LUCID: Latent-Understanding Curriculum for Informed Domain Randomization in Humanoid Motion Tracking

Anonymous Authors

***Abstract*— Domain randomization (DR) improves sim-to-real robustness, but severe perturbations can disrupt learning before a humanoid policy stabilizes, especially under actuation latency. We present LUCID (Latent-Understanding Curriculum for Informed Domain Randomization), an execution-informed curriculum that paces perturbation exposure using the temporal discrepancy between issued joint targets and measured positions. A temporal encoder, pretrained by denoising reference motion and then frozen, compares short multi-joint histories in a learned latent space. Its discrepancy warns before failure and is lower for trained policies under the same injected delay. A block-level upper quantile drives a bounded proportional–integral intensity update; sustained low return triggers multiplicative backoff. Across five training seeds, LUCID reaches 88.9% completion under full-range randomization and 73.8% under an unseen 60 ms delay, exceeding filtered-error PI by 12.1 and 21.7 percentage points. Ablations quantify gains from denoising and return backoff; live feedback achieves higher completion than replay of a donor schedule. On a physical Unitree G1 with 40 ms added delay, LUCID completes 38/60 trials versus 23/60 for filtered-error PI, a gain of 25.0 percentage points.**

I. Introduction

Reinforcement learning trains humanoid motion-tracking policies in simulation [1], [2], while domain randomization (DR) varies dynamics and observations [3], [4]. Perturbations tolerated after convergence can disrupt early learning, especially under delay [2]. Robust tracking therefore depends on when exposure increases.

ADR adapts exposure using boundary performance [5], and DORAEMON expands distribution entropy subject to a success constraint [6]. LUCID makes the command–execution relationship a direct curriculum signal: temporal histories of issued targets and measured positions guide perturbation pacing before failure, with task return providing independent backoff.

Instantaneous command error is ambiguous: proportional–derivative (PD) control needs a tracking offset to generate torque, while delay separates issued and applied targets. Error can thus reflect control effort or impaired execution. LUCID compares short multi-joint histories through a temporal encoder pretrained by denoising reference motion and then frozen. Latent understanding denotes the temporal structure learned from reference kinematics and used as a fixed comparison space for execution feedback.

LUCID adjusts a shared DR intensity between training blocks with a bounded proportional–integral (PI) update driven by the upper quantile of the latent discrepancy (Fig. 1). Two consecutive low-return blocks trigger multiplicative backoff when enabled. Deployment uses only the trained policy and low-level controller.

LUCID reaches 88.9% completion under full-range randomization and 73.8% under an unseen 60 ms delay, exceeding filtered-error PI by 12.1 and 21.7 percentage points. At 40 ms added delay, it completes 38/60 G1 trials versus 23/60. Our contributions are (i) a frozen latent-understanding signal that warns before failure and distinguishes policy responses at fixed delay; (ii) a bounded PI/backoff scheduler that converts execution feedback into adaptive DR pacing; and (iii) controlled evaluations of representation, pacing, and live feedback, with MuJoCo and G1 deployment validation.

II. Related Work

Humanoid tracking and transfer. DeepMimic learns motion imitation [1]; BeyondMimic combines compact observations and actuator modeling [2]; SONIC scales tracking models, data, and compute [7]. ASAP refines simulation-trained policies using a delta-action model learned from real trajectories [8]. AdaMimic learns phase and tracking adapters from edited keyframes [9]; RobotDancing learns reference-conditioned residual joint targets for long-horizon tracking [10]. LUCID uses execution feedback to pace DR.

DR curricula. ADR tests parameter boundaries [5]; Active DR selects informative variations [11]; self-paced methods adapt task distributions [12]. DORAEMON maximizes entropy subject to success [6]; GACL represents tasks and monitors performance [13]; VertiSelector uses learning potential and staleness [14]. LUCID represents command–execution histories, retaining return for backoff.

Timing and runtime adaptation. Delay-aware RL accommodates timing [15]; RMA adds runtime adaptation [16]. Our online-delay control uses a similar history interface. LUCID's encoder operates only during training.

Learned comparison spaces. VAEs and denoising learn representations [17], [18]; CAPS regularizes control smoothness [19]. LUCID freezes a learned space for curriculum feedback.

III. Method

LUCID uses command–execution histories to set the next block's randomization intensity.

A. Commanded and Executed Motion

The reference-conditioned policy $\pi_\theta(a_t \mid o_t)$ produces issued joint targets $q_t^{\mathrm{cmd}} \in \mathbb{R}^J$. Its observation includes anchor error and reference phase (Section IV). Delay yields applied targets q_t^{app}, distinct from measured positions q_t^{exec}. The encoder sees only issued commands and measured positions.

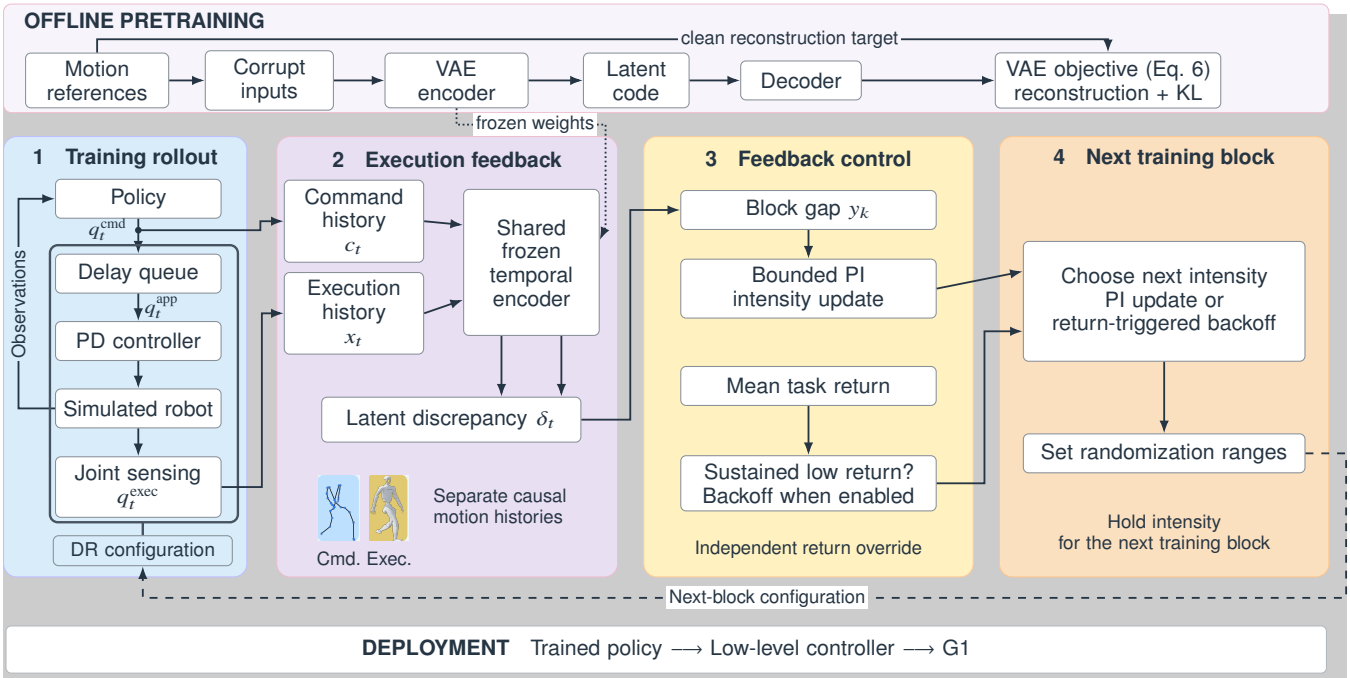

Fig. 1. LUCID training flow. (1) Issued targets and measured joint positions form separate causal histories. (2) A frozen temporal encoder, pretrained by denoising reference motion, supplies their discrepancy in the learned comparison space. (3) The block gap y_k drives a bounded PI update; after activation, two consecutive low-return blocks trigger backoff. (4) The selected intensity sets next-block DR ranges while each channel retains its sampling clock. Deployment uses the trained policy and low-level controller, without the encoder or scheduler.

To interpret command error, consider an ideal, unsaturated PD controller with invertible $\mathbf{K}_p$, zero desired velocity, and no feedforward torque:

$$\tau_t = \mathbf{K}_p(q_t^{\text{app}} - q_t^{\text{exec}}) - \mathbf{K}_d \dot{q}_t^{\text{exec}}. \tag{1}$$

Rearranging gives

$$q_t^{\text{cmd}} - q_t^{\text{exec}} = (q_t^{\text{cmd}} - q_t^{\text{app}}) + \mathbf{K}_p^{-1}\left(\tau_t + \mathbf{K}_d \dot{q}_t^{\text{exec}}\right). \tag{2}$$

Instantaneous error thus combines command mismatch and a torque-generating offset. LUCID compares multi-joint histories in a learned space, calibrates discrepancy against nominal tracking, and retains return as independent backoff feedback.

B. A Shared Randomization Intensity

Let $\boldsymbol{\varphi}$ collect the randomized parameters of the dynamics and observations. For delay perturbations, the effective state also includes the command queue. LUCID holds a shared intensity $\lambda_k \in [0,1]$ fixed within training block k. For a continuous episode-level parameter with nominal value φ_i^0 and full-range bounds $[\ell_i, u_i]$,

$$\xi_i \sim \mathcal{U}(\ell_i, u_i), \qquad \varphi_i = \varphi_i^0 + \lambda_k(\xi_i - \varphi_i^0). \tag{3}$$

Equation (3) interpolates from nominal ($\lambda_k = 0$) to the full range ($\lambda_k = 1$), allowing asymmetric bounds. Ranges are nested when nominal lies within the bounds.

Each channel retains its sampling clock. A 200 Hz FIFO quantizes delay to 5 ms ticks. The 50 Hz policy target is held over four 200 Hz simulation steps. Reachable delays therefore change discretely with λ_k.

The ranges define the perturbation mix; a single λ_k couples channels with potentially different learning dynamics. The signal-swap PI controls share this scalar intensity; Table IV varies which channels it paces, while the hybrid adapts timing and non-timing channels separately.

C. Latent Motion Understanding and Discrepancy

At control time t, we stack H samples with stride s:

$$c_t = [q_{t-(H-1)s}^{\text{cmd}}; \ldots; q_t^{\text{cmd}}], \quad x_t = [q_{t-(H-1)s}^{\text{exec}}; \ldots; q_t^{\text{exec}}], \quad c_t, x_t \in \mathbb{R}^{H\times J}. \tag{4}$$

We use $H = 25$, $s = 1$, and 50 Hz sampling (0.48 s between endpoints). A valid window contains all H samples from one episode, with no reset in its span; issued-command timing is preserved.

A three-layer temporal convolutional network (TCN), with kernel size 5 and width 128, encodes each window into $d_z = 32$ latent coordinates. We pretrain it as the encoder of a temporal variational autoencoder (VAE) [17]. The posterior is a diagonal Gaussian,

$$q_\eta(z \mid w) = \mathcal{N}\Big(\mu_\eta(w), \operatorname{diag}(\sigma_\eta^2(w))\Big). \tag{5}$$

Pretraining corrupts a clean reference-motion window w with joint-position noise ($\sigma = 0.03$ rad) to obtain $\tilde{w}$; the decoder reconstructs w. For $z \sim q_\eta(z \mid \tilde{w})$ and decoder D_ξ, the

objective is

$$\mathcal{L} = \mathbb{E}\big[\|w - D_\xi(z)\|_F^2/(HJ)\big] + \beta D_{\mathrm{KL}}\big(q_\eta(z \mid \widetilde{w}) \,\|\, \mathcal{N}(0, I)\big). \quad (6)$$

Reconstruction averages over HJ coordinates, KL sums over d_z coordinates, and both average over the minibatch; $\beta = 10^{-3}$. Ordinary reconstruction uses $\widetilde{w} = w$. We train on 120,000 reference windows for 50 epochs with Adam (learning rate 10^{-3}, weight decay 10^{-4}, batch size 256). Denoising encourages recovery of multi-joint temporal structure without perturbation labels. This learned structure is LUCID's latent understanding of motion; freezing the encoder fixes the comparison function while the policy changes.

The frozen encoder's posterior mean defines

$$h_t^{\mathrm{c}} = \mu_\eta(c_t), \qquad h_t^{\mathrm{x}} = \mu_\eta(x_t),$$
$$z_t^a = \frac{h_t^a}{\|h_t^a\|_2 + \epsilon}, \qquad a \in \{\mathrm{c}, \mathrm{x}\}, \quad (7)$$
$$\delta_t = 1 - (z_t^{\mathrm{c}})^\top z_t^{\mathrm{x}}.$$

With $\epsilon = 10^{-6}$, the discrepancy lies in $[0, 2]$ for finite features and is approximately scale-invariant when both feature norms greatly exceed ϵ. Sections V-A and V-B evaluate its warning and policy response; the signal does not isolate timing from amplitude or physical cause.

D. PI Scheduling and Return Backoff

Each block collects two 24-step PPO [20] rollouts across 4,096 environments (196,608 transitions) at fixed λ_k. Discrepancy and return then determine the next intensity. For valid windows $\mathcal{V}_k$, the block gap is

$$y_k = \mathrm{Quantile}\big(\{\delta_t \mid t \in \mathcal{V}_k\}, 0.9\big). \quad (8)$$

The upper quantile emphasizes elevated discrepancy. A converged nominal policy ($\lambda = 0$) supplies 500 unperturbed rollouts on the 30 calibration clips to calibrate gap and return references. Nominal block gaps include control offsets and set $r = \mu_{\mathrm{nom}} + 3\sigma_{\mathrm{nom}} = 0.145$.

We normalize the feedback error by the positive nominal reference:

$$e_k = (r - y_k)/r = 1 - y_k/r, \qquad r > 0. \quad (9)$$

Positive error favors increasing intensity; negative error favors decreasing it. The scheduler gains $k_{\mathrm{P}}, k_{\mathrm{I}}$ weight the proportional and integral terms:

$$I_k = \mathrm{clip}(I_{k-1} + e_k, -I_{\max}, I_{\max}),$$
$$u_k = \mathrm{clip}(k_{\mathrm{P}} e_k + k_{\mathrm{I}} I_k, -1, 1),$$
$$\lambda_{k+1} = \mathrm{clip}(\lambda_k + \alpha u_k, 0, 1). \quad (10)$$

The scheduler uses $k_{\mathrm{P}} = 0.8$, $k_{\mathrm{I}} = 0.15$, $\alpha = 0.04$, and $I_{\max} = 0.8$.

Clamping bounds the integral state and proposed increment. Normalization makes e_k invariant to joint positive rescaling of the signal and reference.

The same calibration rollouts fix $\bar{R}_{\mathrm{nom}}$ before training. Warm-up suppresses the backoff counter until the end of the first block satisfying $k \geq 15$ (zero-based) or $\bar{R}_k \geq 0.50\bar{R}_{\mathrm{nom}}$. Warm-up then stays off for the rest of training. Backoff can begin counting in that block (Algorithm 1), where w_k denotes active warm-up. An empty window set holds intensity and integral state unless return backoff takes priority.

Algorithm 1 LUCID curriculum block update

Require: Frozen encoder μ_η; reference $r > 0$; gains and bounds; return threshold $R_{\min}$; backoff factor γ_{DR}

Initialize $\lambda_0 = 0$, $I_{-1} = b_{-1} = 0$, $w_0 = 1$
for curriculum block $k = 0, 1, \ldots, K-1$ **do**
  Train PPO at fixed λ_k; log commands, measurements, and returns
  Construct valid within-episode windows $\mathcal{V}_k$
  $w_{k+1} \leftarrow w_k$
  if $w_k = 1$ and ($k \geq 15$ or $\bar{R}_k \geq 0.50\bar{R}_{\mathrm{nom}}$) **then**
    $w_{k+1} \leftarrow 0$ ▷ Latch warm-up off
  end if
  if $w_{k+1} = 0$ and $\bar{R}_k < R_{\min}$ **then**
    $b_k \leftarrow b_{k-1} + 1$
  else
    $b_k \leftarrow 0$
  end if
  if $b_k \geq 2$ **then**
    Apply Eq. (11); set $b_k \leftarrow 0$
  else if $\mathcal{V}_k = \emptyset$ **then**
    $\lambda_{k+1} \leftarrow \lambda_k$, $I_k \leftarrow I_{k-1}$
  else
    Compute y_k with Eq. (8); normalize by Eq. (9)
    Update I_k, u_k, λ_{k+1} with Eq. (10)
  end if
end for

Once backoff is enabled, two consecutive blocks with mean episodic return below $R_{\min} = 0.65\bar{R}_{\mathrm{nom}}$ trigger an integral reset and set

$$\lambda_{k+1} = \gamma_{\mathrm{DR}}\lambda_k, \qquad I_k = 0, \quad 0 < \gamma_{\mathrm{DR}} < 1. \quad (11)$$

With $\gamma_{\mathrm{DR}} = 0.70$, this return-triggered reduction overrides the gap-based PI proposal. Section V-E tests their contributions.

IV. Experimental Setup

A. Robot, Observations, and Low-Level Control

Table I specifies the full DR ranges. We train in Isaac Lab using a Unitree G1 with 23 active joints: 12 in the legs, three in the waist, and eight in the arms; the hands are locked. The policy runs at 50 Hz; physics and the actuator queue run at 200 Hz (decimation four). Training delays use zero to eight 5 ms FIFO slots. Training episodes end on failure or after 100 control steps (2 s); resets flush the queue and prefill it with q_0. The actor receives an 87-dimensional observation:

$$o_t = [q_t - q_0, \dot{q}_t, a_{t-1}, \omega_t, \mathbf{g}_t, e_{\mathrm{anchor},t}, \psi_t]. \quad (12)$$

Dimensions are $23 + 23 + 23 + 3 + 3 + 9 + 3 = 87$. Angular velocity and gravity use the pelvis frame; anchor error contains translation and six-coordinate relative rotation; $\psi_t = [\sin(2\pi\phi_t), \cos(2\pi\phi_t), \dot{\phi}_t]$.

Equation (12) omits full joint references; these enter the offline task-error diagnostic and dense-task-error PI training control.

TABLE I

DOMAIN RANDOMIZATION RANGES.

Parameter	Full range ($\lambda = 1$)
Actuation and dynamics	
FIFO delay (ms)	$\mathcal{U}(0, 40)$; 5 ms ticks
Static friction (nominal 1.0)	$\mathcal{U}(0.4, 1.6)$
Dynamic friction (nominal 0.8)	$\mathcal{U}(0.3, 1.3)$
Restitution	$\mathcal{U}(0, 0.5)$
Base mass offset (kg)	$\mathcal{U}(-2.5, 2.5)$
CoM x, y offset (m)	$\mathcal{U}(-0.05, 0.05)$
CoM z offset (m)	$\mathcal{U}(-0.025, 0.025)$
Joint offset (rad)	$\mathcal{U}(-0.01, 0.01)$
Ankle offset (rad)	$\mathcal{U}(-0.10, 0.10)$
External velocity perturbations (pushes)	
$v_{x,y}$ (m/s)	$\mathcal{U}(-0.50, 0.50)$
v_z (m/s)	$\mathcal{U}(-0.20, 0.20)$
Zero-mean Gaussian observation noise	
Joint position (rad)	$\sigma_{\text{full}} = 0.01$
Joint velocity (rad/s)	$\sigma_{\text{full}} = 0.50$
Pelvis linear vel. (m/s)	$\sigma_{\text{full}} = 0.50$
Pelvis angular vel. (rad/s)	$\sigma_{\text{full}} = 0.20$

Nominals are zero except friction; zero restitution is inelastic. Uniform draws follow Eq. (3). Noise is drawn at 50 Hz with $\sigma = \lambda\sigma_{\text{full}}$, then clipped at $\pm 3\sigma$. Pelvis pushes last 0.15 s (30 physics steps), every 1–3 s.

We select joint gains with the reflected-inertia heuristic, using design frequency $f_n = 10$ Hz, damping ratio $\zeta = 2.0$, and $\omega_n = 2\pi f_n$. For reflected joint inertia $\mathcal{J}_j$,

$$K_{p,j} = \mathcal{J}_j \omega_n^2, \qquad K_{d,j} = 2\mathcal{J}_j \zeta \omega_n. \tag{13}$$

The resulting design angular frequency is $\omega_n \approx 62.83$ rad/s. The armature-referenced gains follow the actuator-specific convention in BeyondMimic [2].

B. Training Data, Budget, and Comparators

The reference set contains 180 retargeted LaFAN1 [21] and CMU MoCap [22] clips: 120 for training, 30 for calibration, and 30 held out for evaluation. The common benchmark uses five training seeds per method (LUCID: seeds 1–5) and 1,000 shared evaluation scenarios per seed. Its 150 blocks collect 29,491,200 transitions (Table III: rounded to 3.0×10^7); extended fixed DR doubles this budget. PPO uses learning rate 2×10^{-4} with linear decay, clipping 0.2, GAE 0.95, and discount 0.99.

Baselines include minimal DR (BeyondMimic recipe), fixed DR, tuned-constant DR, linear ramping, scalar return feedback, ADR, and DORAEMON (Table III). Constant DR selects $\lambda = 0.65$ from $\{0.15, 0.3, 0.5, 0.65, 0.8, 1.0\}$. In the 150-block benchmark runs, the linear ramp reaches full intensity at block 120 and holds it for 30 blocks. Scalar return feedback changes intensity by $+0.04$ above $0.85\bar{R}_{\text{nom}}$ and -0.04 below $0.65\bar{R}_{\text{nom}}$. PI controls use dense task error, raw command error, causal 4 Hz filtered error, or cross-correlation lag (CC-Lag). The task-error controller uses $\|q_t^{\text{ref}} - q_t^{\text{exec}}\|$ rather than issued commands. CC-Lag averages joint-wise estimated lags before block-quantile aggregation. Raw, CC-Lag, filtered, and latent PI controls share return backoff and clamping bounds. Their references share nominal rollouts. Signal swaps test representations; ADR/DORAEMON comparisons test complete methods. Guard-only ramping retains return backoff with prescribed intensity increases; P-only removes the integral term ($k_{\text{I}} = 0$).

Allocation crosses linear/LUCID pacing with all channels, delay only (dynamics/noise at full range), or dynamics/noise only (delay fixed at 40 ms). Online delay adaptation augments the common actor with a three-layer 1D-CNN, a 16-dimensional adaptation latent, and 50 steps of observation–action history; it is trained jointly under the linear ramp. The hybrid combines LUCID delay pacing with DORAEMON non-timing adaptation. Offline controls use 32-dimensional PCA or random temporal features.

Evaluation freezes the curriculum and runs full held-out clips from time zero (mean 3.8 s, range 2.5–6.0 s; 125–300 steps). Completion requires reaching the clip end without termination. Nominal tests use $\lambda = 0$; full-range tests use $\lambda = 1$. The 60 ms test holds dynamics nominal and fixes delay at 12 FIFO ticks, beyond the eight-tick training maximum. Non-timing tests use full-range dynamics/noise and zero added delay.

Ordered replay transfers LUCID seed 1's intensities $\lambda_{0:149}$ to five recipient runs (seeds 2–6); shuffled replay permutes these values before each run. Both preserve the donor multiset. Recipients generate their own environment draws and visited states.

C. Outcome Populations and Uncertainty

Completion uses all attempts. Simulation fails if reference-relative root or hand/foot height error exceeds 0.25 m, or root tilt exceeds 0.8 rad. Deployment uses Section VI's watchdog. Training collapse denotes a decline of more than 25% from pre-perturbation validation completion for three consecutive evaluation checkpoints.

MPJPE-L is local root-relative positional error on 14 key links, in millimeters; E_{vel} denotes the per-frame motion-difference error (mm/frame). Quality metrics use each policy's completed full-range episodes, so the compared subsets can differ. Action HF power is the fraction of joint-target spectral power above 12.5 Hz, within the 25 Hz Nyquist limit of the 50 Hz signal.

Simulation completion and MPJPE-L intervals (Tables III–VI) use 10,000 two-level resamples: five seeds with replacement, then 1,000 episodes independently within each seed; MPJPE-L uses completed episodes. Symmetric entries report mean $\pm(q_{0.975} - q_{0.025})/2$; brackets give those percentile endpoints. Intervals are marginal, without cross-seed scenario pairing. Warning intervals resample 30 motion clusters 10,000 times, retaining all cluster rollouts. Deployment intervals use 10,000 checkpoint–motion cluster resamples, retaining paired trials, sessions, and nominal/delay conditions. Comparisons with the smaller minimal-DR hardware series are descriptive point contrasts. Fixed-delay diagnostics report the median discrepancy across valid windows in 1,000 rollouts per condition, distinct from the scheduler's block quantile. Bold marks best point estimates within each stated comparison, including ties, not statistical significance; shading identifies scalar LUCID.

TABLE II

FAILURE-WARNING PERFORMANCE.

Signal	AP (%) ↑	Sensitivity (%) ↑	Lead time (s) ↑	Alerts/ safe min ↓
Direct / filtered / lag feedback				
Raw discrepancy	34.2 ± 1.8	42.5 ± 2.3	0.38 ± 0.04	12.4
Filtered-error	55.4 ± 1.4	63.8 ± 1.9	0.53 ± 0.03	5.4
CC-Lag	44.8 ± 1.7	51.3 ± 2.2	0.44 ± 0.04	9.2
Task and feature feedback				
Task error	49.6 ± 1.5	59.2 ± 2.0	0.45 ± 0.03	7.8
PCA windows	47.1 ± 1.6	54.8 ± 2.1	0.42 ± 0.03	8.5
Random TCN	40.5 ± 1.9	46.7 ± 2.4	0.41 ± 0.04	10.9
Latent (LUCID)	**69.8** ± 1.2	**79.5** ± 1.5	**0.69** ± 0.02	**2.6**

2,000 held-out rollouts across 30 motions. Thresholds: 5% validation FPR; event horizon: 1 s. ±: 95% motion-cluster bootstrap half-widths. AP: average precision. Bold: best point estimates; alerts are point estimates.

Deployment contrasts appear in Fig. 3; Table VII gives success rates and trial counts.

V. Signal Diagnostics and Policy Evaluation

A. The Signal Provides Pre-Failure Information

We evaluate a policy trained at $\lambda = 0$ on 2,000 two-second rollouts across 30 held-out motions, sampling evaluation intensity uniformly from $[0, 1]$.

All Table II quantities use valid within-episode windows (Section III-C); no post-termination values enter any metric or threshold calibration. An alert within $[t_{\text{fail}} - 1.0, t_{\text{fail}}]$ seconds detects a failure event; event sensitivity is the detected fraction. Thresholds are calibrated on the separate 30 calibration motions and locked at 5.0% false-positive rate. False alerts are normalized by safe operating time: successful trajectories plus failing trajectories' segments before $t_{\text{fail}} - 1$ s.

LUCID reaches 69.8% average precision (AP), compared with 55.4% for filtered error and 44.8% for CC-Lag (Table II). Compared with filtering, latent feedback raises event sensitivity from 63.8% to 79.5% and reduces false alerts from 5.4 to 2.6 per safe minute. PCA windows reach 47.1% AP and a random TCN 40.5%: gaps of 22.7 and 29.3 points favor the learned representation over dimensionality reduction or windowing alone. We next evaluate how this feedback changes curriculum pacing.

With independent Gaussian joint-position noise sampled at 50 Hz ($\sigma = 0.03$ rad), latent feedback has the smallest false-alert increase: 0.6 per safe minute, versus 2.9 for filtering and 8.2 for raw error. The resulting rates are 3.2, 8.3, and 20.6 per safe minute.

B. Discrepancy Depends on the Policy Response

At a fixed 40 ms injected delay, the nominal, final filtered-error PI, and final LUCID policies have median latent discrepancies of 0.645, 0.412, and 0.215 (Fig. 2(b)). Latent discrepancy thus captures the policy's execution response at a fixed perturbation level.

The same nominal-policy diagnostic uses 1,000 rollouts to probe delay, payload, friction, and noise. Adding 20 ms delay raises the nominal median discrepancy from 0.082 to 0.364; adding 2 kg raises it to 0.104. Combining delay and payload gives 0.388. At 40 ms delay, low friction and noise raise the median from 0.645 to 0.672. These medians differ from the scheduler's block quantile (Eq. (8)).

C. Final-Policy Robustness

With fixed test distributions and disabled curriculum updates, LUCID achieves 88.9% full-range and 73.8% unseen-delay completion (Table III), exceeding filtered-error PI by 12.1 and 21.7 percentage points and dense-task-error PI by 15.8 and 27.6 points. From nominal to 60 ms, completion drops 22.6 points for LUCID, 43.3 for filtered-error PI, and 68.6 for minimal DR.

Under unseen 60 ms delay, scalar LUCID exceeds DORAEMON by 19.2 percentage points (73.8% versus 54.6%). DORAEMON has a higher non-timing mean (83.1% versus 82.0%). Combining LUCID delay pacing with DORAEMON non-timing adaptation yields the highest tested means: 89.5% full-range, 74.6% unseen-delay, and 84.2% non-timing completion. This hybrid demonstrates the benefit of combining execution and outcome feedback. Minimal DR leads nominal completion (96.8%) and conditional positional error (28.4 mm), while completing 28.2% of the 60 ms delay trials.

At half the policy-training budget, LUCID exceeds doubled-budget fixed DR by 25.5 full-range and 39.0 unseen-delay completion points.

Among each method's successful full-range episodes, LUCID has lower positional error (31.5 versus 34.2 mm), motion-difference error (3.8 versus 4.3 mm/frame), and action HF ratio (0.081 versus 0.092) than filtered-error PI, alongside higher completion.

D. Feedback Within Matched Channel Allocations

With only delay paced and other channels at full range, LUCID improves full-range completion over linear ramping from 74.8% to 85.4% (Table IV). Fig. 2(a) compares all three matched allocations.

Dynamics-only pacing improves non-timing completion from 74.2% to 80.4% (+6.2 points), whereas its full-range difference is +2.6 points. Pacing all channels adds 3.5 full-range and 5.2 non-timing points over delay-only LUCID. LUCID achieves higher mean full-range completion across all three matched allocations, comparing the complete PI/backoff rule with linear pacing on the same channels.

E. Ablations and Schedule Replay

Table V tests full-range robustness, unseen delay, and stability. Removing return backoff reduces full-range completion from 88.9% to 72.1% and 60 ms completion from 73.8% to 50.4% (−23.4 points), with 2/5 collapses. Removing integral feedback costs 8.1 full-range and 12.3 delay points. Denoising adds 6.4 full-range and 9.0 delay points over ordinary reconstruction. Guard-only ramping reaches 74.0% and 48.2%, showing that return protection alone does not recover latent PI pacing.

Across five recipients of one donor schedule, ordered replay exceeds shuffled replay by 26.0 full-range and 33.4 delay points; live LUCID exceeds ordered replay by 10.5 and 15.6 points. For full-range completion, the ordered-replay interval

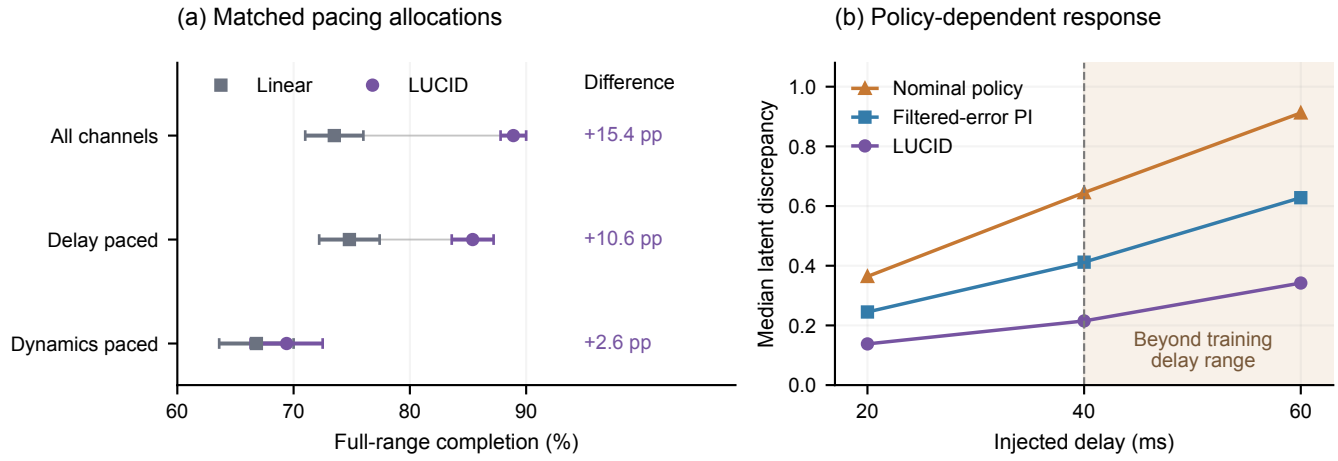

Fig. 2. Pacing and policy-dependent execution response. (a) Full-range completion under matched allocations (Table IV); unpaced delay is fixed at 40 ms, and unpaced dynamics/noise use full ranges. Bars show 95% bootstrap intervals for individual means; labels give LUCID–linear point differences. (b) Median latent discrepancy over valid windows in 1,000 rollouts per condition at 20, 40, and 60 ms added delay. Shading marks delays beyond the 40 ms training maximum.

TABLE III

POLICY ROBUSTNESS AND TRACKING ACCURACY.

Method	Budget (10^7 trans.)	Nominal % ↑	Full range % ↑	60-ms added delay (%) ↑	Δ_{60} (pp) ↓	Non-timing % ↑	MPJPE-L (mm) ↓
Non-adaptive baselines							
Minimal DR (BeyondMimic-style) [2]	3.0	**96.8** ± 0.6	68.4 ± 2.9	28.2 ± 3.8	68.6	70.5 ± 2.6	**28.4** ± 0.9
Fixed full-range DR	3.0	89.4 ± 2.2	62.1 ± 3.6	33.5 ± 4.2	55.9	69.2 ± 3.1	46.8 ± 2.2
Fixed full-range DR (2× budget)	6.0	89.8 ± 2.0	63.4 ± 3.4	34.8 ± 3.9	55.0	70.1 ± 2.9	45.2 ± 2.1
Tuned constant DR	3.0	94.2 ± 1.3	71.4 ± 2.7	42.0 ± 3.5	52.2	72.8 ± 2.5	38.2 ± 1.7
Linear ramp	3.0	93.0 ± 1.5	73.5 ± 2.5	45.2 ± 3.1	47.8	73.5 ± 2.4	37.0 ± 1.5
Feedback and curriculum baselines							
Scalar return feedback	3.0	94.1 ± 1.2	74.2 ± 2.2	47.8 ± 2.9	46.3	74.8 ± 2.2	35.8 ± 1.4
Canonical ADR [5]	3.0	92.8 ± 1.7	75.1 ± 2.4	49.5 ± 3.0	43.3	76.4 ± 2.1	35.2 ± 1.4
DORAEMON [6]	3.0	95.0 ± 1.0	79.8 ± 1.8	54.6 ± 2.6	40.4	83.1 ± 1.7	32.8 ± 1.1
Execution-informed pacing (signal-swap PI controls)							
Dense task error + PI	3.0	93.8 ± 1.4	73.1 ± 2.4	46.2 ± 3.1	47.6	74.0 ± 2.2	36.4 ± 1.6
Raw discrepancy + PI	3.0	91.2 ± 1.6	65.4 ± 3.4	41.8 ± 3.8	49.4	68.2 ± 3.0	44.2 ± 2.0
CC-Lag + PI	3.0	93.6 ± 1.4	71.2 ± 2.8	56.4 ± 3.2	37.2	72.5 ± 2.6	37.8 ± 1.6
Filtered-error PI	3.0	95.4 ± 0.9	76.8 ± 2.0	52.1 ± 2.8	43.3	74.2 ± 1.9	34.2 ± 1.2
LUCID (scalar)	3.0	96.4 ± 0.7	88.9 ± 1.1	73.8 ± 2.1	22.6	82.0 ± 1.6	31.5 ± 1.0
Architectural and curriculum extensions							
Online delay adaptation (RMA-style)*	3.0	94.2 ± 1.2	81.4 ± 2.0	58.2 ± 2.9	36.0	79.5 ± 1.9	33.1 ± 1.3
Hybrid (LUCID + DORAEMON)	3.0	96.6 ± 0.6	**89.5** ± 1.1	**74.6** ± 1.8	**22.0**	**84.2** ± 1.4	30.8 ± 0.9

Bold: best column means, including ties. Five seeds, 1,000 shared scenarios/seed. Means ± 95% bootstrap half-widths. Completion uses all attempts; MPJPE-L uses completed full-range episodes (Section IV-C). Δ_{60}: nominal minus 60-ms completion (pp). *Online adaptation adds 2.4 ms of inference; other methods need no auxiliary deployment module.

[65.0, 85.7]% lies below the live LUCID mean (88.9%) and above the shuffled-replay interval [46.5, 61.8]%.

F. Sensitivity and Training Variation

Table VI reports local sweeps from the shared default. Median aggregation yields 76.4% completion and one collapse, versus 88.9% and zero at $p = 0.90$. Changing r affects both the target and normalized feedback gain (Eq. (9)). At $\alpha = 0.08$, completion is 83.5% (seed SD 8.58 points), with one collapsed seed at 68.2%; the other four score 86.4–88.0%. The larger update has substantially greater seed variation. The five full-range LUCID seed scores are 89.4, 88.1, 89.8, 87.9, and 89.3% (mean 88.9%, sample SD 0.85 points). Runs average 1.4 backoffs and first reach full intensity at block 104 of 150, versus block 120 for the linear ramp; observed collapses are 0/5 versus 1/5. LUCID achieves higher completion and reaches full intensity earlier than linear ramping.

Computational cost. Encoder pretraining takes 0.4 h (RTX 4090); policy training takes 4.8 h for LUCID versus 4.5 h for filtered-error PI (+6.7%). Encoder rollout overhead is 2.1%. At deployment, online adaptation adds 2.4 ms of estimator inference; LUCID adds none (Table III).

TABLE IV

EFFECT OF PACING DIFFERENT CHANNELS.

Allocation	Rule	Full range (%) ↑	60-ms delay (%) ↑	Non-timing (%) ↑	Collapsed runs ↓
All	Linear	73.5 ± 2.5	45.2 ± 3.1	73.5 ± 2.4	1/5
All	LUCID	**88.9** ± 1.1	**73.8** ± 2.1	**82.0** ± 1.6	**0/5**
Delay	Linear	74.8 ± 2.6	48.6 ± 3.4	72.1 ± 2.5	1/5
Delay	LUCID	**85.4** ± 1.8	**72.1** ± 2.4	**76.8** ± 2.0	**0/5**
Dynamics	Linear	66.8 ± 3.2	35.8 ± 4.0	74.2 ± 2.2	2/5
Dynamics	LUCID	**69.4** ± 3.1	**36.2** ± 3.8	**80.4** ± 1.8	**1/5**

Bold: best within each allocation. Five seeds; mean ± 95% bootstrap half-width. Unpaced delay: 40 ms; unpaced dynamics/noise: full ranges. Collapse definition: Section IV-C.

TABLE V

COMPONENT ABLATIONS AND SCHEDULE REPLAY.

Variant	Full range (%) ↑	60-ms delay (%) ↑	Collapsed runs ↓	Time (h) ↓
Full LUCID	**88.9** ± 1.1	**73.8** ± 2.1	**0/5**	4.8
No return backoff	72.1 ± 4.7	50.4 ± 4.6	2/5	4.7
Guard-only ramp	74.0 ± 3.8	48.2 ± 4.1	1/5	**4.5**
P-only ($k_I = 0$)	80.8 ± 3.3	61.5 ± 4.3	**0/5**	4.7
Ordinary reconstruction	82.5 ± 3.3	64.8 ± 4.3	**0/5**	4.8
Donor replay: mean [95% CI]				
Ordered replay	78.4 [65.0, 85.7]	58.2 [44.6, 66.0]	1/5	**4.5**
Shuffled replay	52.4 [46.5, 61.8]	24.8 [19.7, 33.4]	4/5	4.6

Bold: best columns, including ties. Five seeds, 1,000 scenarios/seed. Means ± 95% bootstrap half-widths; replay shows percentile intervals (donor 1, recipients 2–6). Time excludes encoder pretraining.

TABLE VI

SENSITIVITY TO CURRICULUM PARAMETERS.

Parameter	Setting	Completion (%) ↑ Mean [95% CI]	Collapsed runs ↓
Default	—	**88.9** [87.8, 90.0]	**0/5**
Quantile	$p = 0.50$	76.4 [72.8, 80.0]	1/5
	$p = 0.98$	85.1 [81.8, 88.4]	**0/5**
Reference	$r = 0.116$	82.3 [78.9, 85.7]	**0/5**
	$r = 0.174$	86.4 [83.2, 89.6]	**0/5**
Window	$H = 10$	81.8 [78.2, 85.4]	**0/5**
	$H = 50$	87.1 [84.0, 90.2]	**0/5**
PI gains and bounds (other parameters at default)			
k_P	0.4	85.6 [84.4, 86.8]	**0/5**
	1.2	87.2 [86.0, 88.4]	**0/5**
k_I	0.05	84.1 [82.8, 85.4]	**0/5**
	0.25	86.8 [85.5, 88.0]	**0/5**
α	0.02	86.2 [84.9, 87.4]	**0/5**
	0.08	83.5 [75.6, 88.1]	1/5
I_{max}	0.4	87.1 [85.9, 88.2]	**0/5**
	1.2	88.2 [87.0, 89.4]	**0/5**

Bold: best values, including ties. One parameter varies; five seeds, 1,000 scenarios/seed. Mean [95% percentile interval]; 10,000 two-level bootstrap resamples (Section IV-C).

TABLE VII

DEPLOYMENT SUCCESS RATES.

Domain / condition	Success rate (%) ↑; completed/total		
	Minimal DR	Filtered-error PI	LUCID
MuJoCo nominal	—	91.7 (55/60)	**96.7** (58/60)
MuJoCo +20 ms	—	71.7 (43/60)	**88.3** (53/60)
MuJoCo +40 ms	—	46.7 (28/60)	**73.3** (44/60)
G1 nominal	**95.0** (19/20)	85.0 (51/60)	90.0 (54/60)
G1 +20 ms	45.0 (9/20)	65.0 (39/60)	**81.7** (49/60)
G1 +40 ms	15.0 (3/20)	38.3 (23/60)	**63.3** (38/60)
G1 +2.0 kg	65.0 (13/20)	70.0 (42/60)	**80.0** (48/60)

Bold: highest rate per condition. Minimal DR is a separate 20-trial G1 series; —: not evaluated. Paired 95% intervals: Section VI and Fig. 3.

VI. CROSS-SIMULATOR AND PHYSICAL DEPLOYMENT

A. Trial Design, Completion, and Timing

Each filtered-error PI/LUCID condition combines four full-length motions (walking, 5.0 s; turning, 4.0 s; squatting, 4.5 s; side-stepping, 4.0 s), three independent checkpoints, and five repetitions (60 trials). With 20 minimal-DR G1 trials per condition, evaluation totals 360 simulated and 560 physical executions.

MuJoCo 3.1.2 uses a G1 URDF converted to MJCF, preserving joint armature, damping, policy weights, and the 200 Hz actuator queue.

MuJoCo and G1 trials start from a static zero-phase stance and succeed only after the full clip without a stop. Tracked-link deviation above 0.25 m or a joint-velocity-limit violation is a tracking stop. Tilt above 0.8 rad, a handler tether trigger, or fall-watchdog activation is a balance stop. Handlers are blind to randomized labels; manual hazard overrides remain available. Both stop types count as unsuccessful trials (Table VII).

The Jetson Orin publishes commands at 500 Hz. PTP sensor-to-command-dispatch latency summaries are 18.4 ms nominally (including 1.8 ms policy inference), 38.4 ms with +20 ms FIFO delay, and 58.4 ms with +40 ms. These exclude mechanical response. Nominal means zero added delay; no encoder or scheduler runs onboard.

B. Transfer Under Added Delay and Payload

In MuJoCo, the paired LUCID–filtered completion difference rises from 5.0 points [−1.5, 11.5] nominally to 16.7 [5.8, 27.6] at +20 ms and 26.7 [13.2, 40.2] at +40 ms. At +40 ms, the 16 additional completions correspond to ten fewer tracking stops and six fewer balance stops.

On G1 with 40 ms added delay, LUCID completes 38/60 trials versus 23/60 for filtered-error PI, a paired difference of 25.0 points [11.8, 38.2]. Nominally, the difference is 5.0 points [−2.1, 12.1]; at +20 ms, it is 16.7 [5.2, 28.2]. Fig. 3 summarizes these paired contrasts. The separate minimal-DR series has a higher nominal completion fraction (19/20, 95%) than LUCID (54/60, 90%), but completes 3/20 at +40 ms.

For G1 completion estimates $\widehat{p}_L(d)$ and $\widehat{p}_F(d)$ at added delay d, the interaction contrast subtracts the nominal method difference:

$$\mathcal{I}(d) = [\widehat{p}_L(d) - \widehat{p}_F(d)] - [\widehat{p}_L(0) - \widehat{p}_F(0)]. \quad (14)$$

The LUCID–filtered effect increases relative to nominal

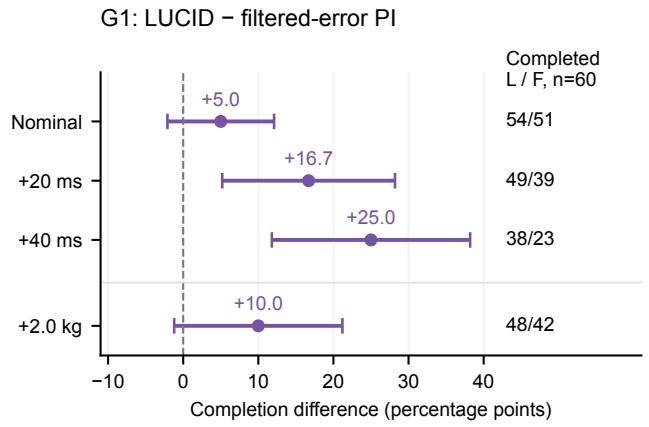

Fig. 3. Physical G1 completion differences for LUCID minus filtered-error PI, with paired 95% intervals. Each method has 60 trials per condition; L/F completion counts appear beside the estimates. Table VII gives success rates and the separate minimal-DR series.

timing by 11.7 points [1.4, 22.0] at +20 ms and 20.0 [7.6, 32.4] at +40 ms.

With an added 2.0 kg payload, within the ±2.5 kg training range, LUCID completes 48/60 trials versus 42/60 for filtered-error PI (+10.0 percentage points; paired 95% interval [−1.2, 21.2]).

VII. Discussion

LUCID turns command–execution histories into effective feedback for perturbation pacing. Its frozen representation distinguishes policy responses at fixed delay (Fig. 2(b)). Below-reference discrepancy contributes a positive proportional term; the integral term incorporates past errors, and sustained low return triggers backoff. Matched allocations show higher mean completion with LUCID across three channel sets; ablations quantify the contributions of denoising, integral feedback, and return protection. Live feedback also exceeds replay from a single donor schedule across five recipient runs.

The hybrid's leading completion means show how execution-informed timing feedback complements non-timing distribution adaptation. The position-controlled G1 results under constant injected delay motivate per-channel pacing, magnitude-sensitive temporal features, and stochastic delay models. Transferring the robot-specific encoder to other platforms and torque-controlled policies offers a concrete test of representation transfer.

VIII. Conclusion

LUCID establishes command–execution discrepancy as an effective curriculum signal for humanoid motion tracking. A frozen, denoising-pretrained temporal encoder drives bounded PI adaptation, with task return providing backoff. Controlled comparisons demonstrate improved robustness from execution-informed pacing, reaching 88.9% full-range completion and 73.8% under an unseen 60 ms delay. On G1 with 40 ms added delay, LUCID completes 38/60 trials versus 23/60 for filtered-error PI. These gains transfer to hardware using the trained policy and low-level controller, with all auxiliary inference confined to training.

References

[1] 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.

[2] Q. Liao *et al.*, "BeyondMimic: From motion tracking to versatile humanoid control via guided diffusion," *arXiv preprint arXiv:2508.08241v4*, 2025.

[3] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 3803–3810.

[4] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, "Robot learning from randomized simulations: A review," *arXiv preprint arXiv:2111.00956*, 2021.

[5] OpenAI, I. Akkaya, M. Andrychowicz *et al.*, "Solving Rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[6] G. Tiboni, P. Klink, J. Peters, T. Tommasi, C. D'Eramo, and G. Chalvatzaki, "Domain randomization via entropy maximization," in *International Conference on Learning Representations*, 2024.

[7] Z. Luo, Y. Yuan, T. Wang *et al.*, "SONIC: Supersizing motion tracking for natural humanoid whole-body control," *Science Robotics*, vol. 11, no. 117, p. eaed4592, 2026.

[8] T. He *et al.*, "ASAP: Aligning simulation and real-world physics for learning agile humanoid whole-body skills," *arXiv preprint arXiv:2502.01143v3*, 2025.

[9] T. Huang *et al.*, "Towards adaptable humanoid control via adaptive motion tracking," *arXiv preprint arXiv:2510.14454*, 2025.

[10] Z. Sun *et al.*, "RobotDancing: Residual-action reinforcement learning enables robust long-horizon humanoid motion tracking," *arXiv preprint arXiv:2509.20717*, 2025.

[11] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active domain randomization," in *Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 100, 2020, pp. 1162–1176.

[12] P. Klink, H. Abdulsamad, B. Belousov, and J. Peters, "Self-paced contextual reinforcement learning," in *Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 100, 2020, pp. 513–529.

[13] L. Wang, Z. Xu, P. Stone, and X. Xiao, "GACL: Grounded adaptive curriculum learning with active task and performance monitoring," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025, pp. 591–596.

[14] T. Xu, C. Pan, and X. Xiao, "VertiSelector: Automatic curriculum learning for wheeled mobility on vertically challenging terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025, pp. 11 136–11 143.

[15] Y. Bouteiller, S. Ramstedt, G. Beltrame, C. Pal, and J. Binas, "Reinforcement learning with random delays," in *International Conference on Learning Representations*, 2021.

[16] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems*, 2021.

[17] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[18] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[19] S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko, "Regularizing action policies for smooth control with reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 1810–1816.

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[21] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 60:1–60:12, 2020.

[22] CMU Graphics Lab, "Motion capture database," Carnegie Mellon University, http://mocap.cs.cmu.edu/.