Proposal: Lookahead Learning Progress (LLP) for Unsupervised Environment Design

Eric Liu
`rundong.liu@outlook.com`

1. One-paragraph summary

Unsupervised Environment Design (UED) methods often frame the learning process as a zero-sum minimax game, which prioritizes environments where the current policy performs much worse than the optimal policy (a.k.a. high regret).

Putting aside the quality of regret surrogates, even if we have access to oracle regret, it can lead to the prioritization of environments that do not actually help the student produce immediate learning progress, but rather environments with best-case learning potential (with no guarantee of realizing such potential during actual training).

Based on observations of high-quality teacher-student interactions among humans:

> *(1) Good human teachers usually have a cooperative relationship with the student, where the goal is the student's ability to generalize from learning the curriculum. A good teacher may give slightly harder questions, but usually to serve the above purpose, instead of merely tricking the student.*
>
> *(2) If the curriculum is good, students will usually revisit it and learn more from revisiting. It is the process of researching, going back-and-forth with the teacher, or discussing with others that gives the student a sense of how good the curriculum is.*

This proposal introduces **Lookahead Learning Progress (LLP)**: a cooperative score function that measures how much a student *would improve* on an environment after a few (simulated) gradient updates. The teacher can then prioritize environments that produce immediate student improvement, rather than environments that simply maximize a regret surrogate.

2. Motivation

The Objective Misalignment (Competition vs. Cooperation)

Seminal work such as PAIRED [2] formulates curriculum generation as a minimax zero-sum game, where the teacher's (adversary) utility is the regret of the student and the student's utility is the negative of its own regret.

While theoretically elegant, the teacher may converge to distributions of tasks that are effectively impossible for the student to learn [1]. In addition, as noted by [5], regret is only a best-case signal (an upper bound on learning potential) and can therefore prioritize environments that look promising on paper but do not reliably translate into actual student improvement. Related work ("No Regrets") proposes *Sampling for Learnability* (SFL) [7] as an alternative way to bias environment selection toward learnable levels based on uncertainty of the student's performance. More broadly, the "teacher vs. student" dynamic is also a mismatch with real-world teaching: good instructors do not try to maximize the gap between an expert and a learner; they try to maximize the learner's improvement.

This motivates a **cooperative** objective: the teacher should be rewarded when the student can quickly learn from the environment that the teacher proposes.

The Metric Misalignment (Instantaneous vs. Sustained Error)

Methods such as PLR [4], Robust-PLR [3], and ACCEL [6] attempt to estimate regret with high instantaneous value-residuals. For instance, Robust-PLR uses the *Positive Value Loss* scoring function [3], which (when using GAE) takes the form:

$$S_{\text{PVL}}(\tau) = \frac{1}{H}\sum_{t=1}^{H} \max\left(\sum_{k=t}^{H}(\gamma\lambda)^{k-t}\delta_k, 0\right), \tag{1}$$

where δ_t is the TD-error at time t and λ is the GAE parameter. Similarly, *Maximum Monte Carlo (MaxMC)* replaces bootstrapped targets with the maximum achieved return on the level so far, yielding the estimator:

$$S_{\text{MaxMC}}(\tau) = \frac{1}{H}\sum_{t=1}^{H}\left(R_{\max} - V_\theta(s_t)\right). \tag{2}$$

These metrics are typically computed from a single policy's rollouts on different seeds against the same level (often using JAX Vmap). However, a level can look "hard" under the current policy while still being either (i) not learnable at all, or (ii) learnable but only after a small amount of learning. What we want the teacher to capture is the **learning progress**: how quickly the student's error decreases *after* a few optimization steps on that level. This mirrors how humans learn: mastery is demonstrated by improvement over repeated attempts of the same task, not by a single snapshot of error.

3. Formulation

Underspecified POMDP Formulation

A UPOMDP is defined as a tuple $\mathcal{M} = \langle \Theta, A, O, S, \mathcal{T}, \mathcal{O}, \mathcal{I}, R, \gamma\rangle$, where:

- Θ is the space of all possible environment configurations,
- A is the set of actions,
- O is the set of observations,
- S is the set of states,
- $\mathcal{T} : S \times A \times \Theta \to \Delta(S)$ is the transition function,
- $\mathcal{O} : S \to \Delta(O)$ is the observation function,
- $\mathcal{I} : \Theta \to \Delta(S)$ is the initial state distribution,
- $R : S \times A \times \Theta \to \mathbb{R}$ is the reward function,
- γ is the discount factor.

In this framework, the teacher selects the environment parameters $\phi \in \Lambda \subseteq \Theta$, where Λ is a finite subset of Θ. The student observes $o \in O$ and takes actions $a \in A$ to maximize rewards in the instantiated POMDP / ϕ. For simplicity, we use the term "level" interchangeably with "environment" or "POMDP."

The Student's Objective & Update Rule

The student is a parameterized RL policy $\pi_\theta : O \to \Delta(A)$. For a fixed level $\phi \in \Lambda$, the level-wise discounted return is:

$$J(\theta, \phi) = \mathbb{E}_{\tau \sim \pi_\theta, \phi}\left[\sum_{t=1}^{H}\gamma^{t-1} r_t\right]. \tag{3}$$

The student's objective is to maximize the expected return over all levels in Λ:

$$\max_\theta \ \mathbb{E}_{\phi\sim\Lambda}\left[J(\theta, \phi)\right], \tag{4}$$

After seeing a single level, the student's update rule (e.g., PPO) is an operator $\mathcal{U}$ that approximates gradient ascent on J:

$$\theta' \leftarrow \mathcal{U}(\theta, \phi) \approx \theta + \alpha\nabla_\theta J(\theta, \phi). \tag{5}$$

Student's Critic

To internalize the transition dynamics and stabilize learning, the student (typically) trains a critic by minimizing a value loss $\mathcal{L}(\theta, \phi)$. We use this as a proxy for "what the student currently does not know" on environment ϕ:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{s_t \sim \tau}\left[\left(V_\theta(s_t) - V_t^{\text{target}}\right)^2\right], \tag{6}$$

where V_t^{target} is the return target (e.g., GAE-based or Monte Carlo returns).

Score Function (Lookahead Learning Progress)

To approximate the learning progress of the student on a fixed level, we measure how much the critic loss decreases after a few updates on data from ϕ.

Concretely, let $\mathcal{U}^K(\theta, \phi)$ denote applying the student's update rule K times using rollouts on level ϕ ("virtual update attempts"). The **LLP score** of a level is the realizable reduction in value loss:

$$S_{\text{LLP}}(\phi, \theta) = \mathcal{L}(\theta, \phi) - \mathcal{L}(\mathcal{U}^K(\theta, \phi), \phi). \tag{7}$$

Intuitively, this helps the teacher prioritize levels where the student can make the most progress given a reasonable number of attempts, rather than scoring based on a single rollout.

The Cooperative Game

Putting the idea together, we would like to design the following training loop:

The teacher maintains a distribution $y \in \Delta(\Lambda)$ over levels, updated to favor high-scoring levels. At each iteration n:

$$\textbf{(Teacher / outer)} \quad y_n \propto \exp\big(S_{\text{LLP}}(\phi, \theta_n)\big) \quad \text{for each } \phi \in \Lambda, \tag{8}$$

$$\textbf{(Level sampling)} \quad \phi_n \sim y_n, \tag{9}$$

$$\textbf{(Student / inner)} \quad \theta_{n+1} = \mathcal{U}(\theta_n, \phi_n), \tag{10}$$

where the student maximizes expected return $J(\theta, \phi)$ and the teacher induces a curriculum by weighting levels according to their LLP score.

In adversarial UED (e.g., PAIRED), the score function is *regret*, which *decreases* when the student improves—the players have opposing incentives. In contrast, the LLP score $S_{\text{LLP}} = \Delta\mathcal{L}$ *increases* when the student learns. Both players benefit from the same outcome: the student improving on the selected level. This makes LLP a *team game* rather than a zero-sum game.

Remark: In practice, entropy regularization on y may be added to encourage diversity.

References

[1] Michael Beukman, Samuel Coward, Michael Matthews, Mattie Fellows, Minqi Jiang, Michael Dennis, and Jakob Foerster. Refining minimax regret for unsupervised environment design. *arXiv preprint arXiv:2402.12284*, 2024.

[2] 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*, 2020.

[3] 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*, 2021.

[4] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, 2021.

[5] Dexun Li, Wenjun Li, and Pradeep Varakantham. Marginal benefit driven rl teacher for unsupervised environment design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18253–18261, 2025.

[6] Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In *International Conference on Machine Learning*, 2022.

[7] Alexander Rutherford, Michael Beukman, Timon Willi, Bruno Lacerda, Nick Hawe, and Jakob Foerster. No regrets: Investigating and improving regret approximations for curriculum discovery. *arXiv preprint arXiv:2408.15099*, 2024.