TRAINING STABILITY UNDER ADAPTIVE CURRICULA

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised Environment Design (UED) approaches train reinforcement learning (RL) agents on diverse curricula that may evolve with their capabilities. While prior work has largely focused on how these curricula are constructed, we study whether their students remain trainable under the resulting distributions. Across multiple UED mechanisms, environment complexities, and extended training horizons, we identify recurring learner-side pathologies, including parameter growth, degraded value–outcome ordering, and loss of representational diversity, that accompany stagnation and performance degradation. These findings connect limitations in UED to optimization failures previously identified in conventional deep RL, while showing how adaptive curricula couple the learner's optimization state to its future training distribution. Controlled interventions show that mitigating selected pathologies improves learning. Stabilizing the learner improves long-horizon performance and zero-shot generalization without changing the curriculum rule or increasing environment interaction. Our results establish learner trainability as a complementary design principle for UED. Effective curricula require students that remain capable of learning from the distributions they produce.

"The teacher can advance learning only by influencing what the student does to learn[1]*."*
— Herbert A. Simon

1 INTRODUCTION

Unsupervised Environment Design (UED) trains deep RL agents on environment distributions that are constructed and adapted throughout training to promote generalization to unseen tasks. Rather than relying on a fixed task set, UED methods generate, select, edit, or replay environments throughout training (Dennis et al., 2020; Jiang et al., 2021b;a; Parker-Holder et al., 2022; Rutherford et al., 2024). Many of these methods adapt the training distribution to the evolving capabilities of a student, directing experience toward tasks estimated to be informative for its current policy. This literature has therefore largely centered on constructing better curricula. We study the complementary question of whether the student remains trainable under the distributions those curricula produce.

Deep RL training is already non-stationary because policy updates alter the state–action occupancy distribution, while bootstrapped targets depend on evolving value estimates and, in actor–critic methods, an evolving policy (Sutton & Barto, 2018; Kumar et al., 2021). Adaptive UED adds a feedback loop in which changes in the student affect which environments are subsequently generated, selected, or replayed, coupling learner optimization to its future data. This interaction may compound pathologies documented in conventional deep RL, including parameter growth, dormant neurons, feature-rank collapse, and unstable critic and gradient dynamics (Nikishin et al., 2022; Lyle et al., 2023; Qin et al., 2024; Nauman et al., 2024; Mayor et al., 2025; Castanyer et al., 2025; Obando-Ceron et al., 2025). Consequently, learner degradation can affect both how experience is used and which experience is encountered next. The curriculum may therefore continue to assign high priority to environments even as the student's capacity to benefit from them declines. Fig. 1 illustrates this tension. Prioritized Level Replay (PLR) achieves higher sampled return than the domain-randomization baseline, but is accompanied by larger gradient norms, higher representation similarity, and weaker value–outcome rank correlation. Better task selection can therefore coexist with greater optimization strain and weaker indicators of learner trainability.

[1]UED instantiates this principle algorithmically. While prior work has focused on giving examples a student can effectively learn from, we study whether it sets up the student to continue to learn reliably in the future.

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.

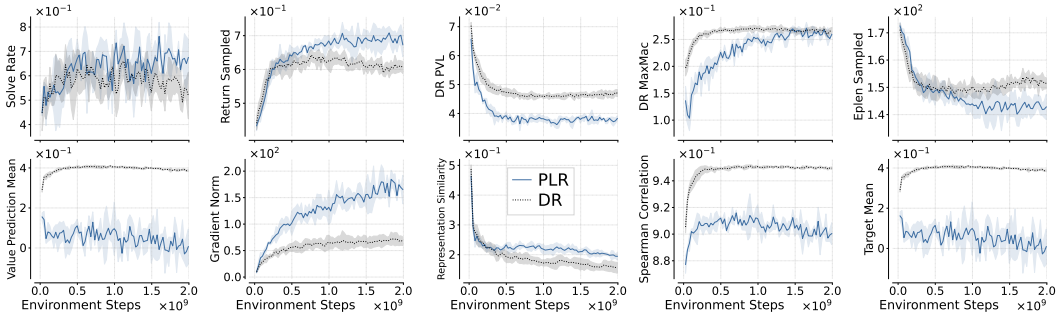

Figure 1: **Adaptive replay reveals a performance–trainability tension.** Compared with domain randomization (DR, Tobin et al. (2017)), PLR achieves higher performance but exhibits larger gradient norms, more aligned representations, and weaker value–outcome rank correlation. Curriculum scores, episode lengths, and value targets further characterize the different training distribution induced by student-dependent replay. We report the interquartile mean after 2B environment steps, aggregated over 5 seeds each; shaded regions indicate 95% stratified bootstrap confidence intervals.

We investigate learner trainability in Kinetix (Matthews et al.), whose heterogeneous physics tasks, graded complexity configurations, and long training horizons provide a demanding testbed. A detailed analysis of PLR (Sec. 3) reveals multiple indicators of declining learner trainability, including degradation in parameter dynamics, value prediction, and representations, that become more pronounced in more complex configurations and over longer training horizons. Complementary to recent work sustaining PPO learning in Kinetix through massive rollout parallelism (Beukman et al., 2026), we examine learner-side degradation and whether it can be mitigated without increasing rollout scale.

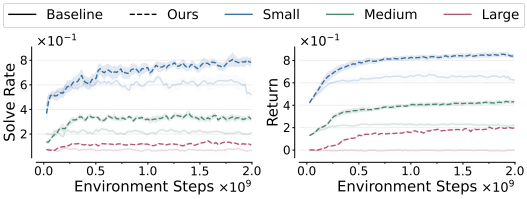

Figure 2: **Learner stabilization improves performance across adaptive UED mechanisms.** Evaluation solve rate and sampled return aggregated over PLR, SFL, ACCEL, and random seeds for **Small**, **Medium**, and **Large** Kinetix configurations. Faint solid curves denote baselines; dashed curves denote stabilized learners.

We evaluate established interventions targeting different aspects of learner optimization[2]. Their mixed outcomes show that not all stabilization interventions are equally effective. Weight clipping and categorical value prediction provide the most consistent individual gains and act at complementary points in the learning process by controlling parameter scale and reducing the sensitivity of the value objective to scalar target errors (Sec. 4). As summarized in Fig. 2, the combined intervention improves performance across multiple curriculum mechanisms, including PLR, SFL (Rutherford et al., 2024), ACCEL (Parker-Holder et al., 2022), and domain randomization, and across environment complexity levels, without changing the curriculum rules or environment-interaction budgets. Its benefits persist over extended training horizons, under limited rollout parallelism, across sequential distribution shifts, and in zero-shot generalization (Sec. 5). These results show that learner-side degradation can prevent existing curricula from realizing their full benefit. Progress in UED therefore depends not only on constructing informative training distributions, but also on keeping students trainable under the distributions they produce.

2 Problem Setting

Actor–critic learning. We consider actor–critic agents that jointly learn a policy $\pi_\theta(a \mid s)$ and value approximation $V_\phi(s)$. Our agents use PPO (Schulman et al., 2017), whose clipped surrogate objective limits the influence of large likelihood-ratio changes within each optimization phase, while the critic is trained against trajectory-based targets $\widehat{V}_t$ using $\mathcal{L}_V(\phi) = \mathbb{E}_t[(V_\phi(s_t) - \widehat{V}_t)^2]$. Training nevertheless remains non-stationary because policy updates alter the experience distribution, while newly collected

[2]Unless stated otherwise, experiments use two billion environment interactions and five random seeds. Curves report the interquartile mean (IQM) (Agarwal et al., 2021), and shaded regions show 95% bootstrap confidence intervals, stratified by algorithm when results are aggregated.

trajectories and evolving value estimates change the targets and advantages used in subsequent updates. PPO clipping does not stabilize these quantities or the network representations across optimization phases. Prior work shows that these dynamics can be accompanied by representation degradation, loss of plasticity, and learning stagnation even in on-policy agents (Moalla et al., 2024; Juliani & Ash, 2024; Mayor et al., 2025; Beukman et al., 2026). See App. G for details.

Adaptive environment curricula and student coupling. UED extends the MDP setting to a level space $\mathcal{L}$, where each level $\ell \in \mathcal{L}$ defines an environment $\mathcal{M}_\ell$ (Dennis et al., 2020). At iteration k, the curriculum induces a distribution $p_k(\ell)$ over levels, and the resulting state–action distribution is $d_k(s,a) = \mathbb{E}_{\ell\sim p_k}[d^{\pi_{\theta_k}}_{\mathcal{M}_\ell}(s,a)]$, where $d^{\pi_{\theta_k}}_{\mathcal{M}_\ell}$ denotes the occupancy measure induced by the current policy in environment $\mathcal{M}_\ell$. A training batch $\mathcal{D}_k$ is collected by sampling levels from p_k and rolling out π_{θ_k}. Let $w_k = (\theta_k, \phi_k)$ denote the student parameters. The student update is written as $w_{k+1} = \mathcal{U}(w_k; \mathcal{D}_k)$, where $\mathcal{U}$ denotes the learning update, while the curriculum evolves according to $p_{k+1} = \mathcal{C}(p_k, \mathcal{H}_{k+1})$, where $\mathcal{C}$ is the curriculum update rule and $\mathcal{H}_{k+1}$ is the training history, including returns, success estimates, or replay statistics generated by the student. Under domain randomization, p_k remains fixed to the generator distribution, although d_k still changes as the policy evolves. Adaptive UED methods instead modify p_k by generating, selecting, editing, or replaying levels using student-dependent signals (Jiang et al., 2021b;a; Parker-Holder et al., 2022; Rutherford et al., 2024). The curriculum therefore determines the student's current data, while the updated student influences its future training distribution. A fixed curriculum rule can consequently produce different training distributions when coupled to different learners. See App. H for details.

3 Diagnosing Optimization Instability in UED

Fig. 1 reveals a performance–trainability tension. Prioritized Level Replay (PLR) outperforms domain randomization (DR), but also exhibits larger gradient norms, more aligned representations, and weaker value–outcome correspondence. This comparison does not establish that adaptive replay causes learner degradation, but it exposes signs of optimization strain that performance alone does not reveal. We focus our detailed diagnosis on PLR (Jiang et al., 2021a;b) for two reasons. First, PLR is a foundational and widely used adaptive UED method. Second, it provides a particularly transparent view of curriculum–student coupling. Its generator proposes levels independently of the student, while explicit student-dependent scores determine which stored levels are replayed. We can therefore examine how replay priorities, the realized training distribution, and learner optimization co-evolve without the additional mechanism of explicit level editing. We later test whether the benefits of stabilization extend to SFL and ACCEL, which construct curricula through learnability-based selection and level editing, respectively.

Benchmark and evaluation protocol. We conduct this analysis in Kinetix (Matthews et al.), a heterogeneous space of procedurally generated physics-based control tasks that supports billions of environment interactions and provides held-out human-designed tasks for evaluating zero-shot generalization. Its **Small (S)**, **Medium (M)**, and **Large (L)** configurations vary the permitted numbers of shapes, joints, and thrusters, providing an interpretable axis of structural complexity. This combination enables us to study optimization over extended training, its relationship with task complexity, and whether stabilization transfers beyond the training distribution.

Optimization and representation degradation. Fig. 3 relates performance to the learner's optimization, value predictions, and representations. Performance decreases with environment complexity. Meanwhile, parameter magnitudes grow and the relative applied update, $\|\Delta w_k\|_2 / \|w_k\|_2$, declines, meaning that each update changes a progressively smaller fraction of the current parameter scale. Because this ratio also depends on the learning-rate schedule and optimizer state, we treat it as a comparative diagnostic rather than a direct measure of plasticity. Gradient norms increase in every configuration but are largest for Small environments, showing that gradient magnitude alone does not explain the performance ordering.

The critic's output distribution also shifts with complexity. Value predictions become more negative and less dispersed, but these moments depend on the corresponding target distribution and are not direct measures of prediction quality. More importantly, the rank correlation between value predictions and empirical returns deteriorates most strongly in Large environments, indicating weaker

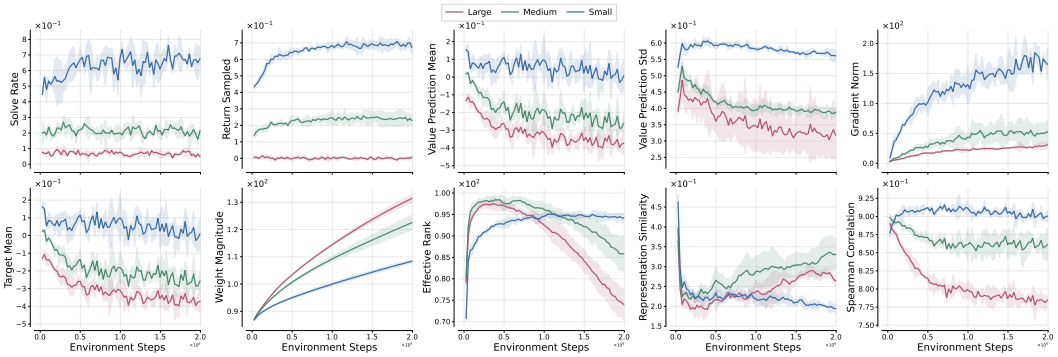

Figure 3: **Optimization degradation becomes more pronounced with environment complexity.** PLR exhibits weaker performance, greater parameter growth, diminishing relative updates, degraded value ordering, and loss of representational diversity in more complex Kinetix configurations.

value ordering. Representation quality follows a similar pattern. Effective rank declines most sharply in Large environments, while late-training cosine similarity increases, indicating more aligned and less diverse features. Poor performance is therefore associated with greater parameter scale, smaller relative updates, weaker value ordering, and reduced representational diversity. These diagnostics establish a coherent pattern of degradation, but do not identify a single cause. Definitions, measured network components, probe distributions, and aggregation procedures are provided in App. E.

Curriculum dynamics under PLR. To compare PLR's replay priorities with the student's current ability to learn from the selected levels, we report two student-dependent signals in Fig. 4. Following Rutherford et al. (2024), we define the learnability of level ℓ as $\mathcal{L}_{\text{learn}}(\ell) = p_\ell(1 - p_\ell)$, where p_ℓ is the current policy's estimated probability of solving that level. Learnability is maximal at $p_\ell = 0.5$ and vanishes when a level is never or always solved, making it a proxy for whether the level lies near the student's learning frontier rather than a direct measure of future improvement. Positive value loss (PVL), by contrast, aggregates positive value residuals along a trajectory and is used by PLR to prioritize levels for replay (Jiang et al., 2021b;a).

Mean learnability rises in all configurations but remains highest for Small environments. PVL follows a different ordering, remaining elevated for Medium and especially Large environments. High replay priority (i.e. high PVL) therefore does not necessarily imply that the selected levels are currently learnable. This mismatch is clearest in the Large configuration, where PLR continues to assign high priority despite low learnability and solve rate.

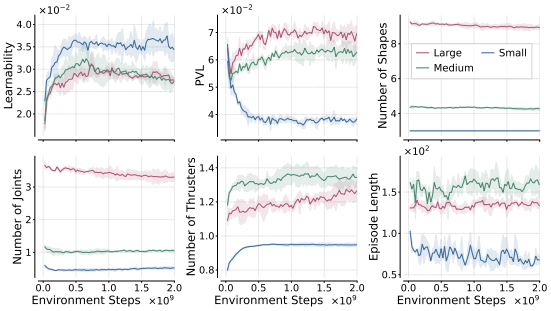

Figure 4: **Curriculum and environment dynamics under PLR.** Results for **Small**, **Medium**, and **Large** Kinetix configurations. Shaded regions denote 95% bootstrap confidence intervals across seeds.

This separation between priority and learnability resembles the *regret stagnation* phenomenon identified by Beukman et al. (2026). In that setting, minimax-regret curricula continue sampling levels with high irreducible regret after the student can no longer improve on them. We do not establish irreducible regret or minimax-regret convergence in Kinetix. Instead, we identify a related empirical mismatch and investigate a complementary learner-side explanation: optimization degradation may reduce the student's ability to benefit from levels that remain highly prioritized.

Takeaway. At greater complexity, PLR maintains high replay priority despite lower measured learnability and broader optimization degradation. These associations do not establish causality, motivating controlled interventions under the same curriculum rule.

4 Testing the Actionability of Optimization Pathologies

The preceding analysis shows that parameter growth, weaker value–outcome correspondence, and deteriorating representations accompany poor performance. Because these patterns may contribute to degradation or merely reflect it, we test whether directly targeting them improves learning. We therefore evaluate controlled learner-side interventions while holding the curriculum rule, actor objective, architecture, interaction budget, and unrelated training choices fixed.

4.1 Controlled Optimization Interventions

We screen established interventions designed to control parameter scale, reduce sensitivity to non-stationary value targets, or preserve representation quality (Juliani & Ash, 2024; Moalla et al., 2024; Mayor et al., 2025; Castanyer et al., 2025). Our goal is not to identify an exhaustive or globally optimal stabilization recipe, but to test whether specific pathologies identified in Sec. 3 can be mitigated. Definitions, hyperparameters, and selection procedures are provided in App. D.

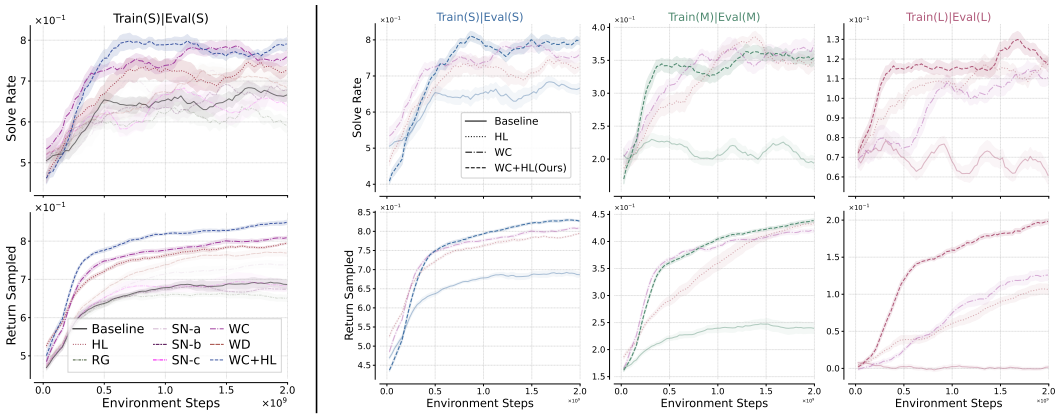

Figure 5: **Intervention screen and factorial ablation under PLR. Left:** Evaluation solve rate and sampled return on Small for the complete intervention suite. See Table 6 for acronym definitions and App. E for a detailed explanation. **Right:** Baseline, weight clipping (WC), HL-Gauss (HL), and WC+HL across Small, Medium, and Large. Several interventions improve over baseline, but WC+HL provides the most consistent gains, including a substantially larger return improvement in Large.

Fig. 5 shows that not all regularization interventions improve performance consistently. Weight clipping yields the strongest performance among the parameter-control interventions on Small, while weight decay and HL-Gauss also improve over baseline. Other interventions produce smaller or inconsistent gains. We select WC and HL not because they exhaust all promising combinations, but because each performs strongly and targets a different aspect of the learner degradation identified in Sec. 3. Weight clipping directly constrains critic parameter magnitudes, addressing the parameter growth observed during training, while HL-Gauss replaces scalar value regression with categorical prediction over a discretized value support, reducing sensitivity to large scalar target errors and targeting instability in value learning. In combination, these interventions target complementary aspects of learner degradation and are designed to preserve trainability (Elsayed et al., 2024; Farebrother et al., 2024). Mathematical definitions and implementation details are provided in App. C.

The factorial comparison clarifies this complementarity. In Small and Medium, WC and HL each improve performance, while their combination generally learns faster or reaches a stronger final result. The distinction is clearest in Large, where WC+HL achieves substantially higher sampled return than either component alone. Exhaustively evaluating every intervention combination is infeasible at Kinetix scale, so we do not claim global optimality. Instead, WC+HL is the strongest mechanistically motivated combination supported by our screen and is subsequently evaluated beyond PLR.

4.2 Effects on Optimization and Curriculum Dynamics

We denote WC+HL as the *Stabilized* learner and compare it with baseline PLR across **Small (S)**, **Medium (M)**, and **Large (L)** environments. We first examine whether stabilization mitigates the

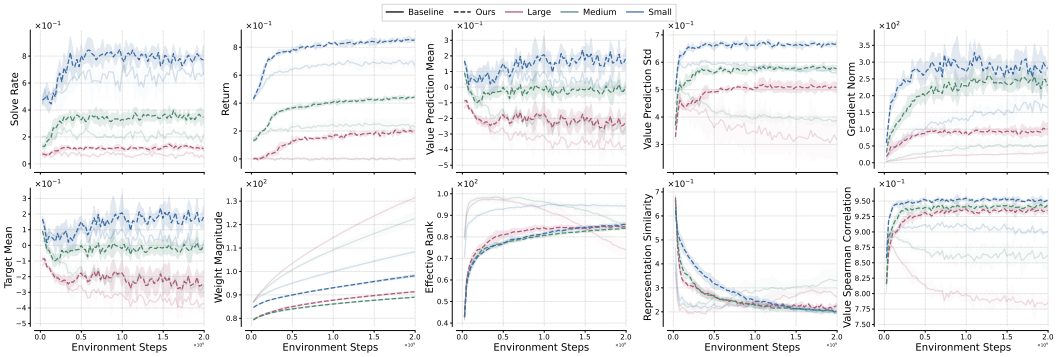

Figure 6: **Training stabilization improves performance and mitigates learner degradation.** Across environment complexities, WC+HL controls parameter growth while maintaining strong learning signals, preserves value–return ordering, and mitigates late representation degradation in the more challenging settings. These changes accompany consistently higher solve rates and returns.

learner-side degradation identified in Sec. 3. We then analyze how these changes affect the curriculum realized by the unchanged PLR rule. As shown in Fig. 6, weight clipping controls parameter growth across all complexity levels. The Stabilized learner also maintains substantial gradient norms while achieving higher solve rates and returns. This joint pattern indicates that stabilization does not operate by suppressing the learning signal. Instead, it allows the learner to remain responsive to experience while preventing parameter scale from growing unconstrained.

Value-prediction dynamics. Fig. 6 further shows that WC+HL does not stabilize the critic by simply shrinking its outputs. Prediction and target means remain broadly aligned, while prediction dispersion is better preserved than in the baseline. More importantly, stabilization maintains stronger correspondence between value predictions and empirical returns. By contrast, the baseline shows contracting prediction dispersion alongside deteriorating value–return correlation in the more challenging configurations. Critic quality is therefore better characterized by whether predicted values remain aligned with observed outcomes than by low prediction variance alone. WC+HL thus preserves a more informative value signal across complexity levels.

Representation dynamics. As shown in Fig. 6, stabilization also preserves representation structure in the regimes where the baseline degrades most strongly. WC+HL mitigates the late-training decline in effective rank and the increase in feature similarity observed in the more challenging configurations. The Small results show that higher effective rank alone is not sufficient for improved performance. Rather, stabilization preserves feature diversity in settings where baseline learning stagnates. These representation changes accompany, but do not by themselves explain, the improvements in learning.

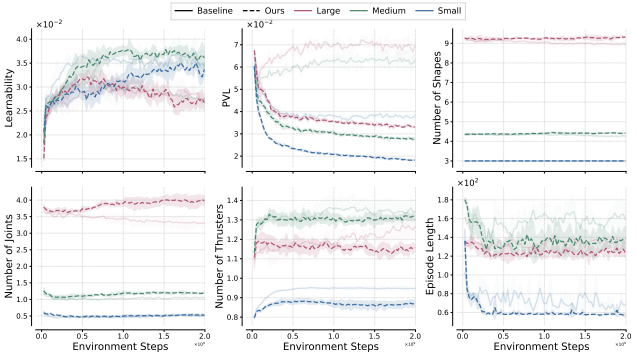

Figure 7: **Training stabilization changes the curriculum realized by PLR.** Although the replay rule remains fixed, WC+HL changes student-dependent curriculum signals and shortens episodes while the measured structural properties of replayed levels remain broadly comparable.

Effects on the realized curriculum. Because PLR computes replay priorities from student-dependent statistics, improving the learner can change the training distribution realized by an otherwise unchanged curriculum rule. As shown in Fig. 7, WC+HL produces lower PVL scores across complexity levels, indicating that the same PLR rule receives different value-based priority signals from a more trainable student. Learnability changes non-uniformly across complexity levels, remaining broadly comparable in Small, increasing in Medium, and decreasing late in Large. Thus, stabilization does not simply shift the curriculum toward uniformly more learnable levels.

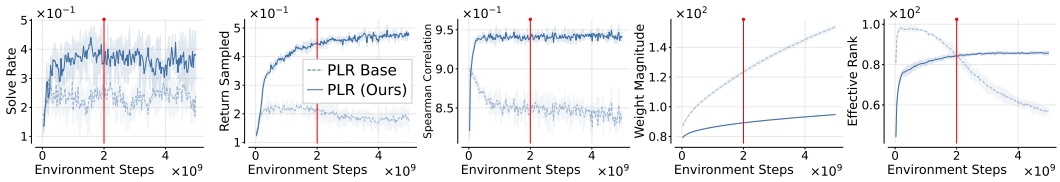

Figure 9: **Stabilization extends the useful training horizon of PLR.** Beyond the original two-billion-step budget marked in orange, baseline return and learner diagnostics deteriorate. WC+HL maintains solve rate, continues improving return, and preserves value ordering and effective rank through five billion interactions.

The realized environments also retain broadly comparable measured structural complexity: shape counts remain similar, joint counts are comparable or higher under stabilization, and thruster counts change only moderately. Episode lengths become shorter, particularly in Small and Medium. Taken together with the performance improvements established above, these results suggest that stabilization improves how the student uses structurally comparable curricula rather than relying on uniformly simpler levels.

Takeaway. Training-stability interventions improve learner trainability across environment complexities. Under adaptive UED, these learner-side improvements also reshape the curriculum realized by an unchanged curriculum rule.

5 Scaling across Training Regimes

The preceding experiments show that controlling parameter scale and value-loss sensitivity improves PLR training. We now test whether these benefits persist across three dimensions: the mechanism constructing the training distribution, the scale of training, and the evaluation distribution. Unless stated otherwise, *Stabilized* denotes WC+HL, using the configuration selected in Sec. 4. Within each comparison, the baseline and Stabilized learners use the same curriculum rule, architecture, and environment-interaction budget.

Across training-distribution mechanisms. We extend the evaluation beyond PLR to the domain-randomization (DR) baseline (Jakobi, 1997; Tobin et al., 2017), SFL (Rutherford et al., 2024), and ACCEL (Parker-Holder et al., 2022). These approaches construct their training distributions differently: DR samples directly from the generator, SFL selects levels according to estimated learnability, and ACCEL expands the task distribution through environment editing. As shown in Fig. 8, stabilization yields consistent gains across these mechanisms and Kinetix complexity levels. The effect is therefore not specific to PLR's replay rule, although its magnitude varies across curriculum mechanisms.

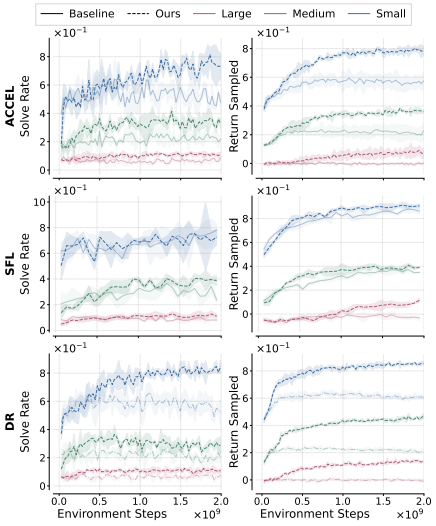

Figure 8: **Stabilization improves performance across training-distribution mechanisms.** WC+HL improves solve rate and return for DR, SFL, and ACCEL across Kinetix complexity levels.

Scaling the training horizon. Additional interaction is useful only while the learner can convert experience into improved behavior. We therefore extend PLR training on Small environments from two to five billion interactions. As shown in Fig. 9, the baseline obtains little benefit beyond the original budget: return declines, parameter magnitude continues to grow, value–outcome correspondence remains weak, and effective rank deteriorates. The Stabilized learner instead maintains solve rate, continues to gain return, and preserves value ordering and representation rank. The performance gap therefore widens beyond two billion interactions, showing that conclusions about a UED method can depend on the training horizon at which it is evaluated.

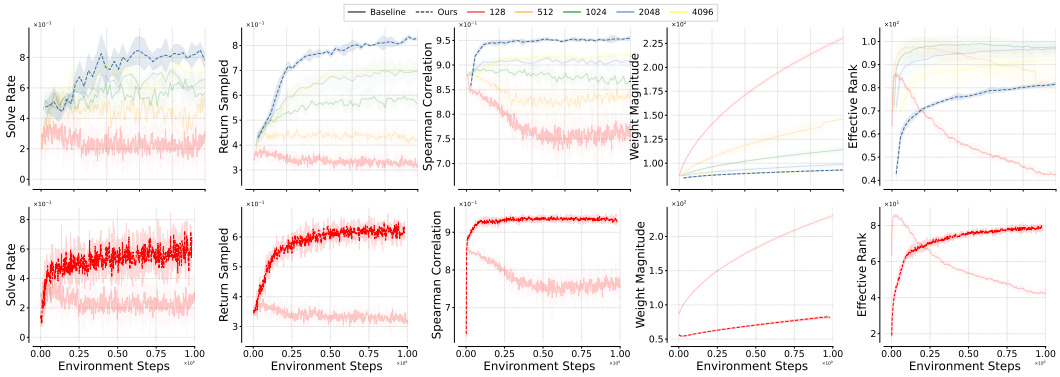

Figure 10: **Stabilization improves PLR across parallelism regimes. Top:** Increasing parallelism improves performance and learner diagnostics up to the default 2048-environment setting, with little additional gain at 4096 environments. At the default scale, WC+HL achieves stronger performance together with healthier training dynamics than the corresponding baseline. **Bottom:** Under limited parallelism, with only 128 environments, WC+HL still improves performance, preserves value ordering and effective rank, and controls parameter growth.

Stabilization across parallelism regimes. Recent work shows that increasing PPO rollout batch size can prevent a particular form of learning stagnation in open-ended Kinetix training (Beukman et al., 2026). We study how this scaling strategy interacts with the learner-side degradation considered here by varying the number of parallel environments used by PLR on Small while holding total interaction fixed. As shown in Fig. 10, increasing parallelism yields better performance, value ordering, and effective rank up to the default 2048-environment setting, with little additional gain at 4096 environments. At the default scale, WC+HL further strengthens performance while maintaining healthier training dynamics than the corresponding baseline. Moreover, with only 128 environments, stabilization substantially raises solve rate and return while controlling parameter growth and representation degradation. These results suggest that rollout scaling and learner stabilization are complementary. Increasing parallelism can enhance PPO updates, but its benefits eventually saturate, while stabilization remains effective both at the default scale and when large-scale parallelism is unavailable.

Cross-complexity generalization. We next test whether stabilization improves only the training regime or also transfers across task complexity. Following the Kinetix protocol (Matthews et al.), agents trained on Small, Medium, or Large environments are evaluated without adaptation on all three complexity levels. Fig. 11 reports episodic return on generated evaluation levels, while Fig. 21 reports solve rate on human-designed holdout tasks. Stabilization improves both matched-complexity performance and transfer across complexity levels. The largest gains occur for agents trained on Medium and Large, where the corresponding baselines exhibit the strongest optimization and representation degradation. The consistent improvements in solve rate and sampled return show that stabilization benefits both successful task completion and graded progress.

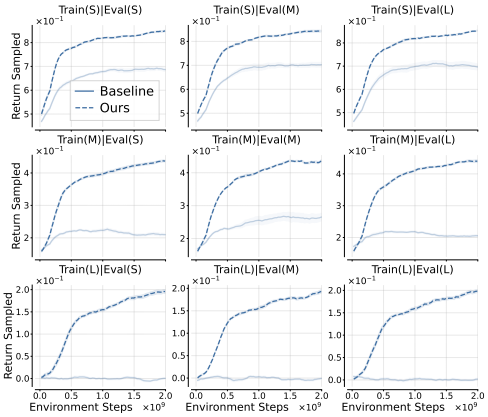

Figure 11: **Generalization to randomly generated tasks.** Zero-shot sampled return across training and evaluation complexities.

Robustness to sequential distribution shifts. Finally, we test robustness when the training distribution changes over time. Following the Kinetix lifelong-learning protocol (Matthews et al.), PLR is trained sequentially on **Small**, **Medium**, and **Small** environments. The optimizer state and PLR buffer are retained across transitions for both learners, and performance is continuously measured on the Medium human-designed holdout set.

As shown in Fig. 12, the Stabilized learner transfers more strongly to Medium tasks during the initial Small stage and maintains its advantage after training switches to Medium. When training returns to Small, the baseline loses much of the performance acquired during the Medium stage, whereas the Stabilized learner preserves more of this gain. Because both learners use the same replay and state-retention protocol, this difference cannot be attributed to additional rehearsal or an explicit continual-learning objective. The result therefore supports improved retention on the Medium holdout set under sequential distribution shift.

This pattern is consistent with reduced forgetting, interference, or loss of plasticity, but the Medium-only evaluation does not distinguish among these mechanisms or fully characterize the adaptation–retention trade-off. We therefore interpret the experiment as evidence of improved retention rather than a complete continual-learning characterization.

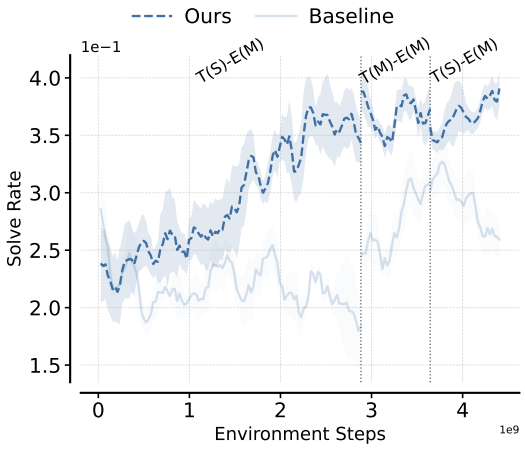

Figure 12: **Stabilization improves retention under sequential distribution shifts.** Under Small → Medium → Small training, Ours transfers more strongly to the Medium holdout set and better preserves that performance after training returns to Small. Dotted lines mark distribution changes.

Takeaway. Learner stabilization generalizes beyond PLR, remains effective over longer training horizons and across parallelism regimes, and improves transfer and retention across distributions. Learner trainability therefore shapes how effectively UED systems convert additional scale and task diversity into sustained performance.

6 Discussion and Conclusion

A better curriculum cannot indefinitely compensate for a learner that is losing its ability to learn. Our detailed analysis of PLR shows that extended training can be accompanied by parameter growth, degraded value prediction, and deteriorating representations. Controlled interventions targeting these pathologies improve learning, with benefits extending across SFL, ACCEL, and domain randomization. Weight clipping controls parameter scale, while HL-Gauss reduces the sensitivity of value learning to scalar target errors. Their combination yields stronger performance and more stable training dynamics without changing the curriculum rule or interaction budget.

These results suggest that UED should be viewed not only as a problem of constructing better curricula, but as a coupled system in which the learner also determines how useful a curriculum can be. Because adaptive curricula depend on student-generated signals, learner degradation can affect both how experience is used and which experience is encountered next. Conversely, stabilizing the learner changes the training distribution realized by an otherwise unchanged curriculum rule. Apparent limitations of curriculum design may therefore partly reflect degradation in the learner used to evaluate it. Training dynamics should thus be considered part of curriculum design, rather than treating the learner update as a fixed component.

This perspective also has implications for evaluation. Performance at a fixed training horizon may hide whether a learner remains capable of benefiting from additional experience. Our long-horizon and parallelism results show that more interaction or rollout scale does not automatically translate into continued learning. Monitoring learner-side dynamics alongside return can therefore help distinguish limits of the curriculum from limits of the learner. Scaling the curriculum and preserving learner trainability should not be treated as independent problems.

Limitations. Our evidence is limited to Kinetix and PPO, and hyperparameter sensitivity, probe choices, and intervention interactions constrain causal attribution. We do not claim that weight clipping and HL-Gauss are uniquely effective. Continual learning is a particularly challenging extension for UED, where curriculum-induced non-stationarity compounds explicit task shifts. Broader validation and adaptive stabilization remain important directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools for aiding and polishing paragraph-level writing, discussing and refining the structure and flow of the introduction, finding related work, and editing and writing the code. We also used generative AI tools to assist with early drafts of some sections, which were later substantially rewritten and replaced by the authors. We have not used generative AI tools for research ideation, and the rest of the required disclosure tasks, such as generating synthetic datasets or proving mathematical claims, are not applicable to this work. We have reviewed the AI-assisted work. We reviewed LLM-generated code and read and incorporated suggested related work papers manually. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Michael Beukman, Khimya Khetarpal, Zeyu Zheng, Will Dabney, Jakob Foerster, Michael Dennis, and Clare Lyle. Preventing learning stagnation in ppo by scaling to 1 million parallel environments, 2026. URL https://arxiv.org/abs/2603.06009.

Roger Creus Castanyer, Johan Obando-Ceron, Lu Li, Pierre-Luc Bacon, Glen Berseth, Aaron Courville, and Pablo Samuel Castro. Stable gradients for stable learning at scale in deep reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Vqj65VeDOu.

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

Mohamed Elsayed, Qingfeng Lan, Clare Lyle, and A. Rupam Mahmood. Weight clipping for deep continual and reinforcement learning. In *Reinforcement Learning Conference*, 2024. URL https://openreview.net/forum?id=9Vagp8vnPa.

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taiga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, et al. Stop regressing: Training value functions via classification for scalable deep rl. In *International Conference on Machine Learning*, pp. 13049–13071. PMLR, 2024.

Thomas Foster, Anya Sims, Johannes Forkel, and Jakob Foerster. Lilo: Learning to reason at the frontier of learnability. *Advances in Neural Information Processing Systems*, 38:48941–48974, 2026.

Florin Gogianu, Tudor Berariu, Mihaela C Rosca, Claudia Clopath, Lucian Busoniu, and Razvan Pascanu. Spectral normalisation for deep reinforcement learning: an optimisation perspective. In *International Conference on Machine Learning*, pp. 3734–3744. PMLR, 2021.

Nick Jakobi. Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adaptive behavior*, 6(2):325–368, 1997.

Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. *Advances in Neural Information Processing Systems*, 34:1884–1897, 2021a.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pp. 4940–4950. PMLR, 2021b.

Arthur Juliani and Jordan T Ash. A study of plasticity loss in on-policy deep reinforcement learning. *Advances in Neural Information Processing Systems*, 37:113884–113910, 2024.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=O9bnihsFfXU.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization. *arXiv preprint arXiv:2308.11958*, 2023.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. *arXiv preprint arXiv:2204.09560*, 2022.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Michael Matthews, Michael Beukman, Chris Lu, and Jakob Nicolaus Foerster. Kinetix: Investigating the training of general agents through open-ended physics-based control tasks. In *The Thirteenth International Conference on Learning Representations*.

Walter Mayor, Johan Obando-Ceron, Aaron Courville, and Pablo Samuel Castro. The impact of on-policy parallelized data collection on deep reinforcement learning networks. In *International Conference on Machine Learning*, pp. 43331–43352. PMLR, 2025.

Ishita Mediratta, Minqi Jiang, Jack Parker-Holder, Michael Dennis, Eugene Vinitsky, and Tim Rocktäschel. Stabilizing unsupervised environment design with a learned adversary. In Sarath Chandar, Razvan Pascanu, Hanie Sedghi, and Doina Precup (eds.), *Proceedings of The 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pp. 270–291. PMLR, 22–25 Aug 2023. URL https://proceedings.mlr.press/v232/mediratta23a.html.

Skander Moalla, Andrea Miele, Daniil Pyatko, Razvan Pascanu, and Caglar Gulcehre. No representation, no trust: Connecting representation, collapse, and trust issues in PPO. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Wy9UgrMwD0.

Nathan Monette, Alistair Letcher, Michael Beukman, Matthew Thomas Jackson, Alexander Rutherford, Alexander David Goldie, and Jakob Nicolaus Foerster. An optimisation framework for unsupervised environment design. In *Reinforcement Learning Conference*, 2025. URL https://openreview.net/forum?id=WnknYUybWX.

Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzciński, Mateusz Ostaszewski, and Marek Cygan. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International conference on machine learning*, pp. 16828–16847. PMLR, 2022.

Johan Obando-Ceron, Walter Mayor, Samuel Lavoie, Scott Fujimoto, Aaron Courville, and Pablo Samuel Castro. Simplicial embeddings improve sample efficiency in actor-critic agents. *arXiv preprint arXiv:2510.13704*, 2025.

Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17473–17498. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/parker-holder22a.html.

Haoyuan Qin, Chennan Ma, Mian Deng, Zhengzhu Liu, Songzhu Mei, Xinwang Liu, Cheng Wang, and Siqi Shen. The dormant neuron phenomenon in multi-agent reinforcement learning value factorization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=4NGrHrhJPx.

Alex Rutherford, Michael Beukman, Timon Willi, Bruno Lacerda, Nick Hawes, and Jakob Foerster. No regrets: Investigating and improving regret approximations for curriculum discovery. *Advances in Neural Information Processing Systems*, 37:16071–16101, 2024.

Marius Saether, Amir Kolic, Tomaso Poggio, and Pierfrancesco Beneventano. Does weight decay enhance training stability? *arXiv preprint arXiv:2605.16622*, 2026.

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

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. The MIT Press, Cambridge, Massachusetts, 2nd edition, 2018.

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 *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. Harnessing structures for value-based planning and reinforcement learning. *arXiv preprint arXiv:1909.12255*, 2019.

APPENDIX CONTENTS

A Related Work **14**

B Learner Trainability as a Design Principle for UED **14**

C Selected interventions. **15**

D Interventions **15**

D.1 Weight Clipping 15

D.2 Weight Decay 15

D.3 Spectral Norm 16

D.4 Regen 16

D.5 HL-Gauss 16

E Metrics **19**

E.1 Solve Rate 19

E.2 Return Sampled 19

E.3 PVL 19

E.4 MaxMac 19

E.5 Value Prediction Mean 19

E.6 Gradient Norm 19

E.7 Representation Similarity 20

E.8 Spearman Correlation 20

E.9 Targen Mean 20

E.10 Effective Rank 20

E.11 Weight Magnitude 20

E.12 Number of Joints 21

E.13 Number of Shapes 21

E.14 Number of Thrusters 21

E.15 Episode Length 21

E.16 Learnability 21

F Interventions Table **22**

G Deep reinforcement learning **24**

H Unsupervised environment design **25**

I Experiments **26**

A Related Work

UED methods construct training distributions by generating, selecting, or replaying environments using signals derived from the student. PAIRED frames environment generation as an adversarial teacher–student game (Dennis et al., 2020), PLR and replay-guided environment design prioritize previously encountered levels using estimates of regret or learning potential (Jiang et al., 2021b;a), ACCEL expands the curriculum through environment editing (Parker-Holder et al., 2022), and SFL selects environments estimated to be learnable by the current student (Rutherford et al., 2024). Kinetix evaluates these mechanisms in a large and heterogeneous space of physics-based tasks, demonstrating the importance of curriculum choice while also revealing substantial degradation with increasing environment complexity (Matthews et al.). Related work has improved the stability of particular teacher–student algorithms (Mediratta et al., 2023) or studied UED objectives and convergence from an optimization perspective (Monette et al., 2025). These works primarily address how curricula are constructed or optimized. Separately, deep RL research has documented parameter growth, feature-rank deterioration, dormant neurons, representation collapse, and loss of plasticity during prolonged training (Kumar et al., 2021; Nikishin et al., 2022; Qin et al., 2024; Lyle et al., 2023). In PPO, representation degradation can undermine effective trust-region behavior (Moalla et al., 2024), while on-policy data collection choices can substantially alter network optimization and plasticity (Mayor et al., 2025). Existing interventions can mitigate particular failure modes, including parameter growth (Elsayed et al., 2024) and sensitivity to scalar value targets (Farebrother et al., 2024), but these phenomena have largely been studied outside UED. Our work connects these literatures by examining how the student's optimization dynamics evolve under curriculum–student coupling and whether addressing the resulting pathologies improves existing UED methods without redesigning their curricula.

B Learner Trainability as a Design Principle for UED

UED has largely been formulated as a problem of constructing effective training distributions. PAIRED generates environments through an adversarial teacher–student game (Dennis et al., 2020), PLR and replay-guided methods prioritize previously encountered levels using estimates of learning potential or regret (Jiang et al., 2021b;a), ACCEL expands the curriculum through environment editing (Parker-Holder et al., 2022), and SFL selects environments estimated to be learnable by the current student (Rutherford et al., 2024). These methods differ substantially in how they construct curricula, but they are commonly evaluated through the performance ultimately achieved by a shared class of student learners. Related work has improved the stability of particular teacher–student algorithms (Mediratta et al., 2023) and studied UED objectives and convergence from an optimization perspective (Monette et al., 2025). Our results highlight a complementary concern. Even when a curriculum identifies useful training environments, degradation within the student can prevent that experience from producing further capabilities.

This distinction is important because the observed performance of a UED method reflects two coupled processes. The curriculum determines which environments generate experience, while the student determines how effectively that experience is converted into learning. Deep RL research outside UED has shown that prolonged optimization can produce parameter growth, feature-rank deterioration, dormant neurons, representation collapse, and loss of plasticity (Kumar et al., 2021; Nikishin et al., 2022; Qin et al., 2024; Lyle et al., 2023). These failures also arise in PPO, where representation degradation can weaken effective trust-region behavior (Moalla et al., 2024), and can depend strongly on how on-policy experience is collected and reused (Mayor et al., 2025). Under UED, such degradation has a further consequence. Changes in the student can alter the environments subsequently generated, selected, or replayed, allowing learner optimization to influence its own future training distribution. Apparent limitations of a curriculum mechanism may therefore partly reflect limitations of the student through which it is evaluated.

Learner trainability should consequently be treated as a design and evaluation principle for UED. Comparisons between curriculum mechanisms should report not only final return and properties of the generated environments, but also whether their students remain capable of learning as task complexity and interaction budgets increase. This motivates tracking parameter and update statistics, value predictions, neuronal activity, and representation quality alongside performance. Evaluations should also distinguish the curriculum rule from the curriculum it ultimately realizes, since two

students trained under the same adaptive rule can induce different environment distributions as their behaviors diverge. Developing stronger curriculum mechanisms remains essential, but their benefits may be obscured when fundamental optimization failures dominate training. Revisiting existing UED methods with more stable learners may therefore reveal capabilities that were previously hidden, while future methods could account not only for which environments appear informative, but also for whether the student remains sufficiently trainable to benefit from them.

C SELECTED INTERVENTIONS.

Weight clipping constrains the critic parameters ϕ after every optimizer update (Elsayed et al., 2024). Given unconstrained parameters $\widetilde{\phi}_{k+1}$, we apply $\phi_{k+1} = \Pi_{\mathcal{C}_c}(\widetilde{\phi}_{k+1})$, where $\mathcal{C}_c = \{\phi : \|\phi\|_\infty \leq c\}$. This directly controls parameter scale but does not bound gradients or function changes. HL-Gauss replaces scalar squared-error regression with a histogram cross-entropy loss (Farebrother et al., 2024). The critic predicts probabilities $p_\phi(\cdot \mid s) = \mathrm{softmax}(h_\phi(s))$ over a fixed set of ordered bins. A scalar target y is converted into a categorical label $m^{\mathrm{HL}}(y)$ by integrating a Gaussian centered at y over those bins, and the scalar value prediction is recovered as the expectation of the predicted histogram. Unlike distributional RL, HL-Gauss represents a scalar expected-value target rather than the full return distribution. Its logit gradient satisfies $\partial \mathcal{L}_{\mathrm{HL}} / \partial h_j = p_\phi(z_j \mid s) - m_j^{\mathrm{HL}}(y) \in [-1, 1]$, removing the direct linear dependence of the prediction-head gradient on the raw scalar error. The full parameter gradient remains dependent on the network Jacobian. Parameter scope, histogram support, smoothing scale, and sensitivity analyses are provided in Apps. D.1 and D.5.

D INTERVENTIONS

D.1 WEIGHT CLIPPING

We implemented the weight clipping (WC) intervention by varying the scaling factor of the clipping range, using values of 1, 2, 4, and 8. We compared each configuration against the PLR baseline to evaluate the effect of the clipping range on performance Elsayed et al. (2024).

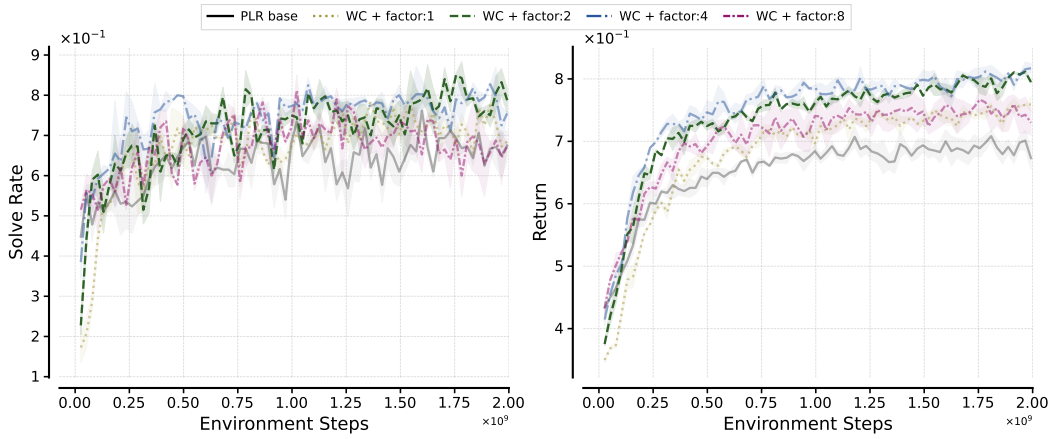

Figure 13: **Solve rate (left) and sampled return (right) over 5 seeds.** We compare the WC intervention on PLR using different scaling factors for the clipping range. Scaling factors below 2 result in lower performance, while factors of 2 and 4 achieve similar performance. Increasing the scaling factor to 8 leads to a substantial performance degradation.

D.2 WEIGHT DECAY

Based on the results reported in Castanyer et al. (2025), we set the weight decay coefficient to 1×10^{-2}. We applied Weight Decay uniformly across the network to regularize its parameters and promote training stability Saether et al. (2026).

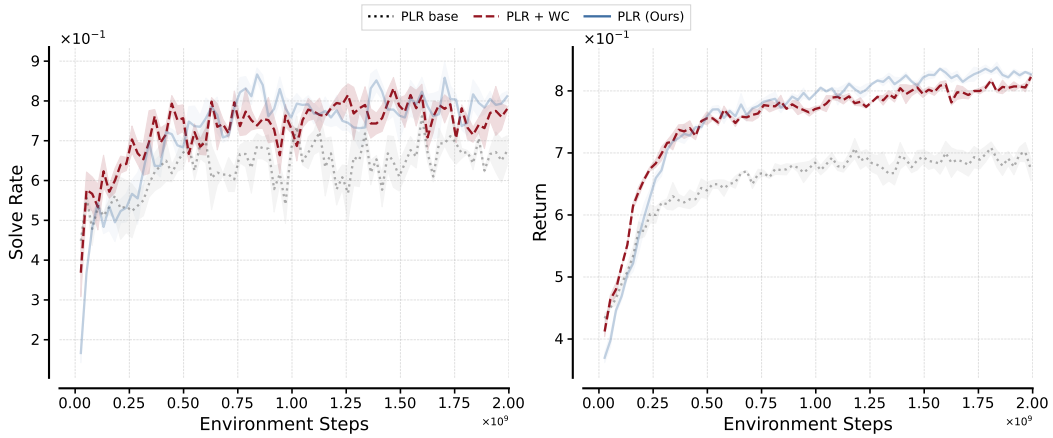

Figure 14: **Solve rate (left) and sampled return (right) over 5 seeds.** We compare the PLR baseline, the best WC configuration using a scaling factor of 2, and our selected interventions. Our interventions show a modest performance improvement over WC, while both substantially outperform the PLR baseline.

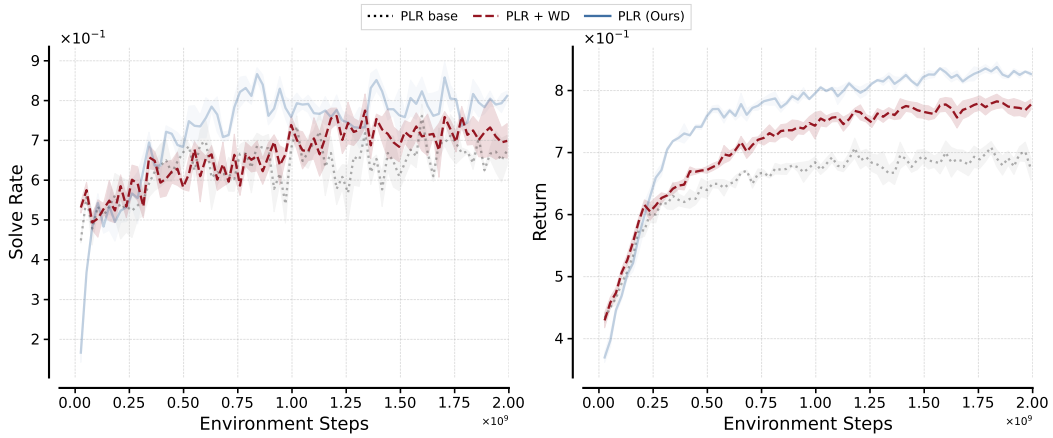

Figure 15: **Solve rate (left) and sampled return (right) over 5 seeds.** We compare the PLR baseline, the WD intervention, and our selected intervention. WD outperforms the PLR baseline, while our intervention achieves higher performance than both WD and the baseline.

D.3 Spectral Norm

We implemented Spectral Normalization to constrain layer norms and stabilize training, evaluating its application to the actor, critic, and both networks. We then compared the best-performing configuration against the PLR baseline and our proposed intervention, allowing us to directly assess their behavior and performance Gogianu et al. (2021).

D.4 ReGen

Following the results reported in Obando-Ceron et al. (2025), we conducted our experiments using ReGen with $\tau = 10^{-5}$. Specifically, we applied ReGen by zeroing weights whose magnitude falls below the threshold, thereby inducing sparsity in the network, following the procedure described by Kumar et al. (2023).

D.5 HL-Gauss

We use HL-Gauss to represent value predictions as a categorical distribution and train the critic using a cross-entropy loss. We evaluate this intervention with different numbers of bins: 21, 51, 101, and

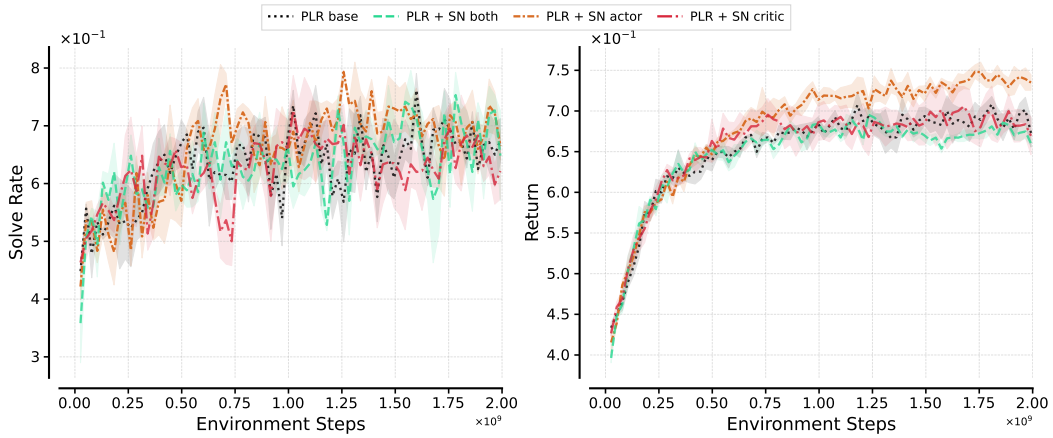

Figure 16: **Solve rate (left) and sampled return (right) over 5 seeds.** We evaluate Spectral Normalization (SN) on PLR by applying it to the actor, the critic, or both networks. Applying SN to the critic or both networks yields similar results, while applying it to the actor achieves higher performance than either configuration.

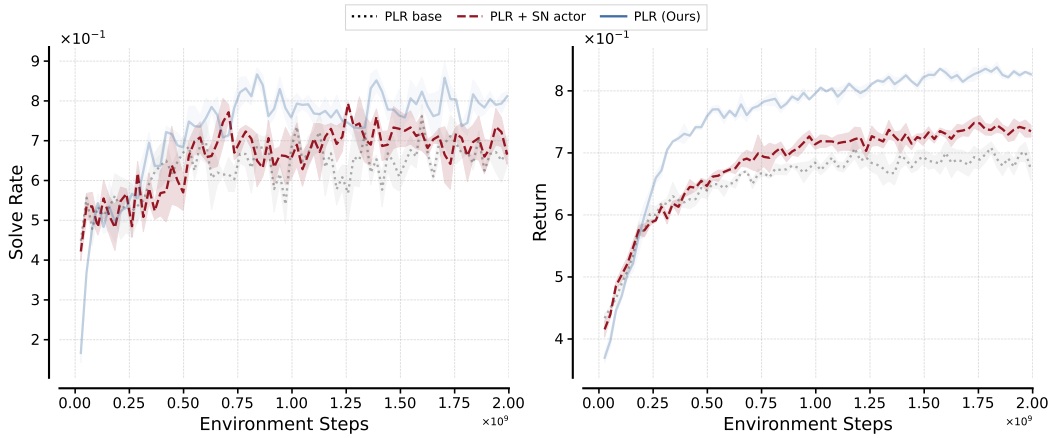

Figure 17: **Solve rate (left) and sampled return (right) over 5 seeds.** We contrast the PLR baseline, the best SN configuration applied to the actor, and our selected intervention. SN on the actor outperforms the PLR baseline, while our intervention yields higher performance than both SN and the baseline.

201. This hyperparameter controls the resolution of the categorical value representation, and we compare each configuration against the PLR baseline Farebrother et al. (2024).

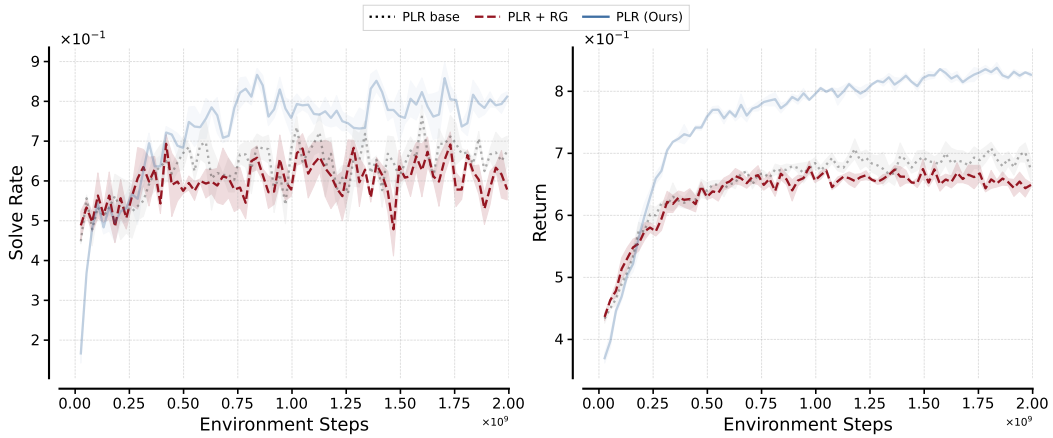

Figure 18: **Solve rate (left) and sampled return (right) over 5 seeds.** We compare the PLR baseline, ReGen, and the WC + CL (Ours) intervention. Ours intervention consistently outperforms both the baseline and ReGen, achieving faster learning and higher values across training, while ReGen performs similarly to or below the baseline.

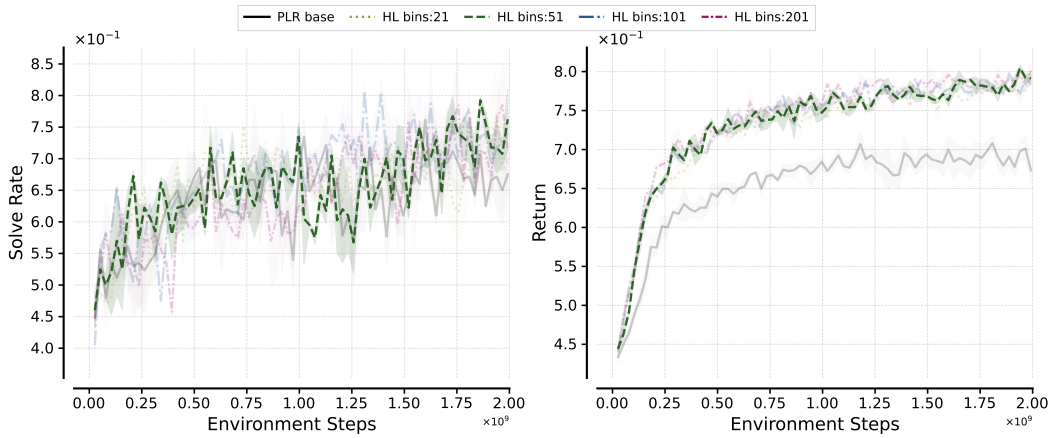

Figure 19: **Solve rate (left) and sampled return (right) over 5 seeds.** We compare the HL intervention on PLR using different numbers of bins: 21, 51, 101, and 201, which control the resolution of the categorical value representation. All tested configurations achieve similar results, consistent with the robustness to this hyperparameter reported in Farebrother et al. (2024).

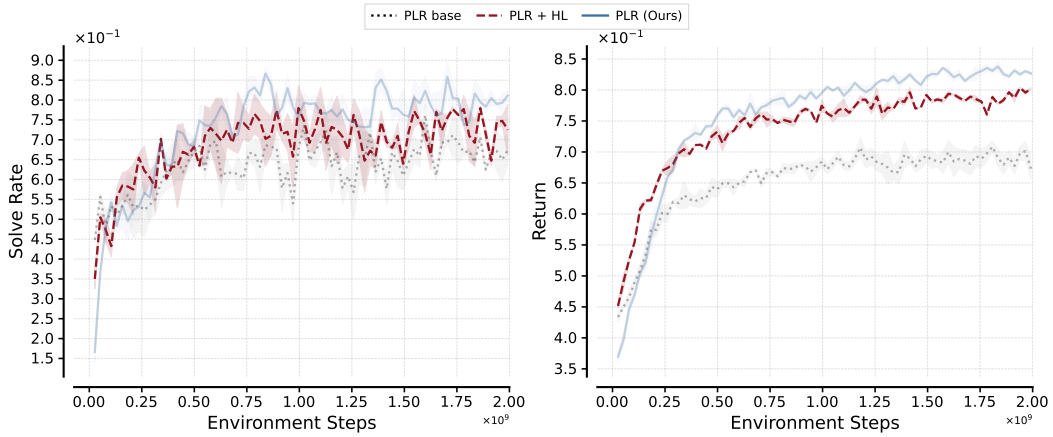

Figure 20: **Solve rate (left) and sampled return (right) over 5 seeds.** We consider the PLR baseline, HL with 51 bins, and our selected intervention. HL outperforms the PLR baseline, while our intervention performs better than both HL and the baseline.

E METRICS

E.1 SOLVE RATE

Solve Rate measures the proportion of evaluation episodes in which the agent successfully completes the task. We evaluate this metric on a static set of hand-designed physics-based holdout levels that are not used for training, providing a zero-shot evaluation of the agent's ability to generalize to unseen environments. A progressive increase in solve rate during training indicates that the agent is increasingly able to successfully solve these unseen tasks, reflecting improved generalization to physics-based challenges Matthews et al..

E.2 RETURN SAMPLED

Sampled Return measures the total return obtained by the agent over episodes evaluated on procedurally generated levels sampled from the Domain Randomization (DR) distribution. An increasing and sustained trend indicates that the agent is progressively obtaining higher returns across the sampled task distribution, suggesting improved performance across the diverse physics-based environments encountered during training Matthews et al..

E.3 PVL

Positive Value Loss (PVL) is a heuristic metric used in Unsupervised Environment Design (UED) algorithms, such as PLR and ACCEL, to approximate regret and prioritize levels for replay. By considering only positive value losses, PVL selectively captures positive advantage estimates, indicating that the observed returns exceed those predicted by the critic. While a high PVL makes an individual level a higher-priority candidate for replay, when tracking PVL across the replay buffer, we expect it to be initially high and gradually decrease over time. This decline indicates that the agent's value estimates are becoming increasingly accurate and that the learning opportunities associated with these levels are being progressively resolved Jiang et al. (2021a).

E.4 MAXMAC

Maximum Monte Carlo (MaxMC) is a heuristic scoring function used in UED algorithms such as PLR and ACCEL to approximate regret and prioritize levels for replay. Unlike PVL, which relies on the critic's value estimates, MaxMC uses the highest return previously achieved on a given level, denoted by $R_{\max}$. By replacing the value target with this maximum historical return, MaxMC reduces the dependence of the regret estimate on the agent's current policy and its value function. For an individual level, a high MaxMC score indicates a large gap between the best return previously achieved on that level and the critic's current value estimate, suggesting greater estimated learning potential. When tracking MaxMC for levels sampled through domain randomization (DR), a gradual increase indicates that the generated levels progressively exhibit higher maximum achieved returns, whereas abrupt changes may indicate sudden shifts in the properties or difficulty of the sampled levels Jiang et al. (2021a).

E.5 VALUE PREDICTION MEAN

Mean Value Prediction measures the arithmetic mean of the critic's value estimates $V(s)$ across the states visited in a rollout batch, reflecting the expected future return estimated by the critic for the states encountered by the agent. A gradual increase in this metric during training can indicate that the agent is increasingly visiting states associated with higher expected returns, consistent with improved task performance. However, this metric should be interpreted alongside performance measures such as solve rate, since higher value estimates alone do not necessarily imply successful task completion Schulman et al. (2017).

E.6 GRADIENT NORM

The gradient norm is the global L_2 norm of the gradient vector of the loss function with respect to all trainable parameters of the neural network. It indicates how strongly the optimizer needs to adjust the

model parameters to correct value prediction errors and improve the policy. A stable and bounded gradient norm suggests that gradients propagate appropriately through the network and that learning progresses smoothly and continuously. It is computed as

$$\text{grad_norm} = \|\nabla_\theta \mathcal{L}\|_2 = \sqrt{\sum_i \left(\frac{\partial \mathcal{L}}{\partial \theta_i}\right)^2},$$

where $\mathcal{L}$ denotes the loss function and θ_i represents the trainable parameters of the neural network.

E.7 Representation Similarity

Representation Similarity measures the average cosine similarity between all latent representations (state embeddings) generated by the neural network within a training batch. Values close to 1 indicate that the representations point in similar directions in the latent space, suggesting a potential collapse of the learned representations. In contrast, values close to 0 indicate that the representations of different states are largely dissimilar and tend to point in orthogonal directions in the latent space.

E.8 Spearman Correlation

Spearman correlation measures the ranking fidelity of the Critic network. It evaluates whether the Critic can correctly rank the states within a rollout from better to worse by comparing its value predictions $V(s)$ with the corresponding empirical returns or targets G_t. A value close to 1 indicates that the Critic preserves the same ordering of states as reflected by their observed returns, suggesting high ranking fidelity.

E.9 Targen Mean

Mean Value Target measures the arithmetic mean of the value targets computed from the trajectories collected in a rollout, using Generalized Advantage Estimation (GAE). It reflects the average return observed by the agent across the states encountered during the rollout. A gradual increase in this metric during training indicates that the agent is experiencing higher returns in the environments it visits, suggesting that it is progressively learning to achieve more rewarding outcomes Schulman et al. (2017).

E.10 Effective Rank

This metric quantifies the dimensionality of learned representations by determining the smallest subspace that preserves at least 99% of the total variance. A higher feature rank indicates greater diversity in the learned representations. Following the approximate rank used by Yang et al. (2019), the feature rank k is defined as the smallest number of singular values required to satisfy

$$\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{j=1}^{n} \sigma_j^2} \geq \tau, \tag{1}$$

where σ_i denotes the i-th singular value of the feature matrix, n is the total number of singular values, and τ is the variance threshold, set to 0.99 in our experiments.

E.11 Weight Magnitude

This metric quantifies the magnitude of the neural network parameters, providing an indication of the scale of the learned weights. Larger weight norms correspond to parameters with greater magnitudes, which can affect the stability and generalization of the model. Following the formulation used by Lyle et al. (2022), the weight norm is computed as

$$\|\theta\|_2 = \sqrt{\sum_i \theta_i^2}, \tag{2}$$

where θ_i denotes the weight parameters of a given layer.

E.12 Number of Joints

In Kinetix, joints connect rigid bodies by imposing constraints that prevent the connected bodies from separating at the point of attachment. Fixed joints maintain a fixed relative configuration, whereas revolute joints allow relative rotation and can be actuated by motors that apply torque. The environment complexity increases with the maximum number of joints: Small (S) levels allow up to 1 joint, Medium (M) levels up to 2, and Large (L) levels up to 6 Matthews et al.. These joints increase the degrees of freedom of the system and can form coupled kinematic chains, where forces and torques applied at one joint can influence the dynamics of the connected bodies.

E.13 Number of Shapes

In Kinetix, the number of shapes determines the number of geometric components used to construct rigid bodies within the 2D scene. Together with Jax2D, Kinetix uses polygons and circles to represent the geometry of its physics-based environments Matthews et al.. The maximum number of shapes increases with environment complexity: Small (S) allows up to 7 shapes, including at most 5 polygons and 2 circles; Medium (M) allows up to 9 shapes, with at most 6 polygons and 3 circles; and Large (L) allows up to 16 shapes, with at most 12 polygons and 4 circles.

E.14 Number of Thrusters

The number of thrusters measures the maximum number of propulsion mechanisms present in a level. Each thruster provides an external linear propulsion force in the direction it points, giving the agent direct control over the motion of the associated body. The maximum number of thrusters increases with difficulty: Small (S) allows up to 1 thruster, while Medium (M) and Large (L) allow up to 2. This provides a range of task configurations, from environments that rely primarily on direct propulsion to more mechanically structured environments involving articulated bodies and joints Matthews et al..

E.15 Episode Length

Episode Length measures the average number of steps per episode, determined by whether the episode terminates after success, failure, or reaching the maximum episode horizon. We report this metric on the static, hand-designed holdout set, which is not used for training, as well as on levels sampled from the stochastic distribution used for Domain Randomization (DR). We refer to the latter as *Episode Length (Sampled)* and report it separately for each environment size. Lower episode lengths indicate that episodes terminate in fewer steps, which, when associated with successful completion, reflects faster task completion Matthews et al..

E.16 Learnability

Learnability measures the immediate learning potential of a task or level given the agent's current capabilities. In PLR-based UED, this metric can be evaluated over either the curated replay buffer or the uncurated Domain Randomization (DR) stream Matthews et al.. Higher learnability indicates that the sampled tasks lie closer to the agent's active learning region. For a level with success rate p, learnability is defined as

$$\mathcal{L} = p(1-p), \tag{3}$$

where

$$p = \frac{N_{\text{success}}}{N_{\text{episodes}}}. \tag{4}$$

For multiple environments, the aggregate learnability is computed as

$$\mathcal{L} = \sum_{e=1}^{N_{\text{env}}} p_e(1-p_e). \tag{5}$$

The metric is maximized at $p = 0.5$ and becomes zero when $p = 0$ or $p = 1$, emphasizing levels that are neither too difficult nor too easy Foster et al. (2026).

F INTERVENTIONS TABLE

Table 1: Default PPO hyperparameter settings for Kinetix

Hyper-parameter	Value
Num Environments	2048
Num Minibatches	32
Num epochs	8
Learning Rate	5e-5
Discount Factor	0.99
Gae Lambda	0.9
Vf Coef	0.5
Entropy Coef	0.01
Level Filter n Steps	64
Level Filter Sample Ratio	2

Table 2: Default ACCEL hyperparameter settings for Kinetix

Hyper-parameter	Value
Use ACCEL	True
Exploratory Grad Updates	True
Num Edits	5
Score Function	MaxMC
Level Buffer Capacity	4000
Replay Probability	0.5
Staleness Coefficient	0.3
Temperature	1.0
Top-k	8
Minimum Fill Ratio	0.5
Prioritization	Rank

Table 3: Default PLR hyperparameter settings for Kinetix

Hyper-parameter	Value
Use ACCEL	False
Exploratory Grad Updates	True
Num Edits	2
Score Function	MaxMC
Level Buffer Capacity	4000
Replay Probability	0.5
Staleness Coefficient	0.3
Temperature	1.0
Top-k	8
Minimum Fill Ratio	0.5
Prioritization	Rank

Table 4: Default SFL hyperparameter settings for Kinetix

Hyper-parameter	Value
Sampled Environments Ratio	0.5
Batch Size	4096
Num Batches	3
Rollout Steps	512
Num to Save	1024
Buffer Update Frequency	128

Table 5: Hyperparameter settings for the evaluated interventions.

Intervention	Weight Clipping	Num. Bins
Baseline	–	–
Weight Clipping (WC)	2	–
HL-Gauss (HL)	–	51
WC + CV	2	51

Table 6: Interventions evaluated in our experiments and their corresponding acronyms.

Acronym	Intervention
HL	HL-Gauss
RG	ReGen
SN-A	Spectral Normalization in the Actor
SN-B	Spectral Normalization in Both Actor and Critic
SN-C	Spectral Normalization in the Critic
WD	Weight Decay
WC	Weight Clipping

G DEEP REINFORCEMENT LEARNING

We consider an agent interacting with a discounted Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P(s' \mid s, a)$ is the transition kernel, $r(s, a)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. At each timestep t, the agent observes a state $s_t \in \mathcal{S}$, samples an action $a_t \sim \pi_\theta(\cdot \mid s_t)$ from a parameterized policy π_θ, receives reward r_t, and transitions to $s_{t+1} \sim P(\cdot \mid s_t, a_t)$. The objective is to maximize the expected discounted return

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{6}$$

The corresponding state-value and action-value functions are

$$V^{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s\right], \tag{7}$$

$$Q^{\pi_\theta}(s, a) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s,\ a_0 = a\right], \tag{8}$$

with advantage $A^{\pi_\theta}(s, a) = Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s)$ (Sutton & Barto, 2018).

We focus on actor–critic methods, which jointly optimize a policy π_θ and a learned value function V_ϕ. In Proximal Policy Optimization (PPO) (Schulman et al., 2017), trajectories are collected under the current behavior policy $\pi_{\theta_{\text{old}}}$ and reused for multiple minibatch optimization steps. Defining the probability ratio

$$\rho_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}, \tag{9}$$

the clipped policy objective is

$$\mathcal{L}_\pi(\theta) = -\mathbb{E}_t\left[\min\left(\rho_t(\theta)\widehat{A}_t,\ \text{clip}\left(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon\right)\widehat{A}_t\right)\right], \tag{10}$$

where $\widehat{A}_t$ is an estimate of the advantage and ϵ controls the clipping range. The critic is trained against trajectory-derived value targets $\widehat{V}_t$, for example using the squared-error objective

$$\mathcal{L}_V(\phi) = \mathbb{E}_t\left[\left(V_\phi(s_t) - \widehat{V}_t\right)^2\right]. \tag{11}$$

The actor and critic may use separate networks or share a learned representation. In the shared setting, both policy and value losses update the same features, coupling representation learning to both policy optimization and value prediction.

Deep actor–critic learning is inherently non-stationary. At optimization iteration k, the current policy π_{θ_k} induces a state-action occupancy distribution $d^{\pi_{\theta_k}}(s, a)$. Updating the policy changes this distribution and therefore changes the states and actions encountered in subsequent trajectories. At the same time, updates to the value function change the estimates used to construct value targets and advantages. Consequently, the learner is repeatedly optimized on changing input distributions and changing learning targets.

These dynamics can make sustained optimization difficult. In PPO, representation rank deterioration and loss of capacity have been associated with stronger policy-induced non-stationarity and, in some settings, eventual performance collapse (Moalla et al., 2024). The way on-policy data are collected also affects network plasticity and optimization stability, with larger batches of independently collected experience often improving learning behavior (Mayor et al., 2025). Separately, learning plateaus in PPO can arise when finite-sample estimates of the update objective become poor proxies for the underlying objective, highlighting an additional source of optimization stagnation in long-running on-policy training (Beukman et al., 2026). Together, these results motivate studying both policy performance and the underlying optimization and representation dynamics throughout training.

H UNSUPERVISED ENVIRONMENT DESIGN

Unsupervised environment design (UED) extends the standard RL setting from a single MDP to a space of environment instances or levels $\mathcal{L}$, where each level $\ell \in \mathcal{L}$ induces an MDP $\mathcal{M}_\ell$ (Dennis et al., 2020). Instead of fixing the training distribution over environments in advance, UED methods determine which levels the student encounters during training.

At training iteration k, let $p_k(\ell)$ denote the distribution over levels induced by the current curriculum mechanism. The student is trained to improve its expected return under this distribution,

$$J_k(\theta) = \mathbb{E}_{\ell \sim p_k} \left[J(\pi_\theta; \mathcal{M}_\ell) \right]. \tag{12}$$

When the curriculum adapts over training, p_k changes as a function of the student's learning history, so the distribution of environments defining the student's optimization problem also evolves over time.

UED methods differ in how they construct and update this curriculum. Domain-randomization baselines sample levels from a fixed generator distribution, whereas adaptive methods modify the level distribution using information gathered during training. Prioritized Level Replay (PLR), for example, preferentially revisits levels according to estimates of their learning potential (Jiang et al., 2021b). Replay-guided adversarial environment design combines replay-based level selection with environment generation and places PLR within a broader UED framework (Jiang et al., 2021a). Other approaches evolve environments toward the frontier of the student's capabilities (Parker-Holder et al., 2022), while more recent work directly prioritizes levels according to estimated learnability (**?**). Thus, the curriculum need not itself be a learned agent; it may instead be implemented through level generation, scoring, replay, sampling, search, or evolutionary mechanisms.

Let $d_\ell^{\pi_{\theta_k}}(s, a)$ denote the state-action occupancy induced by the current policy π_{θ_k} in level ℓ. The effective experience distribution at iteration k can then be written as

$$d_k(s, a) = \mathbb{E}_{\ell \sim p_k} \left[d_\ell^{\pi_{\theta_k}}(s, a) \right]. \tag{13}$$

This distribution can change through two mechanisms. First, even for a fixed level distribution, policy updates change the occupancy $d_\ell^{\pi_{\theta_k}}$. Second, under an adaptive curriculum, updates to p_k change the mixture of environments from which experience is collected. Adaptive UED therefore introduces curriculum-induced distribution shift in addition to the policy-induced non-stationarity already present in standard deep RL.

More generally, if $\mathcal{D}_k$ denotes trajectories collected by sampling $\ell \sim p_k$ and executing the current policy, the learner update can be written abstractly as

$$w_{k+1} = \mathcal{U}(w_k; \mathcal{D}_k), \tag{14}$$

where w_k contains the trainable parameters of the student. For an adaptive curriculum, we can similarly write

$$p_{k+1} = \mathcal{C}(p_k, \mathcal{H}_{k+1}), \tag{15}$$

where $\mathcal{H}_{k+1}$ denotes the information used by the curriculum, such as student trajectories, returns, level scores, or replay statistics. The learner and curriculum are therefore coupled: the current curriculum determines the experience used to update the student, while the student's behavior determines the information available for constructing subsequent curricula.

This coupling creates a particularly non-stationary learning problem. Changes in the curriculum alter the mixture of states, transitions, rewards, and task difficulties encountered during training, while policy updates simultaneously alter visitation within each level. The critic must therefore track value targets under an evolving training distribution, and the actor must optimize advantage estimates computed from experience generated by both a changing policy and, for adaptive methods, a changing curriculum. When the actor and critic share a representation, these shifts also continually change the feature-learning problem faced by the network.

We use this setting to study how curriculum-induced distribution shift interacts with the optimization and representation dynamics of deep RL agents, and in particular whether mechanisms that improve optimization stability help the student continue adapting as the training distribution evolves.

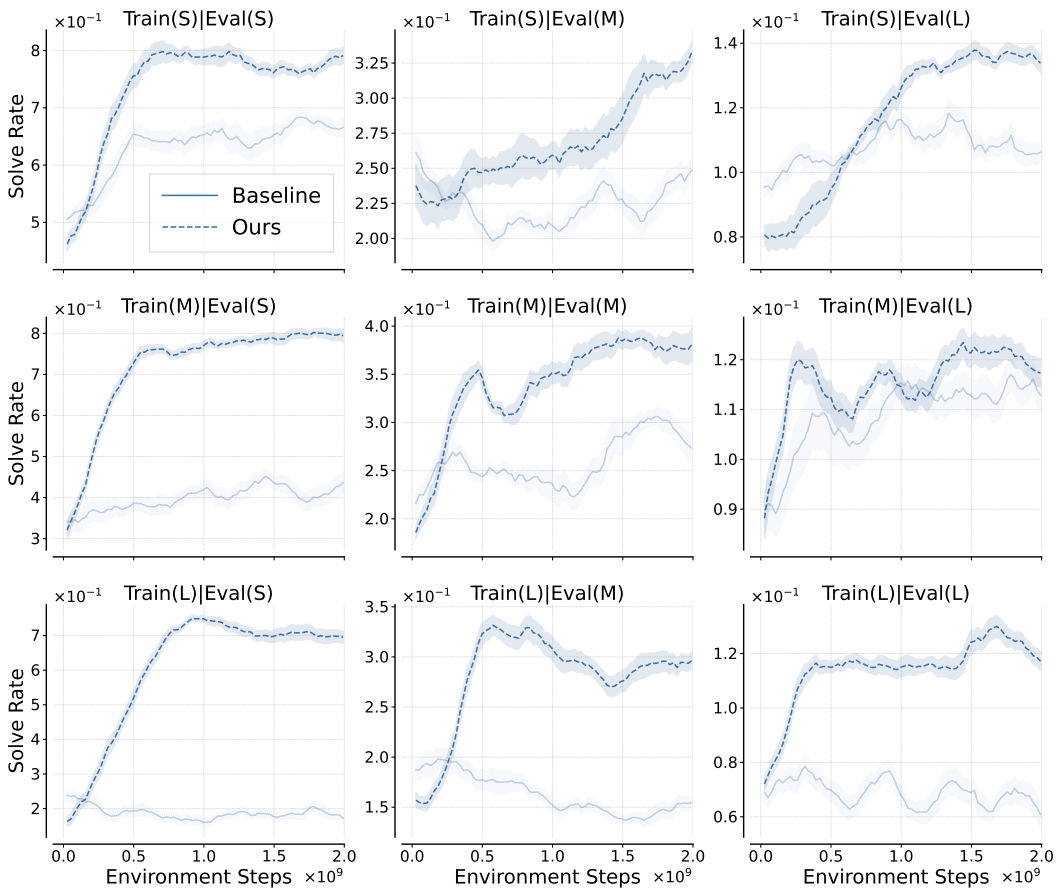

Figure 21: **Generalization to human-designed holdout tasks.** Zero-shot solve rate across training and evaluation complexities.

I EXPERIMENTS

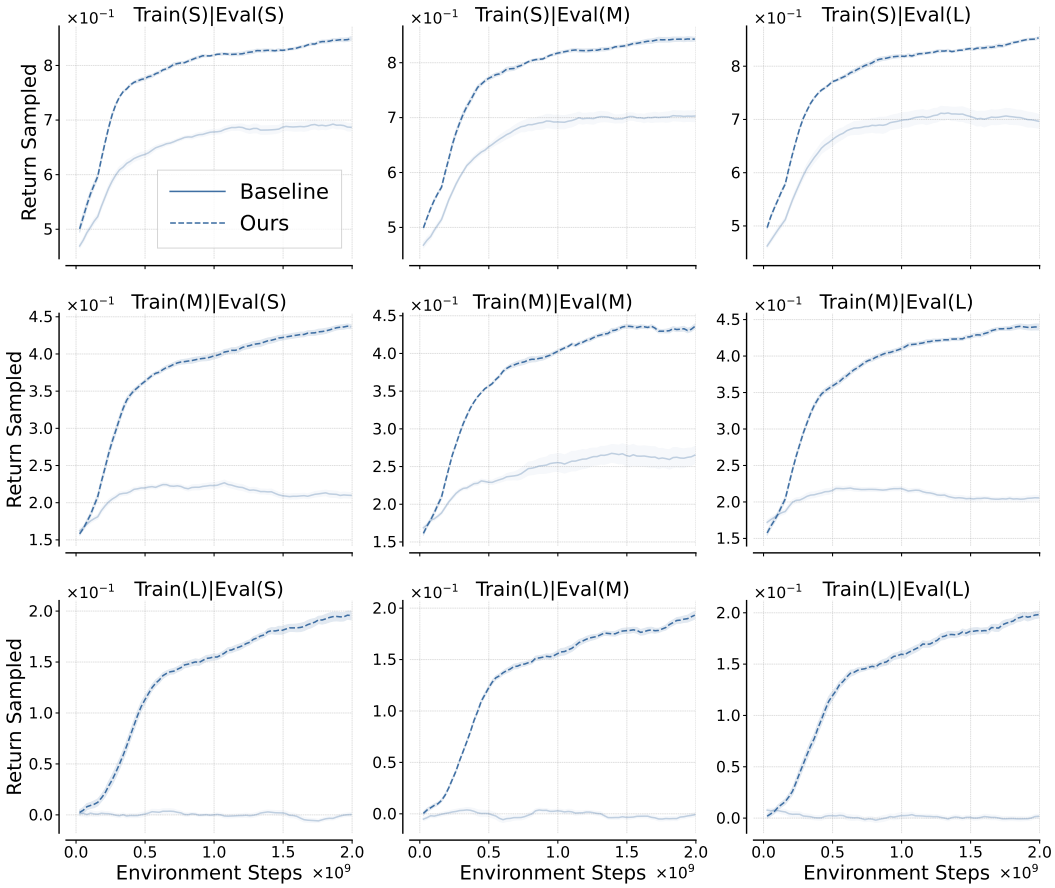

Figure 22: **Generalization to randomly generated tasks.** Zero-shot sampled return across training and evaluation complexities.