WHICH TASKS TRAIN ADAPTATION? PAIRED INTERVENTIONS FOR META-RL TASK SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training a meta-RL agent requires tasks on which within-episode adaptation is useful, not merely difficult tasks. Existing curricula rank tasks by difficulty, regret, learning progress, or post-adaptation return, but these signals do not reveal how much performance depends on adaptation. To address this issue, we introduce *Paired Adaptation-Contrast Ecology* (PACE), which directly measures this dependence by comparing matched executions with the episode-local update enabled or disabled and evaluating both on a frozen suffix. The resulting learner-dependent return difference is used to update the training distribution in a closed loop as the learner changes. In controlled matched-Post experiments, high-AdaptGain task groups improve Post by 0.0069 more than matched low-AdaptGain groups under the same retraining budget. Across native benchmarks, PACE achieves the highest mean Post in 18 of 20 compatible benchmark–learner cells.

1 INTRODUCTION

In many sequential decision problems, an agent must learn while acting. The task or latent context may be only partly known at the start, so early interactions must inform later decisions. Meta-reinforcement learning (meta-RL) addresses this setting through recurrent policies that accumulate task evidence (Duan et al., 2016; Wang et al., 2016), gradient-based methods that adapt task-specific parameters (Finn et al., 2017), and latent-context or in-context methods that use information gathered within an episode (Rakelly et al., 2019; Zintgraf et al., 2020; Laskin et al., 2023). These mechanisms enable fast adaptation, but the training distribution must also contain tasks where adaptation is useful. Tasks already solvable from static parameters provide little signal for learning to adapt, while highly noisy tasks can be difficult yet provide little useful evidence. Task selection should therefore consider not only difficulty, but whether a task actually benefits from episode-local adaptation.

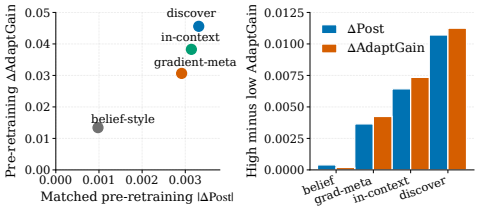
Curriculum learning has long studied how to choose useful training examples. Classical methods organize training by difficulty or pace (Bengio et al., 2009; Kumar et al., 2010), while automated teachers use competence and learning progress (Graves et al., 2017; Matiisen et al., 2020). In reinforcement learning, goal generation, domain randomization, and open-ended environment generation extend this idea to the experience distribution (Florensa et al., 2018; Mehta et al., 2020; Wang et al., 2019). Unsupervised environment design makes task selection explicitly learner-dependent: PAIRED uses regret (Dennis et al., 2020), PLR uses estimated learning potential (Jiang et al., 2021b), and ACCEL edits tasks near the learner's evolving capability frontier (Parker-Holder et al., 2022). Later work improves task representation, regularization, regret estimation, and UED optimization (Garcin et al., 2024; Mead et al., 2025; Monette et al., 2025). These methods capture difficulty, progress, regret, learning potential, or frontier position, but not whether current performance depends on the learner's episode-local adaptation mechanism.

Recent work moves closer to this question. ITTS balances target relevance with diversity (Luna Gutierrez & Leonetti, 2020), ULEE tracks evolving post-adaptation performance (Pappalardo, 2026), and MPTS predicts task-specific adaptation risk while reducing online evaluation cost (Wang et al., 2026). Yet post-adaptation return tells us what the learner achieves after adaptation, and adaptation risk describes how difficult that adaptation may be; neither measures how much performance depends on adaptation itself. Two tasks can have nearly the same post-adaptation return even if one is solved almost as well with adaptation disabled while the other relies strongly on information gathered within

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.

the episode. For an adaptive learner, these tasks are not equivalent. We therefore ask: for the current learner, which tasks actually benefit from its episode-local adaptation channel?

We introduce *Paired Adaptation-Contrast Ecology* (PACE) to measure this dependence and use it for task selection. For each candidate task family, PACE compares matched executions with the episode-local update enabled or disabled. Both executions are evaluated on a suffix in which further local updates are frozen. Their mean suffix-return difference over matched pairs is *AdaptGain*, the finite-sample estimate of adaptation dependence used throughout the experiments. PACE uses positive AdaptGain as its primary signal, combines it with a soft Post frontier and a Static-saturation penalty, and encourages descriptor diversity; in a fixed pool it also favors under-exposed descriptors. These quantities are remeasured as the learner changes, producing the next training distribution. Figure 1 shows the central distinction: task groups can be closely matched on current Post while differing substantially in AdaptGain, and the higher-AdaptGain group can respond more strongly to the same retraining budget.



(a) Matched Post and AdaptGain (b) Equal-retraining response

Figure 1: Matched-Post task families differ in AdaptGain and retraining response.

The paper contributes a matched intervention and paired estimator for adaptation dependence, together with finite-sample guarantees for the measurement. PACE then turns this estimate into a closed-loop task selector that tracks where adaptation remains useful as the learner changes. In matched-Post experiments, high-AdaptGain tasks improve Post by 0.0069 more than matched low-AdaptGain tasks under the same retraining budget; across native benchmarks, PACE achieves the highest mean Post in 18 of 20 compatible benchmark–learner cells.

2 Task Selection for Adaptive Learners

Existing task-selection methods differ mainly in how they assign value to training tasks. Curriculum learning uses difficulty, pace, competence, or learning progress to shape the training distribution (Bengio et al., 2009; Kumar et al., 2010; Graves et al., 2017; Matiisen et al., 2020). In reinforcement learning, goal generation, domain randomization, and open-ended environment generation extend the same idea to the experience distribution (Florensa et al., 2018; Tobin et al., 2017; Mehta et al., 2020; Wang et al., 2019). Unsupervised environment design makes task selection explicitly learner-dependent: PAIRED uses regret (Dennis et al., 2020), PLR uses estimated learning potential (Jiang et al., 2021b), and ACCEL edits tasks near the learner's capability frontier (Parker-Holder et al., 2022). Later work improves task representations, regularization, and UED optimization (Azad et al., 2023; Garcin et al., 2024; Mead et al., 2025; Monette et al., 2025). These signals describe challenge, failure, progress, or where further learning may be possible.

Adaptive learners introduce a different quantity: how much current performance depends on the update made within an episode. ITTS considers target relevance and diversity, ULEE follows post-adaptation performance, and MPTS predicts adaptation risk (Luna Gutierrez & Leonetti, 2020; Pappalardo, 2026; Wang et al., 2026). None of these quantities is equivalent to adaptation dependence. The distinction also does not depend on one meta-RL architecture. Recurrent meta-RL updates hidden state (Duan et al., 2016; Wang et al., 2016); MAML updates task-specific parameters (Finn et al., 2017); PEARL and VariBAD update latent task or belief states (Rakelly et al., 2019; Zintgraf et al., 2020); and in-context RL uses interaction history to shape later behavior (Laskin et al., 2023; Lee et al., 2023). In each case, the same question applies: how much does updating the episode-local object improve later performance for the current learner on the current task?

Adaptation dependence changes as the learner improves. A task that initially requires within-episode adaptation may later be handled well by static parameters, while selecting only tasks with the largest AdaptGain can concentrate training in a narrow part of the task space. To separate these effects, we distinguish three quantities: Post locates the learner's current performance region, AdaptGain estimates adaptation dependence, and descriptor diversity measures task-space breadth. In finite pools, we additionally report *Coverage*, the fraction of descriptors selected at least once, as an experimental diagnostic of this breadth rather than an input to the PACE selection utility. This

Table 1: Controlled correlated-bandit slice at $K = 5$ (mean $\pm$ s.d. over five seeds).

Learner	Teacher	Static	Post	AdaptGain	Coverage
belief-style	PACE	0.6279 ± 0.0129	0.6254 ± 0.0132	-0.0025 ± 0.0005	0.6979 ± 0.0236
belief-style	Post frontier	0.6276 ± 0.0108	0.6255 ± 0.0106	-0.0021 ± 0.0006	0.6188 ± 0.0170
belief-style	Uniform	0.6272 ± 0.0130	0.6250 ± 0.0133	-0.0022 ± 0.0008	0.7417 ± 0.0047
discover	PACE	0.6258 ± 0.0151	0.6457 ± 0.0143	$+0.0199 \pm 0.0012$	0.6031 ± 0.0154
discover	Post frontier	0.6279 ± 0.0140	0.6472 ± 0.0121	$+0.0193 \pm 0.0019$	0.6219 ± 0.0131
discover	Uniform	0.6281 ± 0.0137	0.6469 ± 0.0127	$+0.0188 \pm 0.0015$	0.7385 ± 0.0189

distinction is consistent with quality-diversity methods, which seek strong solutions across different regions (Mouret & Clune, 2015; Pugh et al., 2016), and open-ended search, which relies on diverse stepping stones for continued learning (Wang et al., 2019; Ecoffet et al., 2021). Table 1 illustrates the separation: for a fixed learner, PACE, the Post frontier, and Uniform achieve similar Post while producing different Coverage, whereas across learners AdaptGain behaves differently, remaining near zero or negative for belief-style and positive for discover. Post, AdaptGain, and Coverage therefore capture distinct properties of the training process rather than interchangeable notions of task difficulty.

3 Proposed Design of PACE

PACE follows one closed loop. For the current learner, it first measures how much each candidate task depends on episode-local adaptation. It then combines that signal with the learner's current performance region and task-space breadth to obtain a selection utility. Training changes the learner, so the same quantities are measured again before the next ecology update.

3.1 Measuring adaptation dependence

Let $\mathcal{D}$ be the descriptor space. A descriptor $d \in \mathcal{D}$ indexes a task-family distribution $\mathcal{T}(d)$, and $\omega \sim \mathcal{T}(d)$ denotes one task instance. During measurement, learner parameters θ are fixed. At step t, the learner observes o_t, takes action a_t, receives feedback r_t, and may update an episode-local state m_t initialized at m_0. The policy is $\pi_{\theta,m}(\cdot \mid o)$, and $\mathcal{U}_\theta$ denotes the learner-specific local update. A measured episode has a write prefix of length $K \geq 0$ followed by a frozen evaluation suffix of length $H > 0$. In the adaptive execution,

$$a_t^{\mathrm{ad}} \sim \pi_{\theta,m_{t-1}^{\mathrm{ad}}}(\cdot \mid o_t^{\mathrm{ad}}), \qquad m_t^{\mathrm{ad}} = \mathcal{U}_\theta\big(m_{t-1}^{\mathrm{ad}}, o_t^{\mathrm{ad}}, a_t^{\mathrm{ad}}, r_t^{\mathrm{ad}}\big), \qquad 1 \leq t \leq K, \tag{1}$$

and the state reached at the boundary is then held fixed,

$$a_t^{\mathrm{ad}} \sim \pi_{\theta,m_K^{\mathrm{ad}}}(\cdot \mid o_t^{\mathrm{ad}}), \qquad m_t^{\mathrm{ad}} = m_K^{\mathrm{ad}}, \qquad K < t \leq K + H. \tag{2}$$

The matched Static execution uses the same learner and task instance but disables the local update from the start,

$$a_t^{\mathrm{st}} \sim \pi_{\theta,m_0}(\cdot \mid o_t^{\mathrm{st}}), \qquad m_t^{\mathrm{st}} = m_0, \qquad 1 \leq t \leq K + H. \tag{3}$$

The local object may be recurrent state, inferred context, belief state, fast parameters, or another episode-local representation. Freezing both executions during the suffix keeps further local updates out of the evaluation window. For mode $c \in \{\mathrm{ad}, \mathrm{st}\}$, let $\bar{r}_t^c \in [0, 1]$ be the normalized evaluation utility and define

$$\mathrm{Post}_c = \frac{1}{H} \sum_{t=K+1}^{K+H} \bar{r}_t^c. \tag{4}$$

Taking expectation over task instances and trajectory randomness gives

$$R_{\mathrm{post}}(d; \theta) = \mathbb{E}[\mathrm{Post}_{\mathrm{ad}} \mid d, \theta], \qquad R_{\mathrm{static}}(d; \theta) = \mathbb{E}[\mathrm{Post}_{\mathrm{st}} \mid d, \theta]. \tag{5}$$

We define the population *adaptation dependence* as

$$g(d; \theta) = R_{\mathrm{post}}(d; \theta) - R_{\mathrm{static}}(d; \theta). \tag{6}$$

Post records what the learner achieves after the adaptive prefix, whereas g records how much later performance changes when the local update is available. The Static reference is necessary because

Post alone does not identify this dependence. If $R_{\text{post}} = p \in [0, 1]$, then $g(d; \theta) \in [p - 1, p]$, so the same Post value is compatible with negative, zero, or positive dependence. Conversely, if the local state never changes the policy readout, the two executions induce the same behavior and $g = 0$. In an interactive environment, the adaptive prefix can also change the external states reached by the learner, so g measures the total downstream effect of enabling the designated local update.

PACE estimates g with matched task instances. Draw N independent instances $\omega_1, \ldots, \omega_N \sim \mathcal{T}(d)$ and run both executions on each one. Let

$$A_n \equiv \text{Post}_{\text{ad},n}, \qquad S_n \equiv \text{Post}_{\text{st},n}, \qquad Y_n = A_n - S_n, \tag{7}$$

and define

$$\widehat{g}_N(d; \theta) = \frac{1}{N} \sum_{n=1}^{N} Y_n, \qquad \widehat{R}_{\text{post},N} = \frac{1}{N} \sum_{n=1}^{N} A_n, \qquad \widehat{R}_{\text{static},N} = \frac{1}{N} \sum_{n=1}^{N} S_n. \tag{8}$$

We call the empirical paired estimate $\widehat{g}_N$ *AdaptGain*. Thus g is the population adaptation dependence and AdaptGain is its finite-sample estimate. Pairing holds task identity fixed across the two executions. When the paired returns are positively correlated, it also reduces the variance of their difference,

$$\text{Var}(Y_n) = \text{Var}(A_n) + \text{Var}(S_n) - 2\text{Cov}(A_n, S_n). \tag{9}$$

PACE later uses the positive part of AdaptGain in its selection utility, so the relevant finite-sample question is whether matched measurement also controls the error of that positive term. The following proposition provides the basic guarantee.

Proposition 3.1. *Assume the paired observations are independent across n. For fixed d and θ, $\mathbb{E}[\widehat{g}_N(d; \theta)] = g(d; \theta)$, and for any $\delta \in (0, 1)$,*

$$\Pr\left(|\widehat{g}_N(d; \theta) - g(d; \theta)| \le \sqrt{\frac{2 \log(2/\delta)}{N}} \right) \ge 1 - \delta. \tag{10}$$

Proof. For each matched pair, $Y_n \in [-1, 1]$ and $\mathbb{E}[Y_n] = R_{\text{post}} - R_{\text{static}} = g(d; \theta)$. Linearity of expectation gives $\mathbb{E}[\widehat{g}_N] = g$. Hoeffding's inequality (Hoeffding, 1963) for variables with range length two then gives Eq. 10. □

Define the population and empirical positive parts by

$$g_+(d; \theta) = [g(d; \theta)]_+, \qquad \widehat{g}_{N,+}(d; \theta) = [\widehat{g}_N(d; \theta)]_+. \tag{11}$$

Because $x \mapsto [x]_+$ is 1-Lipschitz,

$$|\widehat{g}_{N,+}(d; \theta) - g_+(d; \theta)| \le |\widehat{g}_N(d; \theta) - g(d; \theta)|. \tag{12}$$

Thus Eq. 10 controls the estimation error of the same positive adaptation term that PACE uses for selection. Its error decreases as $O(N^{-1/2})$ as more matched pairs are collected. The clipping boundary at zero also gives a natural rule for allocating measurement effort. Tasks with $g \le 0$ contribute no positive adaptation term, whereas tasks with $g > 0$ can receive positive weight. For cumulative stages $N_1 < \cdots < N_L$, define

$$\beta_\ell = \sqrt{\frac{2 \log(2L/\delta)}{N_\ell}}. \tag{13}$$

At stage ℓ, a descriptor is sign-resolved when $[\widehat{g}_{N_\ell} - \beta_\ell, \widehat{g}_{N_\ell} + \beta_\ell]$ lies entirely in $[0, \infty)$ or $(-\infty, 0]$.

Corollary 3.2. *Fix a descriptor d and learner parameters θ. With probability at least $1 - \delta$, all L stage intervals for this descriptor are valid simultaneously. If, for some $h > 0$,*

$$N_\ell \ge 8 \log(2L/\delta)/h^2, \tag{14}$$

then stage ℓ sign-resolves d whenever $|g(d; \theta)| \ge h$.

Proof. Apply Proposition 3.1 at each stage with failure probability δ/L and take a union bound. Then $|\hat{g}_{N_\ell} - g| \le \beta_\ell$ holds simultaneously for all stages with probability at least $1 - \delta$. The stated lower bound on N_ℓ implies $2\beta_\ell \le h$. If $g \ge h$, then $\hat{g}_{N_\ell} - \beta_\ell \ge g - 2\beta_\ell \ge 0$; if $g \le -h$, then $\hat{g}_{N_\ell} + \beta_\ell \le g + 2\beta_\ell \le 0$. Hence the interval lies entirely on one side of zero. □

The two guarantees serve different parts of the measurement design. Proposition 3.1 controls the magnitude error of AdaptGain and therefore of its positive part, while Corollary 3.2 shows why tasks clearly above or below the zero boundary can be resolved with fewer pairs. Tasks close to zero require more evidence, which is the reason PACE can use staged rather than uniform measurement budgets.

3.2 From AdaptGain to selection utility

The first stage gives PACE a measured adaptation signal; task selection also needs to place that signal in the learner's current training region and avoid repeatedly concentrating on the same part of the descriptor space. At ecology round ρ, let θ_ρ be the current learner and let the task interface provide a finite measured candidate set $\mathcal{C}_\rho \subset \mathcal{D}$. For a fixed pool, $\mathcal{C}_\rho$ is the pool. For a mutable space, it contains current proposals together with retained descriptors scheduled for remeasurement. Paired evaluation gives $\widehat{R}_{\mathrm{post},\rho}(d)$, $\widehat{R}_{\mathrm{static},\rho}(d)$, and AdaptGain $\widehat{g}_\rho(d)$ for every measured $d \in \mathcal{C}_\rho$. The adaptation term used by PACE is the round-specific positive estimate

$$\widehat{g}_{\rho,+}(d) = \max\{\widehat{g}_\rho(d), 0\}, \tag{15}$$

which is the empirical counterpart of g_+ in Eq. 11. PACE next measures whether the task lies near the learner's current Post frontier,

$$\widehat{\psi}_{F,\rho}(d) = \exp\left(-\frac{(\widehat{R}_{\mathrm{post},\rho}(d) - r_F)^2}{2\sigma_F^2}\right), \tag{16}$$

where $r_F \in [0,1]$ is the target Post level and $\sigma_F > 0$ controls the width. Static performance gives a complementary saturation signal. Since $R_{\mathrm{post}} \le 1$,

$$g_+(d;\theta) \le 1 - R_{\mathrm{static}}(d;\theta), \tag{17}$$

so the possible positive contribution of adaptation shrinks as Static performance approaches one. PACE therefore uses

$$\widehat{s}_{S,\rho}(d) = [\widehat{R}_{\mathrm{static},\rho}(d) - r_S]_+, \tag{18}$$

where $r_S \in [0,1]$ is the Static-saturation threshold. The remaining term controls descriptor breadth. Let $\mathcal{H}_\rho$ be the descriptors selected for learner training through round ρ, and let $v : \mathcal{D} \to \mathbb{R}^{p_v}$ be a fixed descriptor embedding. Define descriptor diversity by

$$\Delta_\rho(d) = \begin{cases} 0, & \mathcal{H}_\rho = \varnothing, \\ \min\limits_{d' \in \mathcal{H}_\rho} \|v(d) - v(d')\|_2, & \mathcal{H}_\rho \neq \varnothing. \end{cases} \tag{19}$$

Because the four components have different scales, PACE standardizes each over $\mathcal{C}_\rho$. For any scalar component x, let $\mu_{\rho,x}$ and $\sigma_{\rho,x}$ be its candidate-set mean and sample standard deviation, and define

$$Z_\rho[x](d) = \begin{cases} 0, & \sigma_{\rho,x} \le \epsilon_{\mathrm{std}}, \\ \mathrm{clip}\left(\dfrac{x(d) - \mu_{\rho,x}}{\sigma_{\rho,x}}, -3, 3\right), & \sigma_{\rho,x} > \epsilon_{\mathrm{std}}, \end{cases} \tag{20}$$

where $\epsilon_{\mathrm{std}} > 0$ is a numerical tolerance. The resulting *PACE selection utility* is

$$U_\rho(d) = w_g Z_\rho[\widehat{g}_{\rho,+}](d) + w_F Z_\rho[\widehat{\psi}_{F,\rho}](d) - w_S Z_\rho[\widehat{s}_{S,\rho}](d) + w_D Z_\rho[\Delta_\rho](d),$$

with nonnegative weights w_g, w_F, w_S, w_D. The four terms have distinct roles: positive AdaptGain measures adaptation dependence, the Post-frontier score locates the learner's current performance region, the Static-saturation penalty downweights tasks that are already easy without adaptation, and descriptor diversity spreads selection across the task space.

3.3 Closed-loop ecology

Freshly measured candidates use the same selection utility U_ρ in both task interfaces; the interfaces differ only in how that utility is turned into future training tasks. In a fixed pool, let $n_\rho(d)$ be the number of times d has been selected through round ρ and define the under-exposure score

$$b_\rho(d) = (1 + n_\rho(d))^{-1/2}. \tag{21}$$

Descriptor diversity measures geometric novelty, whereas b_ρ prevents repeated selection of the same pool elements. The pool score is

$$W_\rho^{\mathrm{pool}}(d) = U_\rho(d) + w_E Z_\rho[b_\rho](d), \tag{22}$$

where $w_E \geq 0$. Let $\overline{W}_\rho$ be Eq. 20 applied to W_ρ^{pool} over $\mathcal{C}_\rho$. With temperature $\tau > 0$,

$$q_{\rho+1}(d) = \frac{\exp(\overline{W}_\rho(d)/\tau)}{\sum_{d' \in \mathcal{C}_\rho} \exp(\overline{W}_\rho(d')/\tau)}, \qquad d \in \mathcal{C}_\rho. \tag{23}$$

Training descriptors are sampled from this distribution without replacement. A mutable task space cannot remeasure every previously useful descriptor at every round, so PACE keeps a bounded set of measured anchors. Let $\mathcal{F}_{\rho-1}$ be the retained set before round ρ, define the normalized selection utility $P_\rho(d) = Z_\rho[U_\rho](d)$, and store each measured entry as

$$e_i = (d_i, \widehat{P}_i, \widehat{g}_i, \rho_i), \qquad \widehat{P}_i = P_{\rho_i}(d_i), \qquad \widehat{g}_i = \widehat{g}_{\rho_i}(d_i), \tag{24}$$

where ρ_i is its latest measurement round. Old measurements become less representative as the learner changes. To make this dependence explicit, suppose population adaptation dependence changes by at most κ per ecology round, $\sup_{d \in \mathcal{D}} |g_{\rho+1}(d) - g_\rho(d)| \leq \kappa$, and the AdaptGain measurement at ρ_i has error $|\widehat{g}_i - g_{\rho_i}(d_i)| \leq \varepsilon_i^{\mathrm{meas}}$. Then

$$|\widehat{g}_i - g_\rho(d_i)| \leq \varepsilon_i^{\mathrm{meas}} + \kappa(\rho - \rho_i), \tag{25}$$

by the triangle inequality and telescoping over the intervening rounds. Equation 25 quantifies the drift of the stored AdaptGain component and motivates a freshness penalty when an anchor is reused. PACE therefore assigns each retained anchor the *reuse score*

$$V_{\rho,i} = \widehat{P}_i + \eta_g \widehat{g}_i - \eta_{\mathrm{age}}(\rho - \rho_i), \tag{26}$$

with $\eta_g, \eta_{\mathrm{age}} \geq 0$. Current measurements replace older entries for the same descriptor and add new entries; for archive capacity M_F, $\mathcal{F}_\rho$ keeps the M_F entries with the largest reuse scores. If $\mathcal{I}_\rho$ indexes $\mathcal{F}_\rho$, anchors are sampled with

$$p_{\rho,i} = \frac{\exp(V_{\rho,i}/\tau)}{\sum_{j \in \mathcal{I}_\rho} \exp(V_{\rho,j}/\tau)}. \tag{27}$$

Let $u : \mathcal{D} \to \mathbb{R}^{p_u}$ be the mutation encoding and $\Pi_{\mathcal{D}} : \mathbb{R}^{p_u} \to \mathcal{D}$ map an encoded point to a valid descriptor. From anchor d_i,

$$\widetilde{u} = u(d_i) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma_d^2 I_{p_u}), \qquad d' = \Pi_{\mathcal{D}}(\widetilde{u}), \tag{28}$$

where $\sigma_d > 0$ and I_{p_u} is the $p_u \times p_u$ identity matrix. This defines the proposal kernel $\mathcal{K}(G \mid d_i) = \Pr(d' \in G \mid d_i)$ for measurable $G \subseteq \mathcal{D}$. With base distribution q_{base},

$$q_{\rho+1}(G) = \begin{cases} q_{\mathrm{base}}(G), & \mathcal{F}_\rho = \varnothing, \\ \displaystyle\sum_{i \in \mathcal{I}_\rho} p_{\rho,i}\, \mathcal{K}(G \mid d_i), & \mathcal{F}_\rho \neq \varnothing. \end{cases} \tag{29}$$

Thus U_ρ scores current measurements, while $V_{\rho,i}$ governs reuse of earlier measurements as the learner changes. The full loop starts from learner parameters θ_0, $q_1 = q_{\mathrm{base}}$, and $\mathcal{H}_0 = \varnothing$; mutable interfaces also set $\mathcal{F}_0 = \varnothing$, and fixed pools initialize $n_0(d) = 0$. Let R be the number of ecology rounds, B the number of training descriptors per round, and $\mathcal{A}$ the learner's training procedure. At round ρ,

$$\mathcal{B}_\rho \sim \mathrm{Sample}_B(q_\rho), \qquad \theta_\rho = \mathrm{Train}_{\mathcal{A}}(\theta_{\rho-1}, \mathcal{B}_\rho), \tag{30}$$

where Sample_B draws B descriptors according to the interface-specific distribution. The training history becomes $\mathcal{H}_\rho = \mathcal{H}_{\rho-1} \cup \mathcal{B}_\rho$, and fixed-pool counts are updated accordingly. The task interface constructs $\mathcal{C}_\rho$, and PACE measures

$$\widehat{\mathcal{M}}_\rho = \left\{ \left(d, \widehat{R}_{\mathrm{post},\rho}(d), \widehat{R}_{\mathrm{static},\rho}(d), \widehat{g}_\rho(d)\right) : d \in \mathcal{C}_\rho \right\}. \tag{31}$$

These measurements produce the selection or reuse scores above and therefore $q_{\rho+1}$. The complete loop is

$$\boxed{q_\rho \to \mathcal{B}_\rho \to \theta_\rho \to \widehat{\mathcal{M}}_\rho \to q_{\rho+1}}. \tag{32}$$

After R rounds, evaluation uses θ_R with the learner's native episode-local adaptation mechanism.

Table 2: Matched-Post response, averaged over four controlled environments. Differences are high-AdaptGain minus low-AdaptGain groups.

Learner	Matched $\lvert\Delta\text{Post}\rvert$	Δ pre-AdaptGain	Δ Post	Δ AdaptGain	Post-positive envs
belief-style	0.0010	0.0134	+0.0004	+0.0002	2/4
gradient-meta	0.0029	0.0306	+0.0037	+0.0042	4/4
in-context	0.0031	0.0383	+0.0064	+0.0073	4/4
discover	0.0033	0.0456	+0.0107	+0.0113	4/4

Table 3: Controlled results in the standard regime at $K = 8$. Adaptive columns average the three adaptation-capable learners; belief-style is shown separately.

Teacher	Adaptive Post	AdaptGain	Coverage	Belief Post	Belief AdaptGain
PACE	0.6441	0.0179	0.6273	0.6235	-0.0018
AdaptGain-only	0.6439	0.0179	0.5866	0.6236	-0.0016
Post frontier	0.6426	0.0172	0.6199	0.6238	-0.0018
Hardness-only	0.6428	0.0171	0.5548	0.6235	-0.0018
Learning-progress	0.6429	0.0168	0.5023	0.6244	-0.0020
ACCEL-style editing	0.6435	0.0174	0.6025	0.6247	-0.0017
DRED-style regularization	0.6431	0.0174	0.5938	0.6245	-0.0018
PAIRED-style regret	0.6429	0.0170	0.5306	0.6241	-0.0020
PLR-style replay	0.6432	0.0176	0.8876	0.6239	-0.0017
Uniform	0.6429	0.0174	0.7386	0.6243	-0.0017

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

• **Controlled Setup.** The controlled suite contains correlated bandits, latent-context tasks, compositional grids, and sequential sessions with drift or switching. Their descriptors vary difficulty, observation and reward noise, latent shift, switch probability, and action correlation; we evaluate standard, high-drift, and high-noise regimes. The four learners expose different episode-local mechanisms: discover accumulates local evidence, in-context uses a learned prefix representation, gradient-meta performs a fast task-specific update, and belief-style weakens the local-state readout. We use $K \in \{5, 8\}$, a frozen suffix of $H = 8$, 192 candidate descriptors per cell, 10 ecology rounds, and 24 training descriptors per round. Within each teacher comparison, learner initialization and training budget are held fixed. Adaptive executions supply training data; Static executions are used for measurement. PACE measures each candidate with eight matched adaptive/Static pairs and uses $w_g = 1.5$, $w_F = 0.3$, $w_S = 0.5$, $w_D = 0.2$, $w_E = 0.03$, $r_F = 0.65$, $\sigma_F = 0.10$, $r_S = 0.70$, and $\tau = 1$. Final controlled evaluation uses 160 unseen descriptors, and all reported controlled results use five independent end-to-end seeds.

• **Comparisons and Matched-Post Test.** Controlled comparisons include Uniform, hardness-only, learning-progress, AdaptGain-only, a Post frontier, and common-pipeline versions of PAIRED (Dennis et al., 2020), PLR (Jiang et al., 2021b), ACCEL (Parker-Holder et al., 2022), and DRED (Garcin et al., 2024). The Post frontier keeps the same Post-frontier score, descriptor-diversity term, under-exposure term, and sampling rule, but omits AdaptGain and the Static-saturation penalty; AdaptGain-only uses positive AdaptGain alone. For the matched-Post test, each environment–learner cell contains 1,800 measured candidates. We match lower- and upper-quartile AdaptGain groups one-to-one without replacement on pre-retraining Post using a .020 caliper. Both branches start from the same checkpoint, receive one ecology-round-equivalent retraining budget, and are evaluated on the same 180 fresh descriptors. Post is the primary held-out metric, AdaptGain is the paired empirical estimate $\widehat{g}$, and Coverage is the fraction of a finite pool selected at least once. Coverage is reported only as a diagnostic of task-space breadth; it is not an input to the PACE selection utility.

• **Native Setup and Compute.** The native suite covers Procgen (Cobbe et al., 2020), MiniGrid (Chevalier-Boisvert et al., 2023), XLand-MiniGrid (Nikulin et al., 2024), CarRacing (Brockman et al., 2016), and KuaiSim/KuaiRand (Zhao et al., 2023; Gao et al., 2022). Learners include feed-forward PPO (Schulman et al., 2017), recurrent RL^2 (Duan et al., 2016), MAML (Finn et al., 2017), PEARL (Rakelly et al., 2019), VariBAD (Zintgraf et al., 2020), and ULEE's in-context learner

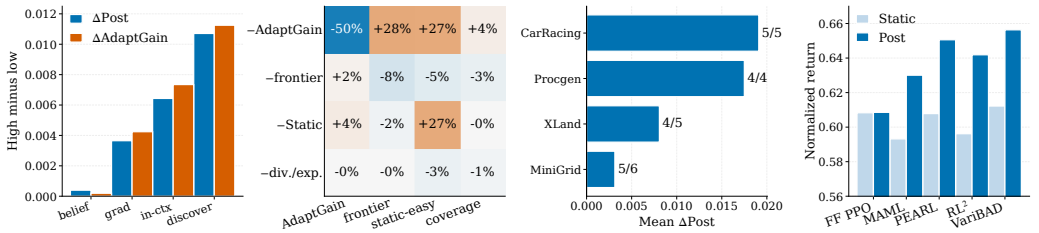


Figure 3: Controlled and native diagnostics: (a) matched-Post response; (b) component ablations; (c) native Post margins; (d) KuaiSim/KuaiRand Static and Post returns.

(Pappalardo, 2026). Static freezes the corresponding recurrent, fast-parameter, context, belief, or history-conditioned update; feed-forward PPO is the no-state control. Evaluation uses held-out level seeds, tasks, tracks, or user/session tasks according to the benchmark, with the same within-benchmark normalization for PACE and its comparator. External comparisons use CLUTR for CarRacing (Azad et al., 2023); ULEE and DRED for XLand-MiniGrid (Pappalardo, 2026; Garcin et al., 2024); and compatible DEGen+MNA, NCC, DRED, ACCEL, PLR, or PAIRED-family baselines for MiniGrid and Procgen (Mead et al., 2025; Monette et al., 2025; Garcin et al., 2024; Parker-Holder et al., 2022; Jiang et al., 2021b; Dennis et al., 2020). Controlled runs use Ubuntu with Python 3.12.3. Accelerator-backed native jobs run one job per NVIDIA RTX PRO 6000 GPU with 96 GiB memory. Appendix A provides the full controlled protocol, random seeds, checkpointing rules, native interfaces, comparator sets, and supplementary analyses.

4.2 Experimental Results

• **Matched-Post Response.** The matched-Post test asks whether AdaptGain carries training-relevant information after current Post has been controlled. Table 2 shows a residual Post mismatch of only 0.0010–0.0033, compared with a pre-retraining AdaptGain separation of 0.0134–0.0456. Figure 3(a) shows the corresponding retraining response: belief-style is nearly flat, while the effect increases from gradient-meta and in-context to discover. Under the same retraining dose, the high-AdaptGain groups of all three adaptation-capable learners improve in all four environments. Across these 12 environment–learner cells, they gain $+0.00693 \pm 0.00078$ Post and $+0.00761 \pm 0.00024$ AdaptGain over their matched low-AdaptGain groups, with positive effects in all five end-to-end seeds. Belief-style changes little. Since the groups begin at nearly the same Post, the difference in retraining response is associated with their pre-retraining AdaptGain rather than a meaningful difference in starting performance alone.

• **Closed-Loop Selection.** Table 3 reports the absolute held-out outcomes in the standard $K = 8$ setting. PACE has the highest adaptive Post, with a seed-matched margin of $+0.00147 \pm 0.00041$ over the Post frontier; all five seed-level differences are positive. Figure 2(a) places this Post margin against descriptor coverage, while Figure 2(b) does the same for AdaptGain. AdaptGain-only is close to PACE in both Post and AdaptGain but covers less of the descriptor pool (0.5866 versus 0.6273). PLR-style replay and Uniform cover substantially more tasks but have smaller Post margins. AdaptGain-only captures most of the return difference in this setting, while the Post-frontier, descriptor-diversity, and under-exposure terms mainly change how training is distributed across the pool. For belief-style, teacher choice changes Post little and AdaptGain stays near zero.

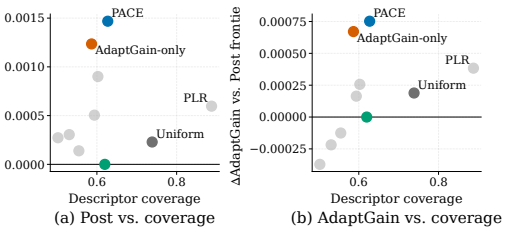


Figure 2: Controlled trade-offs at $K = 8$: (a) Post margin and (b) AdaptGain margin versus descriptor coverage.

• **Ablations and Boundaries.** Figure 3(b) separates the role of the four score components. Removing the AdaptGain term roughly halves the selected AdaptGain ($0.0203 \rightarrow 0.0102$); removing the Post frontier lowers frontier alignment ($0.6452 \rightarrow 0.5926$); removing the Static penalty increases statically easy selections ($0.1136 \rightarrow 0.1449$); and removing diversity/exposure reduces coverage ($0.6442 \rightarrow 0.6349$). Final Post changes little across these one-component removals; over this training horizon, their clearest effect is on the composition of the selected task set. Table 4 tests a different question: whether positive AdaptGain remains useful when the information available during the

Table 4: Information-regime results across write horizons. Δ columns are PACE minus the Post frontier; "Positive cells" counts adaptive cells with positive ΔPost.

Regime	K	PACE Post	PACE AdaptGain	ΔPost	Positive cells	ΔAdaptGain	Belief AdaptGain
standard	5	0.6397	0.0142	+0.0002	6/12	+0.0004	-0.0025
standard	8	0.6441	0.0179	+0.0015	9/12	+0.0008	-0.0018
high-drift	5	0.6153	0.0250	+0.0013	11/12	+0.0009	-0.0007
high-drift	8	0.6217	0.0316	+0.0014	12/12	+0.0013	+0.0004
high-noise	5	0.6236	0.0097	+0.0004	9/12	+0.0005	-0.0077
high-noise	8	0.6259	0.0125	-0.0006	6/12	+0.0004	-0.0068

Table 5: Native benchmark results with compatible external comparators.

Benchmark	Comparable cells	PACE highest	Mean ΔPost	Mean PACE AdaptGain
CarRacing	5	5/5	+0.019	0.034
MiniGrid UED	6	5/6	+0.003	0.034
Procgen	4	4/4	+0.018	0.033
XLand-MiniGrid	5	4/5	+0.008	0.036

prefix changes. Increasing K raises AdaptGain in all three regimes, but the return effect does not move in the same way. At $K = 8$, PACE is above the Post frontier in all 12 adaptive high-drift cells, and all five seed-level Post differences are positive. Under high noise, the mean Post margin is -0.0006 even though ΔAdaptGain remains positive; only one of five seed-level Post differences is positive. Belief-style stays near zero or negative in five of the six regime–horizon cells. In the high-noise setting, AdaptGain remains positive even though the task-selection advantage disappears.

• **Native Benchmarks.** Table 5 summarizes the 20 benchmark–learner cells with compatible external comparators. PACE has the highest mean Post in 18 cells, with a weighted mean normalized margin of $+0.0112$. Mean PACE AdaptGain is comparatively stable across the four benchmark families (0.033–0.036), while Figure 3(c) shows that the external-comparator margin varies more: it is $+0.019$ on CarRacing, $+0.018$ on Procgen, $+0.008$ on XLand-MiniGrid, and $+0.003$ on MiniGrid. The two exceptions are MiniGrid/MAML, where DEGen+MNA is higher by 0.002, and XLand-MiniGrid/RL^2, where ULEE is higher by 0.003. Figure 3(d) uses KuaiSim/KuaiRand for the intervention check rather than an external-teacher comparison. Feed-forward PPO has almost no adaptive–Static separation (0.00024), whereas MAML, PEARL, recurrent RL^2, and VariBAD show AdaptGain of 0.0369, 0.0427, 0.0457, and 0.0441. The adaptive–Static separation is also present under this different task interface and across several forms of episode-local state.

• **Measurement Cost.** The empirical 4/8/16-pair schedule uses 9.42 pairs per descriptor on average, versus 16 under fixed-budget measurement, reducing paired evaluation by 41.1%. Its cell-level AdaptGain estimate differs from fixed-16 evaluation by only 1.83×10^{-4}. Belief-style uses 5.27 pairs on average and resolves 94.0% of descriptors by 8 pairs, while the three adaptation-capable learners use about 10.8 pairs and resolve roughly half by that point; ambiguous cases continue to 16 pairs. This matches Corollary 3.2: descriptors farther from the zero boundary require fewer pairs. Measuring one descriptor with N pairs requires $2N$ executions of horizon $K + H$, or $O(N(K + H))$ environment steps. For candidate set $\mathcal{C}_\rho$, the measurement cost is therefore $O(|\mathcal{C}_\rho|\, \bar{N}\, (K + H))$, where $\bar{N}$ is the average number of pairs used. With $\bar{N} = 9.42$, the staged rule reduces this dominant measurement cost by the same 41.1% without changing the learner-training budget.

5 Conclusions and Limitations

PACE measures how much a task depends on within-episode adaptation through matched adaptive and Static executions, and uses the resulting AdaptGain in a closed-loop task selector. Controlled and native experiments show that AdaptGain provides information beyond Post, while the high-noise regime shows that positive adaptation dependence does not always imply curriculum value. PACE requires a well-defined episode-local update and paired evaluation, though staged measurement reduces this cost. Its estimand captures the total downstream effect of enabling adaptation and does not separate local-state changes from trajectory changes induced during the adaptive prefix.

AI USE STATEMENT

ChatGPT was used as a writing and review aid during the preparation of this manuscript. Its use was limited to improving the wording and readability of the text, reorganizing parts of the manuscript for better flow, helping identify potentially relevant literature, and checking for inconsistencies across sections. All suggested revisions were reviewed by the authors before being incorporated. References found with the assistance of ChatGPT were checked independently using Google Scholar prior to citation. The authors retain full responsibility for the scientific content, the accuracy of the references, and the final version of the manuscript.

REPRODUCIBILITY STATEMENT

To support reproducibility, the learner architecture, training objective, training budget, and evaluation procedure are held fixed within each comparison. PACE hyperparameters and nondefault baseline settings are selected using development configurations disjoint from the evaluation tasks and are fixed before held-out evaluation; test metrics are not used for model selection. Controlled results use five independent end-to-end seeds, with the same seed used within each paired teacher comparison. The main text specifies the intervention, estimator, selection rule, training budgets, and matched-Post protocol, while Appendix A provides the descriptor ranges, PACE hyperparameters, random seeds, checkpointing rules, native intervention definitions, comparator settings, and supplementary analyses.

REFERENCES

Abdus Salam Azad, Izzeddin Gur, Jasper Emhoff, Nathaniel Alexis, Aleksandra Faust, Pieter Abbeel, and Ion Stoica. CLUTR: Curriculum learning via unsupervised task representation learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 1361–1395, 2023.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and J.K. Terry. MiniGrid & MiniWorld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 73383–73394, 2023.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 2048–2056, 2020.

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. In *Advances in Neural Information Processing Systems*, volume 33, pp. 13049–13061, 2020.

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1126–1135, 2017.

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*, pp. 1515–1528, 2018.

Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. KuaiRand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pp. 3953–3957, 2022.

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 (ICML)*, volume 235, pp. 15006–15036, 2024.

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pp. 1311–1320, 2017.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

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, pp. 1884–1897, 2021a.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 4940–4950, 2021b.

M. Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, D. J. Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 43057–43083, 2023.

Ricardo Luna Gutierrez and Matteo Leonetti. Information-theoretic task selection for meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20532–20542, 2020.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, 2020.

Harry Mead, Bruno Lacerda, Jakob Foerster, and Nick Hawes. Improving regret approximation for unsupervised dynamic environment generation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 168734–168771, 2025.

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

Nathan Monette, Alistair Letcher, Michael Beukman, Matthew T. Jackson, Alexander Rutherford, Alexander D. Goldie, and Jakob N. Foerster. An optimisation framework for unsupervised environment design. *arXiv preprint arXiv:2505.20659*, 2025.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, Artem Agarkov, Viacheslav Sinii, and Sergey Kolesnikov. XLand-MiniGrid: Scalable meta-reinforcement learning environments in JAX. In *Advances in Neural Information Processing Systems*, volume 37, pp. 43809–43835, 2024.

Octavio Pappalardo. Unsupervised learning of efficient exploration: Pre-training adaptive policies via self-imposed goals. In *International Conference on Learning Representations (ICLR)*, 2026.

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 (ICML)*, pp. 17473–17498, 2022.

Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 5331–5340, 2019.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

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

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 *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017.

Jane X. Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Qi Cheems Wang, Zehao Xiao, Yixiu Mao, Yun Qu, Jiayi Shen, Yiqin Lv, and Xiangyang Ji. Model predictive task sampling for efficient and robust adaptation. *Nature Communications*, 2026.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. POET: Open-ended coevolution of environments and their optimized solutions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 142–151, 2019.

Kesen Zhao, Shuchang Liu, Qingpeng Cai, Xiangyu Zhao, Ziru Liu, Dong Zheng, Peng Jiang, and Kun Gai. KuaiSim: A comprehensive simulator for recommender systems. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44880–44897, 2023.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: A very good method for bayes-adaptive deep RL via meta-learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

A Additional Experimental Details and Analyses

A.1 Controlled protocol and compute

The controlled descriptor has six coordinates: difficulty, observation noise, reward noise, latent shift or context drift, switch probability, and action correlation. Their standard ranges are $[.15, .95]$, $[.02, .48]$, $[.01, .28]$, $[0, 1]$, $[0, .38]$, and $[0, .9]$, respectively, with coordinates sampled independently. High-drift cells restrict shift/drift to $[.72, 1]$ and switching to $[.23, .40]$; high-noise cells use observation noise $[.34, .52]$ and reward noise $[.19, .32]$, with other coordinates drawn from the standard ranges. The normalized coordinates form the descriptor embedding $v(d) \in \mathbb{R}^6$. We evaluate $K \in \{5, 8\}$ with frozen horizon $H = 8$. Each controlled cell contains 192 candidates. Training starts from a uniform teacher, runs for 10 ecology rounds, and samples 24 descriptors without replacement per round, with descriptors returning to the pool in later rounds. Adaptive executions provide learner-training data, while Static executions are used only for paired measurement.

The four controlled learners differ in their episode-local mechanism: discover accumulates episode-local evidence, in-context uses a learned prefix representation, gradient-meta performs a fast task-specific update, and belief-style weakens the local-state readout. All teachers use the same evolution-strategy update family (Salimans et al., 2017), learner initialization, architecture, objective, and

Table 6: Native benchmark interfaces, held-out units, and evaluated learners.

Environment	Held-out unit	Evaluated learners
MiniGrid UED	level seed	PPO, RL^2, ULEE in-context, MAML, PEARL, VariBAD
XLand-MiniGrid	goal/environment task	PPO, RL^2, ULEE in-context, MAML, VariBAD
Procgen	level seed	PPO, RL^2, MAML, VariBAD
CarRacing	track instance	PPO, RL^2, MAML, PEARL, VariBAD
KuaiSim/KuaiRand	user/session task	PPO, RL^2, MAML, PEARL, VariBAD

Table 7: One-component ablations in the standard $K = 5$ setting.

Variant	Post	AdaptGain	Coverage	Sel. AdaptGain	Frontier	Static-easy
Full PACE	0.6356	0.0100	0.6442	0.0203	0.6452	0.1136
w/o AdaptGain	0.6353	0.0097	0.6685	0.0102	0.8247	0.1447
w/o Post frontier	0.6355	0.0099	0.6231	0.0208	0.5926	0.1080
w/o Static penalty	0.6356	0.0100	0.6428	0.0212	0.6316	0.1449
w/o diversity/exposure	0.6356	0.0100	0.6349	0.0202	0.6436	0.1105

learner-update budget within each comparison. Finite-pool PACE uses $w_g = 1.5$, $w_F = 0.3$, $w_S = 0.5$, $w_D = 0.2$, $w_E = 0.03$, $r_F = 0.65$, $\sigma_F = 0.10$, $r_S = 0.70$, $\tau = 1$, and $\epsilon_{\text{std}} = 10^{-8}$; these values are selected on task-disjoint development descriptors and then fixed. Main measurements use eight matched adaptive/Static pairs per descriptor, while the measurement-efficiency study uses cumulative 4/8/16-pair stages with a development-fixed empirical stopping rule. The five end-to-end seeds are $\{101, 203, 307, 409, 503\}$, with paired teacher comparisons sharing the same seed. Final evaluation uses 160 unseen descriptors per environment and regime from streams disjoint from training candidates.

For matched-Post experiments, each environment–learner cell contains 1,800 measured candidates. Lower- and upper-quartile AdaptGain groups are matched one-to-one without replacement on pre-retraining Post using a .020 caliper, leaving 20–64 matched pairs. Both branches start from the same checkpoint, receive one ecology-round-equivalent retraining budget, and are evaluated on the same 180 fresh descriptors. The Post frontier retains the Post-frontier score, descriptor diversity, under-exposure, and sampling rule but omits AdaptGain and the Static-saturation penalty; AdaptGain-only uses positive AdaptGain alone. Other baselines are Uniform, hardness-only, learning-progress, and common-pipeline versions of PAIRED (Dennis et al., 2020), PLR (Jiang et al., 2021b), ACCEL (Parker-Holder et al., 2022), and DRED (Garcin et al., 2024). Published or default configurations are used when available; otherwise, settings are selected from at most 12 task-disjoint development configurations and fixed before test evaluation. Controlled runs use Ubuntu with Python 3.12.3. Accelerator-backed native jobs use one NVIDIA RTX PRO 6000 GPU with 96 GiB memory per job; native checkpointing, framework, and optimizer settings follow the corresponding public implementations when available.

A.2 Native interfaces and comparators

Table 6 summarizes the held-out unit and learner set for each native interface. The Static intervention follows the learner's episode-local state: it holds recurrent state fixed for RL^2, skips the MAML inner update, fixes the inferred context or belief state for PEARL and VariBAD, and disables the history-conditioned state update for ULEE. Feed-forward PPO has no learned episode-local state and serves as the no-state control. Native outcomes are normalized within each benchmark using the same transformation for PACE and its comparator. Comparator sets follow the benchmark interface: CLUTR for CarRacing (Azad et al., 2023); DEGen+MNA, NCC-family, DRED, ACCEL, and PLR/PAIRED-family methods for MiniGrid (Mead et al., 2025; Monette et al., 2025; Garcin et al., 2024; Parker-Holder et al., 2022; Jiang et al., 2021a;b; Dennis et al., 2020); ACCEL and PLR-family methods for Procgen; and ULEE and DRED for XLand-MiniGrid (Pappalardo, 2026; Garcin et al., 2024). Published or default settings are used when available; otherwise, configurations are selected from at most 12 task-disjoint development settings and fixed before test evaluation.

Table 8: Seed-level paired effects over five independent replications.

Comparison	Outcome	Mean $\Delta \pm$ s.d.	Positive seeds
Matched-Post (adaptive)	Post	$+0.00693 \pm 0.00078$	5/5
Matched-Post (adaptive)	AdaptGain	$+0.00761 \pm 0.00024$	5/5
PACE − Post frontier, standard	Post	$+0.00147 \pm 0.00041$	5/5
PACE − Post frontier, standard	AdaptGain	$+0.00075 \pm 0.00034$	5/5
PACE − Post frontier, high drift	Post	$+0.00141 \pm 0.00057$	5/5
PACE − Post frontier, high noise	Post	-0.00062 ± 0.00112	1/5

Table 9: Staged paired-measurement cost versus fixed 16-pair evaluation.

Learner	Fixed pairs	Staged pairs	Reduction	Early stop	Mean $\lvert\Delta\widehat{g}\rvert$
belief-style	16.0	5.27	67.1%	0.940	2.70×10^{-4}
gradient-meta	16.0	10.75	32.8%	0.504	2.77×10^{-4}
in-context	16.0	10.90	31.9%	0.488	1.36×10^{-4}
discover	16.0	10.75	32.8%	0.504	4.78×10^{-5}

KuaiSim/KuaiRand is used only for the intervention check and has no matched external teacher comparator in our evaluation.

A.3 Supplementary analyses

Table 7 gives the complete one-component ablation in the standard $K = 5$ setting. The four variants set w_g, w_F, w_S, or (w_D, w_E) to zero, respectively. Selected AdaptGain is the mean $\widehat{g}_{\rho,+}$ over selected descriptors, frontier alignment is the mean $\widehat{\psi}_{F,\rho}$, Static-easy is the fraction with $\widehat{R}_{\mathrm{static},\rho} > r_S$, and Coverage is the fraction of the finite pool selected at least once. Each removal mainly affects the quantity controlled by that term: removing AdaptGain lowers selected AdaptGain from 0.0203 to 0.0102; removing the Post frontier lowers frontier alignment from 0.6452 to 0.5926; removing the Static penalty raises the Static-easy fraction from 0.1136 to 0.1449; and removing diversity/exposure reduces Coverage from 0.6442 to 0.6349. Final Post changes by at most 3×10^{-4}. Over this training horizon, these auxiliary terms therefore act mainly by changing which tasks are selected, while the AdaptGain term most directly controls selection toward tasks with large positive adaptation dependence. Table 8 reports seed-level paired effects for both the matched-Post test and the PACE–Post-frontier comparison. Seed identity is preserved within each comparison, so the reported differences are computed between matched runs rather than independently sampled cells. Table 9 gives the corresponding per-learner breakdown for staged measurement. Fixed-budget evaluation uses 16 pairs for every descriptor, whereas the staged procedure stops at 4, 8, or 16 pairs according to the development-fixed empirical rule. "Early stop" denotes the fraction resolved by 8 pairs, and the final column reports the deviation of the staged cell-level AdaptGain estimate from fixed-16 evaluation. The breakdown shows where the average measurement saving reported in the main text comes from: descriptors with clear adaptation dependence terminate early, while uncertain cases retain the full 16-pair budget.