycling limits downstream sample collection, while adaptive sampling uses the remaining capacity.

3.2.2 Implications for Training Allocation

Equation (2) exposes two interacting effects. Attempts at difficult transitions may terminate quickly and contribute few local steps. Longer episodes can continue collecting samples elsewhere, increasing the average episode length and keeping environments occupied before another reset. Difficult regions can therefore receive high reset priority yet contribute little of the fixed rollout budget. Increasing adaptive sampling strength alone does not ensure sufficient local practice for difficult-motion acquisition.

Longer episodes are important for learning sustained tracking, as they expose the policy to accumulated tracking errors and recovery states that may be absent near reference-state initializations. Uniformly shortening the episode horizon would enable more frequent resets but reduce opportunities to learn from these states. This motivates allocating capacity for repeated local rehearsal while preserving longer tracking episodes in the remaining environments. The background sampler can then retain weaker blending for broad refinement while difficult regions receive additional practice.

3.3 ARA: Explicit Rehearsal Allocation

ARA reserves a bounded pool of environments for repeated rehearsal of selected difficult regions. The remaining environments retain adaptive sampling and long episodes for broad tracking refinement. A complementary short horizon recycles rehearsal capacity when successful attempts from regions with moderate failure rates continue beyond their targets.

3.3.1 Allocating Rehearsal Capacity

ARA maintains a recent difficulty estimate $\hat{f}_b \in [0,1]$ for each bin using an exponential moving average (EMA) of failures relative to local rollout exposure. Training begins with ordinary adaptive sampling. Rehearsal activates when competence, $c = 1 - B^{-1}\sum_b \hat{f}_b$, reaches a threshold τ_c. The gate then remains open, while allocation follows the evolving difficulty estimates.

Let $C_{\mathrm{ARA}} \in \{0, \ldots, N\}$ be the rehearsal budget and $D_{\max}$ the maximum demand per bin. The threshold τ_f specifies the minimum failure rate for a bin to be eligible for ARA's rehearsal budget. Bins with $\hat{f}_b \geq \tau_f$ receive demand $d_b = D_{\max}\hat{f}_b$; other bins have zero demand. Candidates are ranked by demand and allocated capacity within the remaining budget:

$$n_b = \left\lfloor \min\left(d_b, \left[C_{\mathrm{ARA}} - \sum_{j \succ b} d_j \right]_+ \right) \right\rfloor, \tag{3}$$

where $[x]_+ = \max(0, x)$ and $j \succ b$ denotes bins ranked ahead of b. The integer n_b specifies the target number of environments assigned to rehearse bin b.

These targets are filled as environments reset. Let R_b count ongoing rehearsal assignments to bin b among environments not being reset. Each environment counts toward its assigned target throughout the approach and rehearsal, and retains that assignment until termination. ARA fills the deficits $\delta_b = [n_b - R_b]_+$ within the remaining pool capacity. Other available environments use background adaptive sampling over bins outside the current rehearsal set, including capacity left unused when demand is low.

3.3.2 Local Rehearsal and Recycling

Unlearned regions often produce short, failed attempts that naturally recycle rehearsal environments. As regions are learned, their failure estimates decrease and they typically leave the rehearsal pool. Some regions, however, retain moderate failure rates and remain allocated capacity even as successful attempts continue beyond their targets. These longer attempts can accumulate excessive downstream samples and delay further local practice. ARA therefore complements capacity allocation with short-horizon recycling, allowing attempts to traverse their target regions before returning the environments to rehearsal.

Each attempt begins within its target's backward window. The target region is the contiguous sequence of selected bins within the same motion that contains the assigned target. For environment e, the region endpoint ϕ_e^{end} is recorded at assignment and remains fixed during the attempt. A rehearsal timeout occurs once both the short horizon has elapsed and the reference phase has passed this endpoint:

$$\mathrm{timeout}_e^{\mathrm{rehearsal}} = \mathbf{1}[k_e \geq H_{\mathrm{short}}]\,\mathbf{1}[\phi_e > \phi_e^{\mathrm{end}}], \tag{4}$$

Here, k_e is the number of steps since reset, H_{short} is the minimum rehearsal duration in steps, and ϕ_e and ϕ_e^{end} are reference times in seconds. The indicator $\mathbf{1}[\cdot]$ equals one when its condition holds and zero otherwise. The endpoint condition allows the approach and target traversal to finish before recycling. Rehearsal timeouts use value bootstrapping from the final state; tracking failures and motion completion end attempts normally. Terminated environments are reassigned according to current demand.

Algorithm 1 summarizes allocation and termination. The set $\mathcal{E}$ contains environments available for reset, and $\mathcal{A}$ contains new environment–target assignments. Bold symbols collect the corresponding per-bin quantities, and K denotes the remaining rehearsal capacity. $\mathrm{BudgetFill}$ applies Eq. (3), and $\mathrm{FillDeficits}$ assigns available environments to unmet targets within the remaining budget. $\mathrm{InitializeRehearsal}$ samples local starts and records region endpoints; $\mathrm{InitializeAdaptive}$ handles the remaining environments. $\mathrm{Envs}(\mathcal{A})$ denotes the environments assigned to rehearsal. The Boolean flags $\mathrm{failure}_e$ and $\mathrm{motion_end}_e$ indicate tracking failure and motion completion.

4 Experiments

We evaluate how adaptive sampling strength and ARA affect difficult-motion acquisition, overall tracking quality, and local rollout exposure. Tracking evaluations, exposure diagnostics, and acquisition curves connect sample allocation with learning across training settings.

4.1 Experimental Setup

We use MJLAB (Zakka et al., 2026) for training and evaluation. Our motion corpus combines 19 difficult motions from AMASS (Mahmood et al., 2019) with the full LAFAN1 dataset (Harvey et al., 2020). The AMASS motions form a subset of those selected in OmniXtreme (Wang et al., 2026a). AMASS motions are retargeted using GMR (Ze et al., 2025a), while LAFAN1 uses references retargeted by the official Unitree repository. The resulting 59 motions contain approximately 450,000 reference frames at 50 fps. Each motion is evaluated over 1,000 trials.

We evaluate motion acquisition and tracking quality using the following metrics. *Success rate* (S.R.) measures the percentage of successful evaluation trials. During evaluation, tracking failure occurs only when the anchor (torso) height deviates from the reference by more than $0.25\,\mathrm{m}$. The *successful-motion count* (S.C.) is the number of evaluated motions whose success rate is strictly greater than 50%, reported out of the 59 target motions. To characterize tracking fidelity, we report *root-position error* E_{pos} in meters, *root-velocity error* E_{vel} in meters per second, and *mean absolute*

Algorithm 1 Core ARA allocation and termination

Require: Budget C_{ARA}, cap $D_{\max}$, thresholds τ_c, τ_f, horizon H_{short}

Persistent gate: $g \leftarrow \mathrm{false}$ ▷ Initially inactive
procedure ASSIGNENVIRONMENTS($\mathcal{E}, \hat{\mathbf{f}}, \mathbf{R}$)
  $\mathcal{A} \leftarrow \varnothing$
  $g \leftarrow g \vee (1 - B^{-1} \sum_b \hat{f}_b \geq \tau_c)$ ▷ Activate rehearsal
  if g **then**
    $d_b \leftarrow D_{\max} \hat{f}_b \mathbf{1}[\hat{f}_b \geq \tau_f]$ for all b ▷ Eligible-bin demand
    $\mathbf{n} \leftarrow \mathrm{BudgetFill}(\mathbf{d}, C_{\mathrm{ARA}})$ ▷ Prioritize highest demand
    $\delta_b \leftarrow [n_b - R_b]_+$; $K \leftarrow [C_{\mathrm{ARA}} - \sum_b R_b]_+$ ▷ Remaining capacity
    $\mathcal{A} \leftarrow \mathrm{FillDeficits}(\mathcal{E}, \boldsymbol{\delta}, K)$ ▷ Fill unmet targets
    $\mathrm{InitializeRehearsal}(\mathcal{A})$ ▷ Local starts; fixed endpoints
  end if
  $\mathrm{InitializeAdaptive}(\mathcal{E} \setminus \mathrm{Envs}(\mathcal{A}))$ ▷ Background sampling
  return $\mathcal{A}$
end procedure
procedure TERMINATEREHEARSAL(e)
  $h_e \leftarrow (k_e \geq H_{\mathrm{short}}) \wedge (\phi_e > \phi_e^{\mathrm{end}})$ ▷ Horizon and region passed
  return $\mathrm{failure}_e \vee \mathrm{motion_end}_e \vee h_e$ ▷ End rehearsal attempt
end procedure

joint-angle error E_{mpjpe} in radians. The position and velocity errors measure Euclidean differences from the reference, while the joint-angle error is averaged across actuated joints.

4.2 MOTION ACQUISITION ACROSS SAMPLING SETTINGS

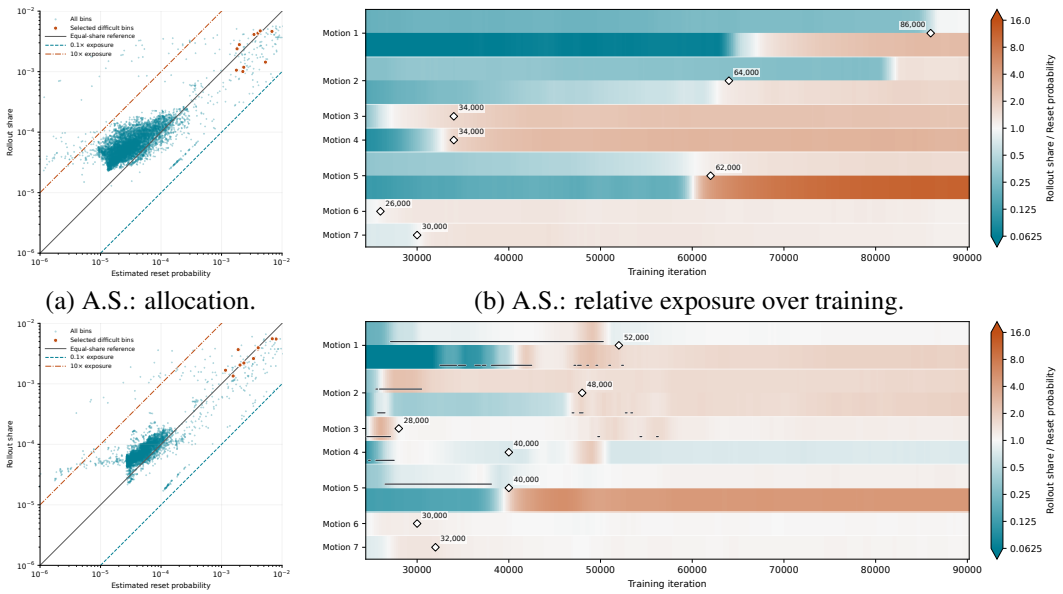
Training setup. We initialize all runs from the same pretrained checkpoint and vary the number of parallel environments and adaptive sampling strength. We compare pure adaptive sampling (A.S.) with adaptive sampling augmented by ARA (A.S.+ARA). ARA reserves at most $C_{\mathrm{ARA}} = 1{,}000$ environments for rehearsal, with each selected bin allocated at most $D_{\max} = 300$ environments. We set the minimum failure rate for ARA eligibility to $\tau_f = 0.5$ and use an EMA update rate of 0.1 for ARA's difficulty estimates. All runs use the same number of training iterations. Results thus reflect progress within a fixed budget rather than fully converged performance.

Table 1: Tracking performance across environment counts and sampling strengths.

#Envs	α	Method	S.R. ↑	S.C. ↑	E_{pos} ↓	E_{vel} ↓	E_{mpjpe} ↓
16384	0.9	A.S.	98.91	59	0.111	0.247	0.088
16384	0.9	A.S.+ARA	99.01	59	0.112	0.247	0.089
8192	0.9	A.S.	96.08	57	0.120	0.256	0.100
8192	0.9	A.S.+ARA	97.63	59	0.122	0.257	0.099
8192	0.3	A.S.	92.62	56	0.114	0.246	0.088
8192	0.3	A.S.+ARA	97.23	59	0.119	0.251	0.093

Motion acquisition and tracking quality. Table 1 shows that A.S.+ARA consistently learns all motions across the evaluated settings. Increasing the environment count or weakening adaptive sampling improves overall tracking quality within the fixed training budget. However, weaker adaptive sampling alone can leave some difficult motions unlearned. Training logs show that motion acquisition has plateaued at this point, suggesting that further training under the same configuration is unlikely to yield newly learned motions.

Under stronger blending, ARA achieves complete motion acquisition while maintaining tracking quality comparable to pure adaptive sampling; under weaker blending, it trades some tracking quality for complete acquisition. Stronger blending, however, improves overall tracking quality more slowly than weaker blending. These results indicate a difference in convergence behavior: sustained emphasis on difficult regions can delay broader tracking refinement, whereas weaker sampling fa-



(a) A.S.: allocation.

(b) A.S.: relative exposure over training.

(c) A.S.+ARA: allocation.

(d) A.S.+ARA: relative exposure over training.

Figure 2: **Reset-to-rollout gap during adaptive training.** (a,c) Highlighted points identify the selected difficult bins. (b,d) Heatmaps are smoothed for visibility. Short dark horizontal lines in (d) indicate periods when the corresponding bin is sampled by ARA. White diamonds mark the first evaluated checkpoint at which the corresponding motion's success rate exceeds 50%, indicating first observed motion acquisition.

vors overall tracking quality but may leave gaps in motion acquisition. ARA therefore enables weaker adaptive sampling to achieve complete motion acquisition while retaining better tracking quality than stronger adaptive sampling alone at the same environment count.

4.3 From-Scratch Training: Exposure and Acquisition

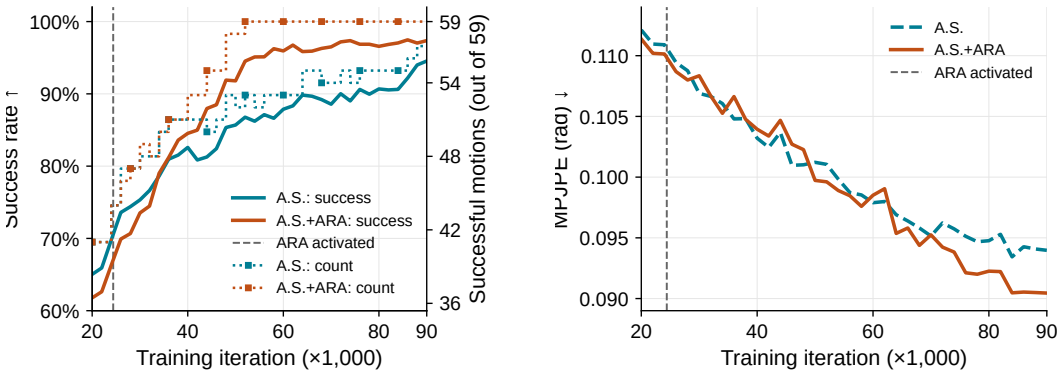
Training setup. We compare from-scratch training with adaptive sampling (A.S.) at $\alpha = 0.9$ and adaptive sampling augmented by ARA (A.S.+ARA), using a weaker nominal adaptive strength of $\alpha = 0.3$. Both use 8192 parallel environments. We use a competence-gate threshold of 99% to activate ARA. Training runs for approximately 90k iterations, ending shortly after A.S. acquires its last learned motion.

Exposure diagnostics. The scatter plots in Figure 2 compare each bin's reset probability, adjusted to account for backward sampling, with its share of rollout steps, both averaged over the entire training run. The heatmaps display only the period after ARA activation and show rollout share divided by adjusted reset probability for ten difficult bins held fixed across methods. For A.S.+ARA, rollout shares include rehearsal environments, while the probability reference includes only adaptive sampling. Heatmap rows are grouped by motion, with matching order and color scales across methods.

4.3.1 Reset-to-Rollout Gap

Training-averaged allocation. In Figures 2a and 2c, most bins with low reset probabilities receive disproportionately large shares of rollout samples. This pattern is consistent with survival bias: easier regions sustain longer episodes and contribute more samples. Several difficult bins remain substantially underexposed under pure A.S., whereas ARA increases their rollout exposure relative to the background sampler's reset allocation despite weaker adaptive sampling.

Persistent deficits and targeted rehearsal. Training-wide averages can obscure more severe deficits during learning. Figure 2b shows that selected difficult bins remain substantially under-



(a) Success rate and successful-motion count.

(b) Tracking fidelity.

Figure 3: **Learning outcomes after 20k iterations.** Left: aggregate success rate (solid lines, left axis) and the number of motions with success rate above 50% (dotted steps, right axis). Right: MPJPE in radians (lower is better). Both panels compare A.S. and A.S.+ARA over 59 motions at 2,000-iteration checkpoints. Vertical dashed lines mark ARA activation at iteration $24{,}400$.

sampled for extended periods. The most severe deficit occurs in Motion 1, where the smoothed rollout-to-reset ratio falls to approximately 0.055: the bin has a rollout share equal to 0.055 times its adjusted reset probability. ARA (Figure 2d) substantially compensates for these deficits in several hard regions by increasing the local practice devoted to them. This pattern is consistent with its allocation strategy: ARA assigns rehearsal environments to hard bins in priority order within a fixed budget, rather than determining their rehearsal allocation from adaptive sampling probabilities. Targeted practice can therefore increase even when the adaptive sampler uses weaker emphasis on difficult regions.

4.3.2 Motion Acquisition and Tracking Quality

We next examine whether weaker adaptive sampling augmented by ARA can effectively acquire difficult motions while improving overall tracking quality, compared with stronger adaptive sampling alone. Motion acquisition is the first evaluated checkpoint at which whole-motion success exceeds 50%, consistent with the markers in Figure 2.

Learning dynamics and exposure. Before ARA activates, weaker adaptive sampling yields slightly lower joint-angle error but lower success rate than stronger adaptive sampling (Figure 3a,b). Following activation, motion acquisition accelerates, while overall tracking quality initially remains comparable between the two runs. During active rehearsal, ARA reserves a bounded portion of the environment budget for selected difficult bins, leaving the remaining environments to support broad tracking refinement. This allocation supports faster acquisition alongside a similar initial rate of improvement in overall tracking quality. The ARA run also exhibits noticeable tracking-error fluctuations, which may reflect the concentration of rehearsal updates on a small set of bins.

As rehearsal demand subsides, ARA returns the reserved environments to weaker adaptive sampling. Once no bins remain eligible, the entire environment budget supports broad tracking refinement. This reallocation is consistent with the tracking-quality advantage that emerges later in training (Figure 3b). By the time stronger A.S. reaches its final observed motion acquisition, A.S.+ARA already achieves lower tracking error. At iteration $90{,}000$, A.S. and A.S.+ARA reach success rates of 94.54% and 97.37%, respectively, with 57 and 59 successful motions and joint-angle errors of 0.09397 rad and 0.09045 rad. Together with the exposure diagnostics, these trends connect reduced hard-bin underexposure with complete motion acquisition and improved tracking quality within the evaluated training budget.

4.4 Generalization Evaluation

We evaluate generalization on $1{,}000$ randomly selected motions from BONES-SEED (Bones Studio, 2026), a motion-capture dataset used in SONIC (Luo et al., 2025). All selected motions are held out from both pretraining and subsequent training. Each motion is evaluated over $1{,}000$ trials using the protocol and metrics in Section 4.1.

Table 2 follows the tracking-quality pattern in Section 4.2. Under stronger blending, adding ARA maintains comparable tracking quality; under weaker blending, it slightly reduces tracking quality relative to pure adaptive sampling while retaining better tracking quality than stronger adaptive sampling alone at the same environment count. Success rates remain comparable across methods. Together with the acquisition results in Table 1, these results show that ARA combines complete acquisition of the training motions with comparable performance on held-out motions.

Table 2: Generalization across environment counts and sampling strengths.

#Envs	α	Method	S.R. $\uparrow$	$E_{\text{pos}} \downarrow$	$E_{\text{vel}} \downarrow$	$E_{\text{mpjpe}} \downarrow$
16384	0.9	A.S.	98.01	0.117	0.186	0.108
16384	0.9	A.S.+ARA	98.05	0.116	0.186	0.107
8192	0.9	A.S.	98.24	0.119	0.192	0.116
8192	0.9	A.S.+ARA	98.20	0.117	0.192	0.113
8192	0.3	A.S.	98.12	0.107	0.184	0.109
8192	0.3	A.S.+ARA	98.07	0.109	0.187	0.110

5 Limitations

ARA introduces additional hyperparameters, including the rehearsal budget C_{ARA}, per-bin cap $D_{\max}$, failure threshold τ_f, and competence threshold τ_c. In our experiments, their selection is more consequential when training uses fewer parallel environments. Motions acquired through rehearsal can subsequently deteriorate when background adaptive sampling provides insufficient samples to maintain them, leading to repeated acquisition and loss. We find that a larger per-bin cap and a lower failure threshold can improve retention by increasing rehearsal capacity and delaying release from rehearsal. With more parallel environments, these choices become less restrictive.

ARA targets a small set of persistently difficult regions and relies on the background sampler to learn and maintain the broader repertoire. Its effectiveness therefore depends on both components supporting acquisition and retention.

6 Conclusion

Probability-based adaptive sampling balances difficult-motion acquisition and overall tracking refinement through its blending factor. Yet difficult regions can receive substantially fewer rollout samples than their reset probabilities suggest. We quantified this reset-to-rollout gap and introduced ARA, a plug-in that allocates bounded environment capacity to difficult regions and recycles it through short-horizon rehearsal.

Across the evaluated settings, ARA consistently acquires all target motions. In the from-scratch comparison, weaker adaptive sampling augmented by ARA reduces hard-bin underexposure, achieves complete motion acquisition, and attains better tracking quality than stronger adaptive sampling alone within the same training budget. Explicit local allocation thus complements reset prioritization, supporting difficult-motion acquisition alongside efficient tracking refinement.

AI Use Statement

We used ChatGPT to assist with text polishing, code generation, and figure design. The authors take responsibility for the final content of this work.

REFERENCES

Bones Studio. BONES-SEED: Skeletal everyday embodiment dataset, 2026. URL https://huggingface.co/datasets/bones-studio/seed.

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.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

Jinrui Han, Weiji Xie, Jiakun Zheng, Jiyuan Shi, Weinan Zhang, Ting Xiao, and Chenjia Bai. Kungfubot2: Learning versatile motion skills for humanoid whole-body control. *arXiv preprint arXiv:2509.16638*, 2025.

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 39(4), 2020.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay, 2021. URL https://arxiv.org/abs/2010.03934.

Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, et al. Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning. *arXiv preprint arXiv:2511.04131*, 2025a.

Yixuan Li, Yutang Lin, Jieming Cui, Tengyu Liu, Wei Liang, Yixin Zhu, and Siyuan Huang. Clone: Closed-loop whole-body humanoid teleoperation for long-horizon tasks. *arXiv preprint arXiv:2506.08931*, 2025b.

Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *arXiv preprint arXiv:2508.08241*, 2025.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castañeda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pp. 5442–5451, October 2019.

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

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

Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *Conference on Robot Learning*, pp. 835–853. PMLR, 2020.

Zekun Qi, Xuchuan Chen, Dairu Liu, Chenghuai Lin, Yunrui Lian, Sikai Liang, Zhikai Zhang, Yu Guan, Jilong Wang, Wenyao Zhang, et al. Humanoid-gpt: Scaling data and structure for zero-shot motion tracking. *arXiv preprint arXiv:2606.03985*, 2026.

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

Younggyo Seo, Carmelo Sferrazza, Juyue Chen, Guanya Shi, Rocky Duan, and Pieter Abbeel. Learning sim-to-real humanoid locomotion in 15 minutes. *arXiv preprint arXiv:2512.01996*, 2025.

Zhenguo Sun, Bo-Sheng Huang, Yibo Peng, Xukun Li, Jingyu Ma, Yu Sun, Zhe Li, Haojun Jiang, Biao Gao, Zhenshan Bing, et al. Mosaic: Bridging the sim-to-real gap in generalist humanoid motion tracking and teleoperation with rapid residual adaptation. *arXiv preprint arXiv:2602.08594*, 2026.

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

Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. *arXiv preprint arXiv:2504.11054*, 2025.

Yunshen Wang, Shaohang Zhu, Peiyuan Zhi, Yuhan Li, Jiaxin Li, Yong-Lu Li, Yuchen Xiao, Xingxing Wang, Baoxiong Jia, and Siyuan Huang. Omnixtreme: Breaking the generality barrier in high-dynamic humanoid control. *arXiv preprint arXiv:2602.23843*, 2026a.

Yuxuan Wang, Ming Yang, Ziluo Ding, Yu Zhang, Weishuai Zeng, Xinrun Xu, Haobin Jiang, and Zongqing Lu. From experts to a generalist: Toward general whole-body control for humanoid robots. *arXiv preprint arXiv:2506.12779*, 2025.

Zepeng Wang, Jiangxing Wang, Shiqing Yao, Yu Zhang, Ziluo Ding, Ming Yang, Yuxuan Wang, Haobin Jiang, Chao Ma, Xiaochuan Shi, et al. General humanoid whole-body control via pretraining and fast adaptation. *arXiv preprint arXiv:2602.11929*, 2026b.

Weiji Xie, Jinrui Han, Jiakun Zheng, Huanyu Li, Xinzhe Liu, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills. *arXiv preprint arXiv:2506.12851*, 2025.

Kangning Yin, Weishuai Zeng, Ke Fan, Minyue Dai, Zirui Wang, Qiang Zhang, Zheng Tian, Jingbo Wang, Jiangmiao Pang, and Weinan Zhang. Unitracker: Learning universal whole-body motion tracker for humanoid robots. *arXiv preprint arXiv:2507.07356*, 2025.

Kevin Zakka, Qiayuan Liao, Brent Yi, Louis Le Lay, Koushil Sreenath, and Pieter Abbeel. mjlab: A lightweight framework for gpu-accelerated robot learning. *arXiv preprint arXiv:2601.22074*, 2026.

Yanjie Ze, João Pedro Araújo, Jiajun Wu, and C. Karen Liu. Gmr: General motion retargeting, 2025a. URL `https://github.com/YanjieZe/GMR`. GitHub repository.

Yanjie Ze, Zixuan Chen, Joao Pedro Araújo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. Twist: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025b.

Yanjie Ze, Siheng Zhao, Weizhuo Wang, Angjoo Kanazawa, Rocky Duan, Pieter Abbeel, Guanya Shi, Jiajun Wu, and C Karen Liu. Twist2: Scalable, portable, and holistic humanoid data collection system. *arXiv preprint arXiv:2511.02832*, 2025c.

Weishuai Zeng, Shunlin Lu, Kangning Yin, Xiaojie Niu, Minyue Dai, Jingbo Wang, and Jiangmiao Pang. Behavior foundation model for humanoid robots. *arXiv preprint arXiv:2509.13780*, 2025.

Zhikai Zhang, Haofei Lu, Yunrui Lian, Ziqing Chen, Yun Liu, Chenghuai Lin, Han Xue, Zicheng Zeng, Zekun Qi, Shaolin Zheng, et al. Learning athletic humanoid tennis skills from imperfect human motion data. *arXiv preprint arXiv:2603.12686*, 2026.

A OBSERVATIONS

The actor and critic each receive current robot features, proprioceptive history, and a reference-motion sequence. Superscript g denotes reference quantities; robot quantities omit this superscript. We distinguish the robot's root (pelvis) frame from its torso frame. Base velocities and projected gravity use the root frame, while reference anchor poses and robot key-body poses use the torso frame. The torso serves as the anchor body.

Table 3 summarizes the notation. Orientations use a six-dimensional vector ρ formed from the first two columns of the rotation matrix. The terms $p_{t+\tau}^{g|\mathrm{tor}_t}$ and $\rho_{t+\tau}^{g|\mathrm{tor}_t}$ describe the reference torso pose at offset τ, expressed in the robot's current torso frame. All reference offsets share this frame.

Table 3: Observation notation.

Symbol	Meaning
$q_t, \dot{q}_t$	Robot joint position and velocity
$q_t^g, \dot{q}_t^g$	Reference joint position and velocity
$\bar{q}$	Default robot joint pose
$v_t^{\mathrm{root}}, \omega_t^{\mathrm{root}}$	Base linear and angular velocity in the pelvis frame
g_t^{root}	Unit gravity direction in the pelvis frame
a_{t-1}	Previous policy action
$p_{t+\tau}^{g\|\mathrm{tor}_t}, \rho_{t+\tau}^{g\|\mathrm{tor}_t}$	Reference torso pose in the current robot torso frame
$p_{M,t}^{\mathrm{tor}_t}, \rho_{M,t}^{\mathrm{tor}_t}$	Robot key-body poses in the current torso frame
x_t^π, x_t^V	Actor and critic proprioceptive vectors
$o_{t,\mathrm{hist}}^j, o_{t,\mathrm{ref}}^j$	History and reference sequence, $j \in \{\pi, V\}$
M	Set of 14 tracked key bodies

Proprioception and history. For the 29 actuated joints, we define the actor and critic proprioceptive vectors as

$$x_t^\pi = \left[v_t^{\mathrm{root}},\, \omega_t^{\mathrm{root}},\, g_t^{\mathrm{root}},\, a_{t-1},\, q_t - \bar{q},\, \dot{q}_t\right] \in \mathbb{R}^{96},$$
$$x_t^V = \left[v_t^{\mathrm{root}},\, \omega_t^{\mathrm{root}},\, a_{t-1},\, q_t - \bar{q},\, \dot{q}_t\right] \in \mathbb{R}^{93}.$$

Here $\bar{q}$ is the default joint pose and g_t^{root} is the projected unit gravity direction. Each history contains the nine preceding control steps, ordered from oldest to newest:

$$o_{t,\mathrm{hist}}^j = \left[x_{t-9}^j, \ldots, x_{t-1}^j\right], \qquad j \in \{\pi, V\}.$$

Together with the current vector, this provides ten proprioceptive frames at a control rate of 50 Hz.

Reference sequence. The oracle sequence uses offsets $\mathcal{T}_{\mathrm{ref}} = \{0, 5, 10, \ldots, 45\}$ in control steps. It contains the current reference and nine future references, spaced 0.1 s apart and extending 0.9 s ahead. The actor receives reference joint states and torso poses, while the critic sequence contains reference joint states:

$$o_{t,\mathrm{ref}}^\pi = \left[q_{t+\tau}^g,\, \dot{q}_{t+\tau}^g,\, p_{t+\tau}^{g|\mathrm{tor}_t},\, \rho_{t+\tau}^{g|\mathrm{tor}_t}\right]_{\tau \in \mathcal{T}_{\mathrm{ref}}},$$
$$o_{t,\mathrm{ref}}^V = \left[q_{t+\tau}^g,\, \dot{q}_{t+\tau}^g\right]_{\tau \in \mathcal{T}_{\mathrm{ref}}}.$$

Brackets denote concatenation, with sequence entries ordered by increasing offset.

Actor and critic inputs. The complete observations are

$$o_t^\pi = \left[x_t^\pi,\, o_{t,\mathrm{hist}}^\pi,\, o_{t,\mathrm{ref}}^\pi\right],$$
$$o_t^V = \left[p_t^{g|\mathrm{tor}_t},\, \rho_t^{g|\mathrm{tor}_t},\, x_t^V,\, p_{M,t}^{\mathrm{tor}_t},\, \rho_{M,t}^{\mathrm{tor}_t},\, o_{t,\mathrm{hist}}^V,\, o_{t,\mathrm{ref}}^V\right].$$

The critic additionally receives the current reference torso pose and privileged robot key-body poses. The set M contains 14 key bodies: the pelvis, torso, and bilateral hip-roll, knee, ankle-roll, shoulder-roll, elbow, and wrist-yaw links. Actor current, history, and reference inputs have dimensions 96, 864, and 670; the corresponding critic dimensions are 228, 837, and 580.

Observation noise. During training, we add elementwise uniform noise to the actor features listed in Table 4. Noise is sampled independently at each control step; history retains the noisy measurements recorded at earlier steps. Reference torso position and orientation noise applies to all ten entries of the actor reference sequence. Reference joint states, previous actions, and all critic observations are noise-free. Observations are clipped elementwise to $[-100, 100]$ after noise is applied.

Table 4: Actor observation noise, sampled uniformly and independently per component. Position noise is in meters, linear velocity noise in m/s, joint position noise in radians, and angular and joint velocity noise in rad/s. Orientation noise perturbs the six-dimensional rotation features; projected-gravity noise is dimensionless. Reference torso noise applies at every offset $\tau \in \mathcal{T}_{\mathrm{ref}}$.

Observation term	Additive noise distribution
Reference torso position, $p_{t+\tau}^{g\|\mathrm{tor}_t}$	$\mathcal{U}[-0.25, 0.25]$
Reference torso orientation, $\rho_{t+\tau}^{g\|\mathrm{tor}_t}$	$\mathcal{U}[-0.05, 0.05]$
Base linear velocity, v_t^{root}	$\mathcal{U}[-0.5, 0.5]$
Base angular velocity, ω_t^{root}	$\mathcal{U}[-0.2, 0.2]$
Relative joint position, $q_t - \bar{q}$	$\mathcal{U}[-0.01, 0.01]$
Joint velocity, $\dot{q}_t$	$\mathcal{U}[-0.5, 0.5]$
Projected gravity, g_t^{root}	$\mathcal{U}[-0.05, 0.05]$

B Model Architecture

We use the SONIC architecture (Luo et al., 2025) with a smaller action decoder and critic. Table 5 lists the hidden-layer widths used in our experiments. All MLP hidden layers use SiLU activations.

Table 5: Model architecture. Hidden-layer widths are listed in order.

Module	Hidden-layer widths
Reference encoder	$[2048, 1024, 512, 512]$
Reconstruction decoder	$[2048, 1024, 512, 512]$
Action decoder	$[2048, 2048, 1024, 1024, 512, 512]$
Critic	$[2048, 2048, 1024, 1024, 512, 512]$

C Training

We train with PPO using the hyperparameters in Table 6. Ordinary episodes have a $10\,\mathrm{s}$ time limit. ARA rehearsal episodes are recycled after at least $0.1\,\mathrm{s}$, once the reference phase passes the target region's endpoint, as specified in Eq. 4. Tracking failure and motion completion also end episodes.

Table 6: PPO training hyperparameters. The total environment count varies by experiment. LR denotes learning rate.

Hyperparameter	Value	Hyperparameter	Value
Parallel environments, N	8192 or 16384	Entropy coefficient	0.01
Control frequency	$50\,\mathrm{Hz}$	Value loss coefficient	1.0
Episode horizon, H_{long}	500 steps (10 s)	Actor LR	1×10^{-4}
PPO rollout steps, H_{roll}	24	Critic LR	1×10^{-3}
Learning epochs	5	Actor LR bounds	$[1 \times 10^{-5}, 2 \times 10^{-4}]$
Mini-batches	4	Critic LR bounds	$[1 \times 10^{-5}, 1 \times 10^{-3}]$
Discount factor, γ	0.99	Desired KL	0.01
GAE parameter, λ	0.95	Max gradient norm	1.0
PPO clip parameter	0.2	Initial policy noise std	0.05

Table 7 lists the ARA hyperparameters and shared sampling settings. Both ordinary adaptive sampling and ARA use a backward sampling window: after drawing a reference time within the target bin, we subtract an offset sampled uniformly from $[0, 4]$ s and clip the result at the motion start. This provides an approach to the selected region.

Table 7: ARA hyperparameters and shared sampling settings. The blending factor α varies as specified in Sections 4.2 and 4.3.

Hyperparameter	Value
ARA allocation and recycling	
Rehearsal budget, C_{ARA}	1000 environments
Maximum demand per bin, $D_{\max}$	300 environments
Competence-gate threshold, τ_c	0.99
Minimum eligible failure rate, τ_f	0.5
EMA update rate	0.1
Minimum rehearsal duration	0.1 s
Shared sampling settings	
Bin duration, Δ_{bin}	1 s
Backward sampling window	4 s
Adaptive failure-weight cap multiplier, β	200

D REWARDS

The reward combines motion-tracking terms with penalties for action changes, joint-limit violations, and undesired contacts:

$$r_t = \mathcal{R}_t + \mathcal{P}_t.$$

Here $\mathcal{R}_t$ and $\mathcal{P}_t$ are the weighted sums of tracking and penalty terms in Table 8. The resulting reward is not clipped.

Reward positions, orientations, and velocities are expressed in the world frame. Superscript g denotes reference quantities, while robot quantities omit this superscript. We write R for a rotation matrix and $\Delta_R(A, B)$ for the magnitude of the relative rotation angle between A and B, in radians. For key-body pose rewards, reference poses are aligned using the yaw component of the relative anchor rotation and the robot anchor's horizontal position. This preserves reference height and tilt. We denote the aligned reference poses by $\tilde{p}^g_{i,t}$ and $\tilde{R}^g_{i,t}$.

The soft joint bounds q_i^- and q_i^+ define the central 90% of each joint's permitted range. Joint-limit penalties sum the distances beyond these bounds, with $[x]_+ = \max(x, 0)$. The allowed contact set C_{allowed} contains the bilateral wrist-yaw, ankle-roll, and elbow links. We denote the maximum contact-force magnitude on body c over the four physics substeps of control step t by $F_{c,t}$.

E TERMINATION

During training, an episode terminates when any of the following tracking failures occurs:

- **Anchor height deviation.** The absolute difference between the robot and reference torso heights exceeds 0.25 m.
- **Anchor tilt deviation.** We express the unit gravity vector in the robot and reference torso frames. The absolute difference between the resulting z components exceeds 0.8.
- **Key-body height deviation.** The height of any wrist-yaw or ankle-roll link differs from its aligned reference height by more than 0.25 m.

During evaluation, we use only the anchor-height failure criterion: a trial fails when the torso height deviates from its reference by more than 0.25 m.

Table 8: Reward terms and weights. M is the set of 14 tracked key bodies; tildes denote aligned reference poses. $F_{c,t}$ is the peak contact-force magnitude on body c within control step t. Positions are in meters, linear velocities in m/s, angular velocities in rad/s, joint positions and rotation errors in radians, and forces in newtons.

Reward term	Equation	Weight
Tracking rewards, $\mathcal{R}_t$		
Anchor position	$\exp\left(-\left\|p_{\text{anc},t} - p^g_{\text{anc},t}\right\|_2^2 / 0.3^2\right)$	0.5
Anchor orientation	$\exp\left(-\Delta_R\left(R_{\text{anc},t}, R^g_{\text{anc},t}\right)^2 / 0.4^2\right)$	0.5
Key-body position, aligned	$\exp\left(-\frac{1}{\lvert M\rvert}\sum_{i\in M}\left\|p_{i,t} - \tilde{p}^g_{i,t}\right\|_2^2 / 0.3^2\right)$	1.0
Key-body orientation, aligned	$\exp\left(-\frac{1}{\lvert M\rvert}\sum_{i\in M}\Delta_R\left(R_{i,t}, \tilde{R}^g_{i,t}\right)^2 / 0.4^2\right)$	1.0
Key-body linear velocity	$\exp\left(-\frac{1}{\lvert M\rvert}\sum_{i\in M}\left\|v_{i,t} - v^g_{i,t}\right\|_2^2 / 1.0^2\right)$	1.0
Key-body angular velocity	$\exp\left(-\frac{1}{\lvert M\rvert}\sum_{i\in M}\left\|\omega_{i,t} - \omega^g_{i,t}\right\|_2^2 / 3.14^2\right)$	1.0
Penalty terms, $\mathcal{P}_t$		
Action rate	$\left\|a_t - a_{t-1}\right\|_2^2$	-0.1
Joint position limit	$\sum_i\left([q_i^- - q_{i,t}]_+ + [q_{i,t} - q_i^+]_+\right)$	-10.0
Undesired contacts	$\sum_{c\notin C_{\text{allowed}}}\mathbf{1}\left[F_{c,t} > 1.0\right]$	-0.1

F Domain Randomization

To improve sim-to-real robustness, we randomize physical and control parameters, apply external pushes, and perturb reset states. Table 9 lists the ranges applied during MJLAB training, with $\mathcal{U}[\ell, u]$ denoting a uniform distribution. MJLAB uses a single sliding-friction coefficient, which is sampled from 64 uniformly generated friction values and assigned to each randomized environment at startup. COM offsets and default joint-pose offsets are also sampled at startup, while joint stiffness and damping factors are sampled independently per joint at each reset. Environment zero retains nominal friction and COM values.

External pushes add world-frame root linear and angular velocity increments at sampled intervals. Reset perturbations are applied to the reference state: root position and velocity increments use world coordinates, and orientation perturbations are composed with the reference rotation using sampled roll, pitch, and yaw increments. Initial joint positions are clamped to the soft joint limits after perturbation.

Table 9: Domain randomization applied during MJLAB training. Position offsets are in meters, orientation and joint-position offsets in radians, linear velocity increments in m/s, and angular velocity increments in rad/s. Gain factors are relative to nominal values. The friction range specifies the distribution used to generate the 64 friction values.

Domain randomization	Sampling distribution
Physical and control parameters	
Sliding friction, μ	$\mathcal{U}[0.3, 1.6]$
Default joint position, $\bar{q}$	$\Delta\bar{q} \sim \mathcal{U}[-0.01, 0.01]$
Joint stiffness, K_p	$K_p \cdot \mathcal{U}[0.9, 1.1]$
Joint damping, K_d	$K_d \cdot \mathcal{U}[0.9, 1.1]$
Base COM offset, (x, y, z)	$\Delta x \sim \mathcal{U}[-0.075, 0.075]$, $\Delta y, \Delta z \sim \mathcal{U}[-0.1, 0.1]$
External pushes	
Push interval, T_{push} [s]	$\mathcal{U}[1.0, 3.0]$
Root linear velocity push, (x, y, z)	$\Delta v_x, \Delta v_y \sim \mathcal{U}[-0.5, 0.5]$, $\Delta v_z \sim \mathcal{U}[-0.2, 0.2]$
Root angular velocity push, (x, y, z)	$\Delta\omega_x, \Delta\omega_y \sim \mathcal{U}[-0.52, 0.52]$, $\Delta\omega_z \sim \mathcal{U}[-0.78, 0.78]$
Reset-state perturbations	
Root position, (x, y, z)	$\Delta x, \Delta y \sim \mathcal{U}[-0.05, 0.05]$, $\Delta z \sim \mathcal{U}[-0.01, 0.01]$
Root orientation, $(\text{roll}, \text{pitch}, \text{yaw})$	$\Delta\text{roll}, \Delta\text{pitch} \sim \mathcal{U}[-0.1, 0.1]$, $\Delta\text{yaw} \sim \mathcal{U}[-0.2, 0.2]$
Root linear velocity, (x, y, z)	$\Delta v_x, \Delta v_y \sim \mathcal{U}[-0.5, 0.5]$, $\Delta v_z \sim \mathcal{U}[-0.2, 0.2]$
Root angular velocity, (x, y, z)	$\Delta\omega_x, \Delta\omega_y \sim \mathcal{U}[-0.52, 0.52]$, $\Delta\omega_z \sim \mathcal{U}[-0.78, 0.78]$
Initial joint position, q_{init}	$\Delta q_{\text{init}} \sim \mathcal{U}[-0.1, 0.1]$

G DEPLOYMENT

For deployment, we train the policy directly with PPO, excluding base linear velocity and reference anchor position from its observations to avoid dependence on external state estimation. We export the trained model to ONNX and run inference onboard the robot at 50 Hz.