# LEARNING TRANSFER-AWARE ENVIRONMENT CURRICULA FROM CROSS-SIMULATOR FEEDBACK

**Anonymous authors**
Manuscript for advisor discussion

## ABSTRACT

A harder navigation task need not offer more useful training for a physical robot. CROSSSIM-CURRICULUM selects geometry edits for cross-floor quadruped navigation by predicting their physical learning gain. Paired short updates in two simulators compare each edit with a control from the same checkpoint. A residual model calibrates these gains with sparse physical feedback; selection then accounts for uncertainty, cost, and edit-family coverage. We derive adaptive calibration intervals and local selection bounds, with seed, horizon, and route-shift allowances connecting probes to deployment. The evaluation protocol tests edit ranking and policy transfer across geometry splits, a held-out backend, and physical routes at matched budgets.

## 1 INTRODUCTION

Stair width, landing shape, clutter, and visibility determine which navigation skills a robot must learn. A curriculum allocates training across these variations. Source-simulator difficulty combines scene challenge with engine-specific contact behavior, making it an imperfect guide to physical learning. For each edit, we instead estimate the physical navigation gain over an equal-budget control.

CROSSSIM-CURRICULUM uses paired updates in two engines to obtain edit-specific evidence, then fits a residual correction from sparse physical feedback. The calibrated estimate guides the learner's next training block. Evaluation asks whether the second engine improves edit ranking and held-out navigation success enough to justify its probe cost.

**Contributions.** Our objective separates realized short-probe gain from expected deployment gain. Bayesian residual calibration predicts probe gain; adaptive intervals and the deployment bridge give a conditional local bound. Matched-budget controls assess cross-engine evidence and physical feedback in the budgeted selector. Figure 1 shows the loop, and Section 7 gives the reporting layout.

## 2 RELATED WORK

**Curricula and environment design.** Adaptive task generation is the focus of GoalGAN, POET, ALP-GMM, and TeachMyAgent (Florensa et al., 2018; Wang et al., 2019; Portelas et al., 2020; Romac et al., 2021). PAIRED, PLR, replay-guided design, and ACCEL prioritize tasks using regret or learning potential (Dennis et al., 2020; Jiang et al., 2021b;a; Parker-Holder et al., 2022); MBeDED and PACE examine provisional policy improvement (Li et al., 2025; Yuan et al., 2026). DRED and CENIE address distribution grounding and diversity (Garcin et al., 2024; Teoh et al., 2024). Our selection target is the physical gain attributable to a geometry edit. Family coverage enters as a separate bonus, so it need not stand in for learning value.

**Transfer feedback.** Domain randomization, Active DR, DORAEMON, and DrEureka adapt the diversity of simulation training (Tobin et al., 2017; Mehta et al., 2020; Tiboni et al., 2024; Ma et al., 2024). SimOpt, BayRn, and transferability surrogates incorporate sparse physical feedback (Chebotar et al., 2019; Muratore et al., 2021; Koos et al., 2013). Cross-engine evidence offers another signal: CROSSER studies dynamics inconsistencies, and RoboVerse supplies multi-backend



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.

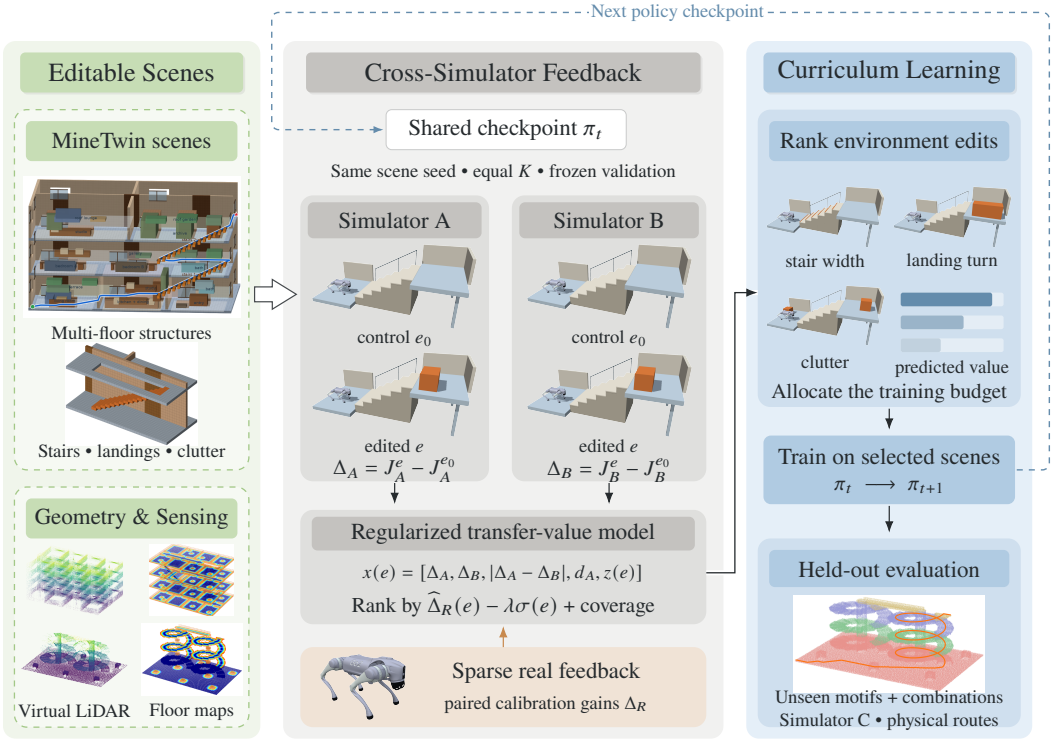



Figure 1: Paired simulator gains and real-world calibration guide the proposed curriculum.

interfaces (Wang et al., 2025; Geng et al., 2025). Sim2Real Predictivity and lower-fidelity transfer results motivate evaluating the rankings induced by these signals (Kadian et al., 2020; Truong et al., 2023). Here, Isaac Lab/PhysX and MuJoCo provide paired training-gain features; Gazebo serves as the held-out backend (Mittal et al., 2025; Todorov et al., 2012; Koenig & Howard, 2004).

**Procedural legged tasks.** ProcTHOR provides procedural embodied scenes (Deitke et al., 2022). Work on terrain learning, rapid training, adaptation, and perception establishes the locomotion setting (Lee et al., 2020; Rudin et al., 2022; Kumar et al., 2021; Miki et al., 2022), while Eurekaverse, LP-ACRL, and TACL study feedback-driven curricula (Liang et al., 2025; Li et al., 2026; Liu et al., 2026). Our geometry-edit interface uses MineTwin's multi-floor assets (Figure 2) to vary stairs, landings, and connected routes.

## 3 Problem Formulation

For scene $g$, a bounded edit $e \in \mathcal{E}(g)$ passes the audit $V(g, e) \in \{0, 1\}$ if geometry, clearance, and route solvability are valid. The finite pool $\mathcal{E}_t$ contains audited edits and control $e_0$. History $\mathcal{H}_t$ records policy $\pi_t$, optimizer state, probes, and remaining budgets. All backends share observations, actions, and training rewards.

**Policy updates and evaluation.** Write $U_s(\pi, e; h, \xi)$ for $h$ stochastic policy updates in simulator $s$. Edited and control updates start from the same checkpoint and couple their training randomness $\xi$, preserving each update's marginal law. On a frozen route distribution $\nu$, the evaluation score is $J_s(\pi; \nu) = \mathbb{E}_{\tau\sim\nu,\omega}[R_s(\pi, \tau, \omega)]$, with evaluation randomness $\omega$ and $R_s \in [0, 1]$. Validation measures route success; training optimizes the shared reward. For a realized update pair, the simulator gain is

$$g^h_{s,t}(e; \xi) = J_s(U_s(\pi_t, e; h, \xi); \nu_s) - J_s(U_s(\pi_t, e_0; h, \xi); \nu_s)\,. \tag{1}$$

The corresponding physical gain evaluates two *source-trained* policies:

$$g^h_{R,t}(e;\xi,\nu) = J_R(U_A(\pi_t, e; h, \xi); \nu) - J_R(U_A(\pi_t, e_0; h, \xi); \nu)\,. \tag{2}$$

Write $\bar{\Delta}^h_{R,t}(e;\nu) = \mathbb{E}_\xi[g^h_{R,t}(e;\xi,\nu) \mid \mathcal{H}_t]$. Calibration estimates the realized short-probe target $g_t(e) = g^K_{R,t}(e;\xi_{t,e},\nu_{\mathrm{cal}})$ for frozen probe policies. The local deployment target is

$$F_t(e) = \bar{\Delta}^L_{R,t}(e;\nu_{\mathrm{test}}), \qquad L > K. \tag{3}$$

A label measures one short-probe realization, whereas $F_t$ averages over training seeds after a longer block on test routes. Evaluation noise, seed variation, horizon mismatch, and route shift therefore enter separately. Test routes never inform selection.

**Paired physical observations.** Label $i$, acquired at checkpoint $t_i$, uses $m_i$ matched route pairs:

$$y_i = \frac{1}{m_i}\sum_{j=1}^{m_i}\left(R^{e_i}_{ij} - R^0_{ij}\right) = g_{t_i}(e_i) + \epsilon_i. \tag{4}$$

These trials evaluate fixed policies on routes drawn according to $\nu_{\mathrm{cal}}$. A label consumes $2m_i$ physical rollouts: at $m_i = 2$, six labels require 24 rollouts per policy seed. More evaluation trials reduce rollout noise; measuring training-seed variation requires independently trained probe pairs.

**Global objective and budgets.** The adapted rule $q_t(e \mid \mathcal{H}_t)$ selects an edit for the student update $\pi_{t+1} = U_A(\pi_t, e_t; L, \xi^{\mathrm{student}}_t)$. Over the complete curriculum, the objective is

$$\begin{aligned} &\max_{\{q_t\}} \ \mathbb{E}[J_R(\pi_T;\nu_{\mathrm{test}})] \\ &\text{subject to} \quad \sum_{t<T}\big(H_t + c_t(e_t)\big) \le B_{\mathrm{sim}}, \\ &\qquad\qquad \sum_{i=1}^{n} 2m_i \le B_{\mathrm{real}}. \end{aligned} \tag{5}$$

Before a student update, $H_t$ is spent on screening, probe updates in both engines, and validation. The update costs $c_t(e)$. Each physics step is counted once; other compute, diagnostics, and final tests are recorded separately. The default schedule, fixed on development data, uses equal-cost student blocks $c_L$. Local selection approximates the full objective; a terminal guarantee needs the condition in Section 5.

## 4 Proposed Method

**Structured candidate pool.** A MineTwin candidate edits a voxel scene while retaining its control's scene seed, robot configuration, start/goal distribution, and remaining geometry. Allowed changes cover stair run or width, landing size, corridor branching, clutter density, and sensor occlusion, each within fixed bounds. Both the edited scene and its control must pass the quality audit. Only edits realizable at the available physical sites enter physical evaluation; the others remain in the simulator analysis.

**Cross-simulator paired probes.** Copies of $\pi_t$ undergo edited and control updates in each engine. The resulting validation gains, $\widehat{g}^K_{A,t}$ and $\widehat{g}^K_{B,t}$, enter the model alongside source difficulty $d_A$ measured before the update. Development data fix the normalization that gives $\widetilde{g}_A, \widetilde{g}_B, \widetilde{d}_A$. We append a fixed-dimensional descriptor $z(e)$ for the edit family and magnitude, obtaining

$$\begin{aligned} \psi_t(e) &= \big[1, \widetilde{g}_A, \widetilde{g}_B, |\widetilde{g}_A - \widetilde{g}_B|, \widetilde{d}_A, z(e)\big], \\ \phi_t(e) &= \frac{\psi_t(e)}{\|\psi_t(e)\|_2}. \end{aligned} \tag{6}$$

The intercept prevents a zero denominator. We fix dimension $d$ and the normalization scales before collecting calibration labels. Probe noise and checkpoint effects absent from these features enter the representation error. In Figure 1, $\Delta_A, \Delta_B$ denote the gains and $\lambda\sigma$ denotes the penalty; the equations use $\kappa_n u_t$ for this penalty and reserve $\lambda$ for ridge precision.

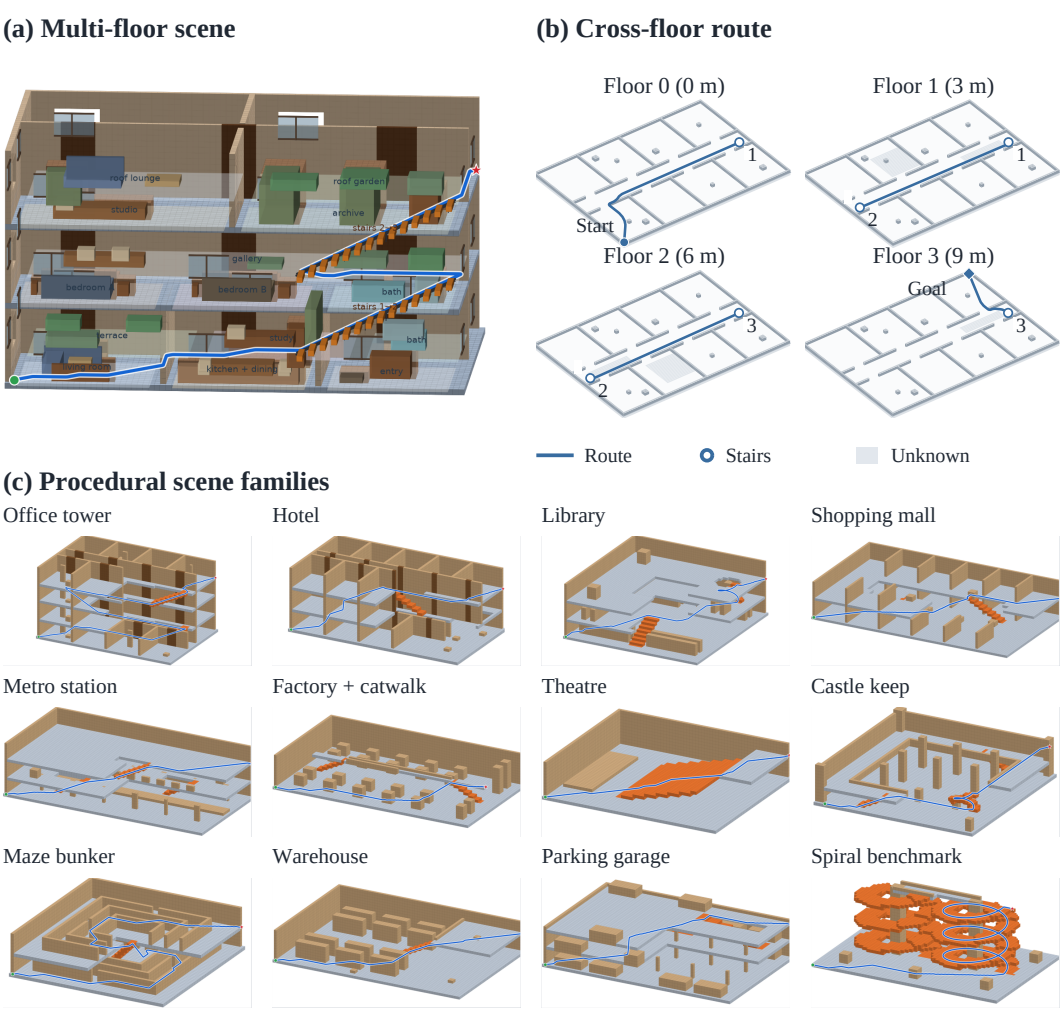



Figure 2: MineTwin assets: multi-floor scenes, procedural families, and point-mass routes.

**Residual Bayesian calibration.** The baseline $f_{0,t}(e) = \mathrm{clip}_{[-1,1]}\{\min(\widetilde{g}_A, \widetilde{g}_B)\}$ uses the smaller normalized simulator gain. We fit the physical labels' residuals relative to this baseline with a Gaussian working model:

$$\begin{aligned} \beta \sim \mathcal{N}(0, \lambda^{-1} I_d), \qquad \epsilon_i \sim \mathcal{N}(0, \sigma_i^2), \\ y_i - f_{0,t_i}(e_i) = \phi_{t_i}(e_i)^\top \beta + \epsilon_i. \end{aligned} \tag{7}$$

The prior is centered on $f_0$ in *function space*. With $\phi_i = \phi_{t_i}(e_i)$ and $r_i = y_i - f_{0,t_i}(e_i)$, the working posterior is $\mathcal{N}(m_n, V_n^{-1})$, where

$$\begin{aligned} V_n &= \lambda I_d + \sum_{i=1}^{n} \frac{\phi_i \phi_i^\top}{\sigma_i^2}, \\ m_n &= V_n^{-1} \sum_{i=1}^{n} \frac{\phi_i r_i}{\sigma_i^2}. \end{aligned} \tag{8}$$

For a current candidate, its mean and coefficient uncertainty are

$$\begin{aligned} \mu_t(e) &= f_{0,t}(e) + \phi_t(e)^\top m_n, \\ u_t(e) &= \sqrt{\phi_t(e)^\top V_n^{-1} \phi_t(e)}. \end{aligned} \tag{9}$$

Equation (9) gives coefficient uncertainty $u_t$. Predicting a new physical label adds rollout-noise variance $\sigma^2$. The Gaussian likelihood yields the ridge posterior, whereas interval guarantees require conditionally centered noise and bounded representation error.

We set $\sigma_i$ before observing label $i$. Since each paired success difference lies in $[-1, 1]$, the choice $\sigma_i = 1$ is valid. Averaging $m_i$ such differences permits $\sigma_i = 1/\sqrt{m_i}$ when they are conditionally independent. Prior strength $\lambda > 0$ and descriptor capacity are chosen on development data.

**Calibration acquisition and coverage.** At a frozen checkpoint, a fixed family-stratified order selects up to six labels. For further labels, choose the unlabelled candidate maximizing $u_t(e)^2/(2m_i)$ in the least-sampled family, with fixed hash order for ties. We store each label with the checkpoint, features, baseline, and noise proxy used to acquire it, then update the posterior. Later reuse keeps this record fixed and requires an allowance for checkpoint drift.

Coverage uses $v_t(e) = 1/(1 + N_t(f(e))) \in [0, 1]$, where $N_t(f)$ counts student blocks on family $f$.

**Budgeted edit selection.** Let $b$ bound residual-model error, and let $\eta_t(e)$ bound the probe-seed, horizon, and route bridge in Section 5. The penalized score and selected edit are

$$\begin{aligned} p_t(e) &= b + \kappa_n u_t(e) + \eta_t(e), \\ \ell_t(e) &= \mu_t(e) - p_t(e), \\ e_t &\in \arg\max_{e \in \mathcal{E}_t^{\text{feas}}} \left\{ \frac{\ell_t(e)}{c_t(e)} + \gamma_t v_t(e) \right\}, \\ q_t(e \mid \mathcal{H}_t) &= \mathbf{1}\{e = e_t\}. \end{aligned} \tag{10}$$

Edits whose declared cost exceeds the remaining budget are excluded. We tune $\kappa_n \geq 0$ and $\gamma_t \geq 0$ on development data; $\kappa_n = \alpha_n$ from Eq. (13) yields the certified rule. Theorem 2 also allows other penalties. The control has zero local gain by definition, so $\mu_t(e_0) = p_t(e_0) = w_t(e_0) = v_t(e_0) = 0$.

For equal-cost non-control edits, we use a common bridge allowance $\eta_t$. Before any labels, $m_0 = 0$ and $u_t(e) = 1/\sqrt{\lambda}$, so the shared penalty leaves their ranking determined by $f_{0,t}(e) + c_L \gamma_t v_t(e)$. With $\gamma_t = 0$, this is baseline ranking. After calibration, a deployment guarantee still requires the interval and bridge assumptions. Evaluation checks interval coverage alongside development-tuned penalties.

**Training loop.** We fix the candidate quota, $K$, $L$, batch size, and schedule on development data before the first round. Each round starts by reserving its student block. We then audit and probe the frozen pool, acquire labels within $B_{\text{real}}$, and train on the selected edit. If the budget cannot fund another full round, we use the remainder on the last selected edit (initially $e_0$). That partial block is logged separately because the theory covers complete rounds. Appendix C gives the procedure and ledgers.

## 5 Theoretical Analysis

We bound calibration error for realized short probes, then bridge to expected deployment gain (Section 3). Proofs are in the full version's Appendix A.

**Proposition 1** (Paired-label precision)**.** *For fixed probe policies, matched route allocation, and $m$ conditionally independent paired trials, $\mathbb{E}[y \mid \xi] = g^K_{R,t}(e; \xi, \nu_{\text{cal}})$ and*

$$\text{Var}(y \mid \xi) = \frac{1}{m} \left( \text{Var}(R^e \mid \xi) + \text{Var}(R^0 \mid \xi) - 2\,\text{Cov}(R^e, R^0 \mid \xi) \right). \tag{11}$$

*Over independently drawn training seeds, $\text{Var}(y) = \text{Var}_\xi(g^K_{R,t}) + \mathbb{E}_\xi[\text{Var}(y \mid \xi)]$.*

**Proposition 2** (Conditional value of simulator B)**.** *On a fixed distribution of checkpoints, edits, and probe seeds, let $G$ be the physical short-probe gain, $X_A$ the A-only features, and $X_{AB}$ their augmentation with B features. Define $m_A = \mathbb{E}[G \mid X_A]$ and $m_{AB} = \mathbb{E}[G \mid X_{AB}]$. Then*

$$\mathbb{E}[(G - m_A)^2] - \mathbb{E}[(G - m_{AB})^2] = \mathbb{E}[(m_{AB} - m_A)^2] \geq 0. \tag{12}$$

*The reduction is strict exactly when simulator B changes the conditional mean on a set of positive probability.*

For fixed probe policies, positive covariance reduces the rollout-noise term in Proposition 1; additional trials shrink it through $1/m$. The training-seed term remains. Proposition 2 concerns population conditional means and gives no guarantee on the error of estimating them from a few labels.

**Assumption 1** (Residual stability and observable noise)**.** Across acquired rows and queried candidates, $g_t(e) = f_{0,t}(e) + \phi_t(e)^\top \beta_\star + a_t(e)$, with $\|\beta_\star\|_2 \le S$ and $|a_t(e)| \le b$. Features, baselines, and $\sigma_i > 0$ are measurable before observing $y_i$. For that pre-observation history $\mathcal{F}_{i-1}$, $\mathbb{E}[\exp(z\epsilon_i/\sigma_i) \mid \mathcal{F}_{i-1}] \le \exp(z^2/2)$ for every $z \in \mathbb{R}$.

$S$ bounds coefficient norm; $b$ allows for probe measurement error, nonlinear residuals, and checkpoint drift. Development fits and sensitivity analysis assess both inputs to the theorem.

**Theorem 1** (Adaptive calibration interval)**.** *Under Assumption 1, fix $\delta \in (0,1)$ and let*

$$\alpha_n = \sqrt{\lambda} S + \sqrt{2\log\left(\frac{\det(V_n)^{1/2}}{\lambda^{d/2}\delta}\right)} + b\sqrt{\sum_{i=1}^{n} \sigma_i^{-2}}. \tag{13}$$

*With probability at least $1-\delta$, simultaneously over all acquired sample sizes and current queries,*

$$|g_t(e) - \mu_t(e)| \le b + \alpha_n u_t(e). \tag{14}$$

Equation (13) adds prior shrinkage, self-normalized noise (Abbasi-Yadkori et al., 2011), and representation error from acquired rows. The separate $b$ covers the query's representation error. With six labels, small fitted residuals can coexist with large $u_t(e)$ when query features lie along poorly constrained directions.

**Bridge to the deployment target.** Suppose, on the current pool, that

$$\begin{aligned} |g_t(e) - \bar{\Delta}^K_{R,t}(e;\nu_{\mathrm{cal}})| &\le b_{\mathrm{seed},t}(e), \\ |\bar{\Delta}^K_{R,t}(e;\nu_{\mathrm{cal}}) - \bar{\Delta}^L_{R,t}(e;\nu_{\mathrm{cal}})| &\le b_{\mathrm{horizon},t}(e), \\ \sup_{\pi\in\Pi_t} |J_R(\pi;\nu_{\mathrm{cal}}) - J_R(\pi;\nu_{\mathrm{test}})| &\le b_{\mathrm{route},t}, \end{aligned} \tag{15}$$

where $\Pi_t$ includes the edited/control policies reachable at horizon $L$. The triangle inequality gives a sufficient bridge allowance $\eta_t(e) = b_{\mathrm{seed},t}(e) + b_{\mathrm{horizon},t}(e) + 2b_{\mathrm{route},t}$. Thus $w_t(e) = b + \alpha_n u_t(e) + \eta_t(e)$ satisfies $|F_t(e) - \mu_t(e)| \le w_t(e)$ on the calibration event. Development evidence fixes the teacher's bounds before held-out assessment. Diagnostics target each component: independently trained probes for seed variation, $K$-versus-$L$ comparisons for horizon mismatch, and site-disjoint tests for route shift.

**Theorem 2** (Budget-normalized local selection)**.** *Suppose $|F_t(e) - \mu_t(e)| \le w_t(e)$ for all feasible edits, $c_t(e) > 0$, and $v_t(e) \in [0,1]$. Let $e_t^\star$ maximize $F_t(e)/c_t(e)$, and write $[x]_+ = \max(x,0)$. The rule in Eq.* (10) *satisfies*

$$\frac{F_t(e_t^\star)}{c_t(e_t^\star)} - \frac{F_t(e_t)}{c_t(e_t)} \le \frac{w_t(e_t^\star) + p_t(e_t^\star)}{c_t(e_t^\star)} + \frac{[w_t(e_t) - p_t(e_t)]_+}{c_t(e_t)} + \gamma_t. \tag{16}$$

*In the certified setting $p_t = w_t$, the right-hand side reduces to $2w_t(e_t^\star)/c_t(e_t^\star) + \gamma_t$.*

At equal cost $c_L$, certification gives a gain gap of at most $2w_t(e_t^\star) + c_L\gamma_t$ relative to the best feasible edit. With other penalties, the selected edit contributes the second term in Eq. (16) when its error bound exceeds its penalty. Two edits have a certified ordering if $\mu_t(e) - \mu_t(e') > w_t(e) + w_t(e')$. Uniform error $\varepsilon$ with $p_t = \gamma_t = 0$ recovers the mean-greedy gap $2\varepsilon$.

**Connection to complete curricula.** $F_t(e)$ compares edits from one checkpoint. Later updates can change their ordering, so positive local gain alone gives no terminal guarantee. Appendix B gives a downstream compatibility condition and a two-round counterexample when it fails. Complete-policy tests assess the accumulated effect.

# 6 EXPERIMENTAL PROTOCOL

**Task and system design.** Waypoint routes require stair ascent, descent, and navigation across landings. A trial succeeds when the robot reaches the goal without a fall, collision intervention, or timeout. To compare the engines, we hold the robot, controller, observations, actions, reward, and optimizer fixed. MineTwin supplies the scenes and maps, and the physical-feedback interface returns calibration labels.

**Disjoint tests and matched budgets.** We separate training and validation by structural motif, then test on unseen motifs and compositions. Simulator C is used only for evaluation. Physical tests also hold out staircase/site and start–goal pair; with access to just one site, conclusions are restricted to routes and compositions within that site. The test plan allows at most five policy seeds across twelve routes with two repeats, or 120 episodes per method. Methods using physical feedback receive another 24 calibration rollouts per seed. Pilot power and site access determine the final plan before testing.

A-only + real omits simulator B and reallocates the saved probe steps to source training. Real edit BO searches the descriptors using six paired labels. These controls use the full method's edit space, calibration routes, and $B_{\mathrm{sim}}$. Comparisons also include progress, regret, randomization, and multi-simulator baselines. Online costs follow Section 3; diagnostics and final tests have separate ledgers. The planned outcomes are physical success and edit-gain rank predictivity. Hierarchical 95% confidence intervals account for seeds and route/site clusters; Appendix C details adaptations and accounting.

# 7 COMPARATIVE ANALYSIS

## 7.1 MAIN COMPARISON

Table 1: Success percentages (mean $\pm$ seed SD). Struct. and Compos. denote the structural and compositional splits; Real averages them. Sim-C denotes the held-out backend. Cal is the calibration budget *per seed*.

| Method | Struct. | Compos. | Real | Sim-C | Cal |
|---|---|---|---|---|---|
| Uniform | $70.0 \pm 11.2$ | $35.0 \pm 9.1$ | $52.5 \pm 7.6$ | $65.3 \pm 6.6$ | 0 |
| Source difficulty | $71.7 \pm 15.1$ | $53.3 \pm 17.3$ | $62.5 \pm 14.1$ | $66.3 \pm 6.3$ | 0 |
| ALP-GMM | $66.7 \pm 15.6$ | $46.7 \pm 7.5$ | $56.7 \pm 9.1$ | $68.4 \pm 7.0$ | 0 |
| PLR | $56.7 \pm 14.9$ | $50.0 \pm 15.6$ | $53.3 \pm 12.3$ | $69.7 \pm 6.5$ | 0 |
| PLR$^{\perp}$ | $61.7 \pm 22.5$ | $58.3 \pm 21.2$ | $60.0 \pm 16.3$ | $71.6 \pm 5.7$ | 0 |
| PAIRED | $70.0 \pm 17.3$ | $65.0 \pm 7.0$ | $67.5 \pm 11.2$ | $69.7 \pm 5.5$ | 0 |
| ACCEL | $76.7 \pm 13.7$ | $48.3 \pm 19.9$ | $62.5 \pm 12.1$ | $72.4 \pm 6.5$ | 0 |
| Active DR | $63.3 \pm 24.0$ | $60.0 \pm 10.9$ | $61.7 \pm 12.6$ | $73.1 \pm 5.2$ | 0 |
| Multi-sim uniform | $78.3 \pm 7.5$ | $50.0 \pm 16.7$ | $64.2 \pm 7.0$ | $74.2 \pm 6.3$ | 0 |
| SimOpt-style | $71.7 \pm 15.1$ | $56.7 \pm 14.9$ | $64.2 \pm 8.1$ | $73.6 \pm 5.9$ | 24 |
| BayRn-style | $66.7 \pm 11.8$ | $63.3 \pm 20.1$ | $65.0 \pm 8.6$ | $74.9 \pm 4.0$ | 24 |
| A-only + real | $75.0 \pm 11.8$ | $56.7 \pm 29.1$ | $65.8 \pm 9.9$ | $76.3 \pm 5.7$ | 24 |
| Real edit BO | $68.3 \pm 16.0$ | $61.7 \pm 9.5$ | $65.0 \pm 9.1$ | $76.1 \pm 1.7$ | 24 |
| Ours, no real | $68.3 \pm 12.4$ | $71.7 \pm 7.5$ | $70.0 \pm 3.5$ | $75.8 \pm 4.5$ | 0 |
| **Ours, full** | $81.7 \pm 16.0$ | $65.0 \pm 17.1$ | $73.3 \pm 4.8$ | $78.2 \pm 4.4$ | 24 |

Real edit BO and A-only + real provide the matched-budget controls for the full method in Table 1. ACCEL tests regret-based editing, while the zero-feedback variant isolates the contribution of physical calibration.

**Mechanism and horizon diagnostics.** The full version's Appendix D (Tables 2 and 3) specifies calibration, pairing, simulator-B, coverage, and permuted-label controls. The horizon diagnostic compares $K$- and $L$-trained pairs on eight held-out edits with 64 physical rollouts, separate from fitting and final tests. Independent probes, site-level tests, and residual stratification assess the remaining deployment-error terms. The eight-edit estimate applies to its sampled checkpoint.

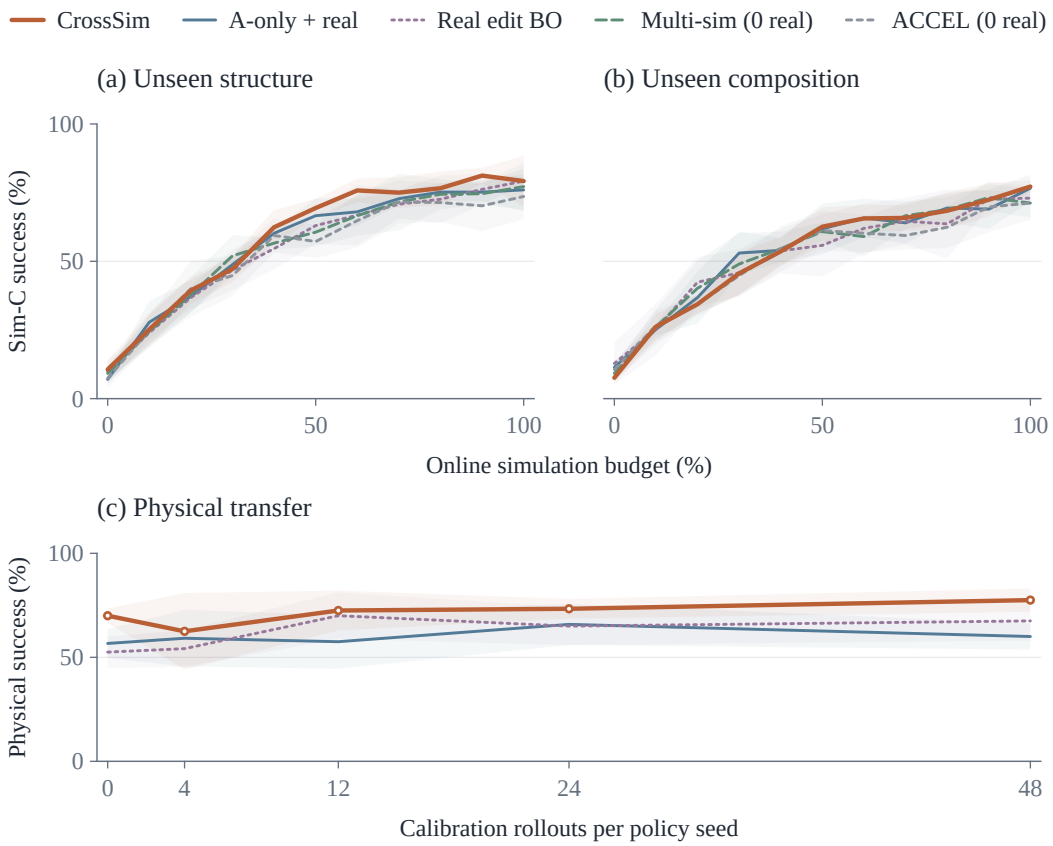



Figure 3: Transfer success and efficiency under simulation and physical calibration budgets.

### 7.2 Training and real-feedback efficiency

Figure 3 separates simulation cost from physical calibration cost. Panels (a,b) use checkpoint success counts. Panel (c) compares 0, 4, 12, 24, and 48 calibration rollouts, corresponding to 0, 1, 3, 6, and 12 paired labels. At zero feedback, the full selector uses $f_0$ plus coverage, A-only uses source gain, and Real edit BO samples uniformly. Shaded bands are $\pm 1$ seed SD, not confidence intervals. The 24-rollout entries share the records in Table 1; Appendix D gives the accounting.

## 8 Limitations and Conclusion

The value of cross-engine feedback depends on whether the navigation gain justifies its acquisition cost. CrossSim-Curriculum estimates each geometry edit's gain over a control from simulator probes, uses physical labels to fit a residual correction, and selects edits with uncertainty, cost, and family coverage taken into account.

Uncorrected shared engine errors, checkpoint drift, and site shift must fall within the residual and bridge allowances. Precision is limited by the eight-edit horizon check; the terminal bound in Appendix B additionally assumes downstream compatibility. Physical scope is one robot and the available staircase sites. Trials require operator supervision, stop rules, and intervention logs; any privileged observations require disclosure and a separate comparison.

## REPRODUCIBILITY STATEMENT

This package includes the manuscript, numerical records, and table/curve reconstruction scripts. A reproducible experimental release must also identify scene seeds, edit manifests, simulator/robot versions, matched observation/action specifications, candidate-pool hashes, policy checkpoints, split IDs, cost ledgers, and per-seed outcomes. Code and supporting materials will be made available in the anonymous repository: `https://github.com/dkuycky/anonymous_iclr_repo`.

## AI USE STATEMENT

GPT-5.6-sol and GPT-6-Astra were used solely for manuscript formatting and language polishing.

## REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24, pp. 2312–2320, 2011. URL `https://proceedings.neurips.cc/paper/2011/hash/e1d5be1c7f2f456670de3d53c7b54f4a-Abstract.html`.

Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *IEEE International Conference on Robotics and Automation*, 2019. doi: 10.1109/ICRA.2019.8793789.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-scale embodied AI using procedural generation. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-0433.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart J. Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1515–1528, 2018.

Samuel Garcin, James Doran, Shangmin Guo, Christopher G. Lucas, and Stefano V. Albrecht. DRED: Zero-shot transfer in reinforcement learning via data-regularised environment design. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15006–15036, 2024.

Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Haozhe Lou, Charlie Tianyue Cheng, Peihao Li, Haozhe Chen, Yutong Liang, Yuxi Qian, Jiageng Mao, Weikang Wan, Yiran Geng, Mingtong Zhang, Jiangran Lyu, Siheng Zhao, Jiazhao Zhang, Chaoyi Xu, Jialiang Zhang, Chengyang Zhao, Haoran Lu, Yufei Ding, Ran Gong, Yuran Wang, Yuxuan Kuang, Ruihai Wu, Baoxiong Jia, Hao Dong, Siyuan Huang, Yue Wang, Jitendra Malik, and Pieter Abbeel. RoboVerse: A unified platform, benchmark and dataset for scalable and generalizable robot learning. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.022.

Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. In *Advances in Neural Information Processing Systems*, volume 34, 2021a.

Minqi Jiang, Edward Grefenstette, and Tim Rockt"aschel. Prioritized level replay. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4940–4950, 2021b.

Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2Real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 2020. doi: 10.1109/LRA.2020.3013848.

Nathan Koenig and Andrew Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2149–2154, 2004. doi: 10.1109/IROS.2004.1389727.

Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17 (1):122–145, 2013. doi: 10.1109/TEVC.2012.2185849.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Proceedings of Robotics: Science and Systems*, 2021. doi: 10.15607/RSS.2021.XVII.011.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020. doi: 10.1126/scirobotics.abc5986.

Dexun Li, Wenjun Li, and Pradeep Varakantham. Marginal benefit driven RL teacher for unsupervised environment design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18253–18261, 2025. doi: 10.1609/aaai.v39i17.34008.

Ziming Li, Chenhao Li, and Marco Hutter. Scaling rough terrain locomotion with automatic curriculum reinforcement learning. *arXiv preprint arXiv:2601.17428*, 2026. URL https://arxiv.org/abs/2601.17428.

William Liang, Sam Wang, Hung-Ju Wang, Osbert Bastani, Dinesh Jayaraman, and Yecheng Jason Ma. Environment curriculum generation via large language models. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 433–454, 2025.

Rocky Liu, Tengyu Liu, Baoxiong Jia, Fangwei Zhong, Xinyi Tong, Hongzhao Xie, and Siyuan Huang. Trajectory-level automatic curriculum learning for legged locomotion on unstructured terrain. *arXiv preprint arXiv:2608.16164*, 2026. URL https://arxiv.org/abs/2608.16164.

Yecheng Jason Ma, William Liang, Hung-Ju Wang, Yuke Zhu, Linxi Fan, Osbert Bastani, and Dinesh Jayaraman. DrEureka: Language model guided sim-to-real transfer. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.094.

Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J. Pal, and Liam Paull. Active domain randomization. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1162–1176, 2020.

Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022. doi: 10.1126/scirobotics.abk2822.

Mayank Mittal et al. Isaac Lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025. URL https://arxiv.org/abs/2511.04831.

Fabio Muratore, Christian Eilers, Michael Gienger, and Jan Peters. Data-efficient domain randomization with bayesian optimization. *IEEE Robotics and Automation Letters*, 2021. doi: 10.1109/LRA.2021.3052391.

Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17473–17498, 2022.

Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep RL in continuously parameterized environments. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 835–853, 2020.

Clément Romac, Rémy Portelas, Katja Hofmann, and Pierre-Yves Oudeyer. TeachMyAgent: A benchmark for automatic curriculum learning in deep RL. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9052–9063, 2021.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 91–100, 2022.

Jayden Teoh, Wenjun Li, and Pradeep Varakantham. Improving environment novelty quantification for effective unsupervised environment design. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Gabriele Tiboni, Pascal Klink, Jan Peters, Tatiana Tommasi, Carlo D'Eramo, and Georgia Chalvatzaki. Domain randomization via entropy maximization. In *International Conference on Learning Representations*, 2024.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017. doi: 10.1109/IROS.2017.8202133.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Joanne Truong, Max Rudolph, Naoki Harrison Yokoyama, Sonia Chernova, Dhruv Batra, and Akshara Rai. Rethinking sim2real: Lower fidelity simulation leads to higher sim2real transfer in navigation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 859–870, 2023.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (POET): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019. URL `https://arxiv.org/abs/1901.01753`.

Zepeng Wang, Xiaochuan Shi, Ziluo Ding, Yuxuan Wang, Zhenguo Sun, Chao Ma, and Zongqing Lu. CROSSER: Learning generalizable humanoid locomotion through inverse dynamics-guided cross-simulator adaptation. *IEEE Robotics and Automation Letters*, 10(12):13336–13343, 2025. doi: 10.1109/LRA.2025.3630544.

Fang Yuan, Quanjun Yin, Siqi Shen, Yuxiang Xie, Junqiang Yang, Long Qin, Junjie Zeng, and Qinglun Li. PACE: Parameter change for unsupervised environment design. In *International Conference on Machine Learning*, 2026. URL `https://arxiv.org/abs/2605.01358`.