

Transferable Curricula through Difficulty Conditioned Generators

Sidney Tio , Pradeep Varakantham

Singapore Management University
sidney.tio.2021@phdcs.smu.edu.sg, pradeepv@smu.edu.sg

Abstract

Advancements in reinforcement learning (RL) have demonstrated superhuman performance in complex tasks such as Starcraft, Go, Chess etc. However, knowledge transfer from Artificial “Experts” to humans remain a significant challenge. A promising avenue for such transfer would be the use of curricula. Recent methods in curricula generation focuses on training RL agents efficiently, yet such methods rely on surrogate measures to track student progress, and are not suited for training robots in the real world (or more ambitiously humans). In this paper, we introduce a method named *Parameterized Environment Response Model* (PERM) that shows promising results in training RL agents in parameterized environments. Inspired by Item Response Theory, PERM seeks to model difficulty of environments and ability of RL agents directly. Given that RL agents and humans are trained more efficiently under the “zone of proximal development”, our method generates a curriculum by matching the difficulty of an environment to the current ability of the student. In addition, PERM can be trained offline and does not employ non-stationary measures of student ability, making it suitable for transfer between students. We demonstrate PERM’s ability to represent the environment parameter space, and training with RL agents with PERM produces a strong performance in deterministic environments. Lastly, we show that our method is transferable between students, without any sacrifice in training quality.

1 Introduction

Consider the education of calculus. We know that there is a logical progression in terms of required knowledge before mastery in calculus can be achieved: knowledge of algebra is required, and before that, knowledge of arithmetic is required. While established progressions for optimal learning exists in education, they often require extensive human experience and investment in curriculum design. Conversely, in modern video games, mastery requires hours of playthroughs

and deliberate learning with no clear pathways to progression. In both cases, a coach or a teacher, usually an expert, is required to design such a curriculum for optimal learning. Scaffolding this curriculum can be tedious and in some cases, intractable. More importantly, it requires deep and nuanced knowledge of the subject matter, which may not always be accessible.

The past decade has seen an explosion of Reinforcement Learning (RL, [Sutton *et al.*, 1998]) methods that achieve superhuman performance in complex tasks such as DOTA2, Starcraft, Go, Chess, etc. ([Berner *et al.*, 2019], [Arulkumar *et al.*, 2019], [Silver *et al.*, 2016], [Silver *et al.*, 2017]) Given the state-of-the-art RL methods, we propose to explore methods that exploit expert-level RL agents for knowledge transfer to humans and to help shortcut the learning process. One possible avenue for such a transfer to take place would be the use of curricula.

Recent methods in curricula generation explores designing a curricula through *Unsupervised Environment Design* (UED, [Dennis *et al.*, 2020]). UED formalizes the problem of finding adaptive curricula in a teacher-student paradigm, whereby a teacher finds useful environments that optimizes student learning, while considering student’s performance as feedback. While prior work in UED (e.g. [Parker-Holder *et al.*, 2022], [Du *et al.*, 2022], [Li *et al.*, 2023a], [Li *et al.*, 2023b]) has trained high-performing RL students on the respective environments, these method rely on surrogate objectives to track student progress, or co-learn with another RL agent ([Dennis *et al.*, 2020], [Du *et al.*, 2022], [Parker-Holder *et al.*, 2022]), both of which would be impractical for transfer between students (artificial or agents in real world settings alike).

For transfer between students, We require methods that do not use additional RL students, or are able to directly track student’s learning progress. In this work, we introduce the Item Response Theory (IRT, [Embretson and Reise, 2013]) as a possible solution to this problem. The IRT was developed as a mathematical framework to reason jointly about a student’s ability and the questions which they respond to. Considered to be ubiquitous in the field of standardized testing, it is largely used in the design, analysis, and scoring of tests, questionnaires([Hartshorne *et al.*, 2018], [Harlen, 2001], [Łuniewska *et al.*, 2016]), and instruments that measure ability, attitudes, or other latent variables. The IRT al-

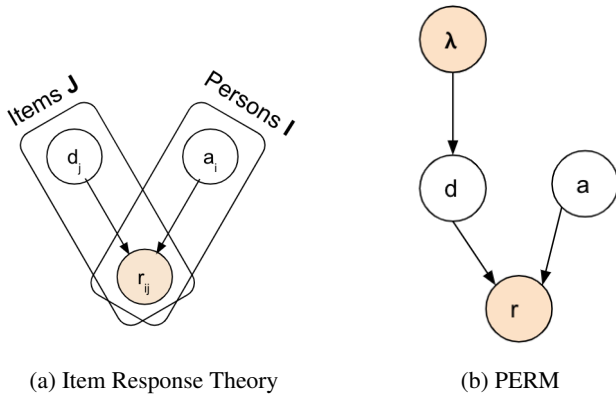


Figure 1: Graphical representation of IRT and PERM. λ , a , d , r represents environment parameters, ability, difficulty, and response respectively. White nodes depict latent variables, while tan colored nodes represent observable variables.

allows educators to quantify the “difficulty” of a given test item by modelling the relationship between a test taker’s response to the test item and the test taker’s overall ability. In the context of UED, we then see that the IRT provides a useful framework for us to understand the difficulty of a parameterized environment with regards to the ability of the student, of which we aim to maximize.

Our current work proposes a new algorithm, called *Parameterized Environment Response Model*, or PERM. PERM applies the IRT to the UED context and generates curricula by matching environments to the ability of the student. Since we do not use a RL-based teacher or regret as a feedback mechanism, our method is transferable across students, regardless of artificial or human student.

Our main contributions are as follows:

1. We propose PERM, a novel framework to assess student ability and difficulty of parameterized environments.
2. PERM produces curricula by generating environments that matches the ability of the student.
3. We investigate PERM’s capabilities in modelling the training process with latent representations of difficulty and ability.
4. We compare agents trained with PERM with other UED methods in parameterized environments.

2 Related Work

2.1 Item Response Theory

In Psychology and Education, the IRT is a method used to model interactions between a test taker’s ability and a certain characteristic of a question, usually difficulty. The goal here is to gauge a student’s ability based on their response to items of varying difficulties. The IRT has many forms, but for the purposes of this paper, we focus on the most standard form: the 1-Parameter Logistic (1PL) IRT, also known as the Rasch model [Rasch, 1993], and a continuous variant. The Rasch

model is given in Eq. 1,

$$p(r_{i,j} = 1|a_i, d_j) = \frac{1}{1 + \exp^{-(a_i - d_j)}} \quad (1)$$

where $r_{i,j}$ is the response by the i -th person, with an ability measure a_i , to the j -th item, with a difficulty measure d_j . The graph of the 1PL IRT can be seen in Figure 1a. We see that the Rasch model is equivalent to a logistic function. Therefore, the probability that a student answers the item correctly is a function of the difference between student ability a_i and item difficulty d_j .

In RL settings, interactions between agent and environment is summarized by cumulative rewards achieved. For us to adopt the IRT for the environment training scenario, we can then replace the logistic function with a normal ogive model [Samejima, 1974], or the cumulative distribution of a standard normal distribution. Eq 1 then becomes:

$$p(Z \leq r_{i,j}|a_i, d_j) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{a_i - d_j} \exp\{-\frac{u^2}{2}\} du \quad (2)$$

To our knowledge, the IRT has not used to train RL agents, nor used as a knowledge transfer method from an RL agent to another student.

While earlier works have used different methods to perform inference for IRT, a recent method, VIBO [Wu *et al.*, 2020], introduces a variational inference approach to estimate IRT. More critically, the formulation of the IRT as a variational inference problem allows us to exploit the learned representation to generate new items. We discuss our modifications to VIBO in Section 3.

2.2 Zone of Proximal Development

Prior work in UED discusses the zone of proximal development [Vygotsky and Cole, 1978], loosely defined as problems faced by the student that are not too easy (such that there is no learning value for the student) and not too difficult (such that it is impossible for the student).

PAIRED [Dennis *et al.*, 2020] features an adversarial teacher whose task is to generate environments that maximizes the regret between the protagonist student and another antagonist agent. To apply our method to human training, a human-RL pairing would be necessary, but the difference in learning rates and required experiences could create bottlenecks for the human student (i.e. Moravec’s Paradox [Moravec, 1988]).

PLR [Jiang *et al.*, 2021b], and its newer variants ([Parker-Holder *et al.*, 2022], [Jiang *et al.*, 2021a]), maintains a store of previously seen levels and prioritizes the replay levels where the average Generalized Advantage Estimate (GAE, [Schulman *et al.*, 2015]) is large. The use of GAE requires access to the value function of the student, a feature that is currently not operationalized for human subjects.

In summary, teacher-student curriculum generation approaches have predominantly focused on the zone of proximal development but have relied on surrogate objectives to operationalize it, without directly measuring difficulty or student ability. However, these surrogate objectives are often non-stationary and not easily transferable between students.

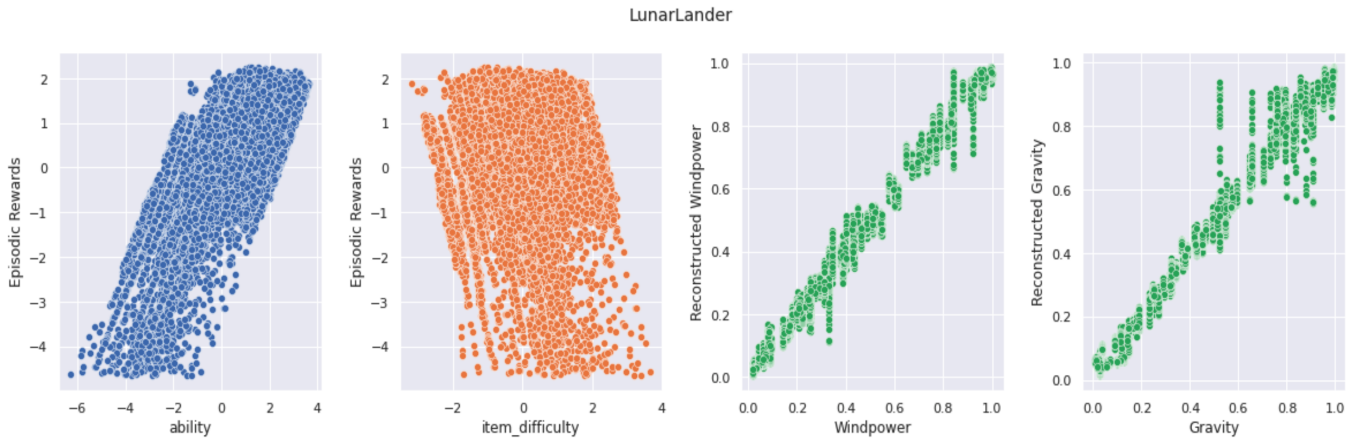


Figure 2: Analysis of PERM’s reconstruction capabilities on LunarLander. Blue and orange plots represent ability and difficulty estimates against actual rewards achieved by agent; latent variables learned by PERM correspond to actual reward accordingly. Green plots visualizes the real environment parameters against parameters recovered by PERM. PERM is able to reconstruct the environment parameters from difficulty. Similar results are obtained in BipedalWalker, as seen in Figure 3

Moreover, resulting curricula may not adequately accommodate large changes in student ability, which is a critical limitation for human subjects.

2.3 Task Sequencing

Related to UED are previous works in Task Generation and Sequencing for Curriculum Learning, which aim to generate and assign tasks to a student agent in a principled order to optimize performance on a final objective task [Narvekar *et al.*, 2020]. Most of the literature in Task Generation focuses on modifying the student agent’s MDP to generate different tasks (e.g. [Fogolino *et al.*, 2019], [Narvekar *et al.*, 2017], [Racaniere *et al.*, 2019]). For example, *promising initialization* [Narvekar *et al.*, 2016] modifies the set of initial states and generates a curriculum by initializing agents in states close to high rewards. On the other hand, *action simplification* [Narvekar *et al.*, 2016] seeks to prune the action set of the student agent to reduce the likelihood of making mistakes.

In contrast to Task Generation, the UED framework investigates domains where there is no explicit representation of tasks. Here, the student agent must learn to maximize rewards across a variety of environment parameters in open-ended domains without a target “final task” to learn. In the UED framework, the teacher algorithms only influences the environment parameters λ , while other features of the student’s MDP remains relatively consistent across training. While prior works in task sequencing generate different tasks by directly modifying the student agent’s MDP, we leave the curricula generation for such domains for future work.

3 Method

In this section we introduce a novel algorithm to train students, combining the training process with an IRT-based environment generator that acts as a teacher. The teacher’s goal is to train a robust student agent that performs well across different environment parameters, while the student’s goal is to

maximize its rewards in a given environment. Unlike previous UED methods which relies on proxy estimates of difficulty and environment feasibility (e.g. Regret [Dennis *et al.*, 2020]), we propose to directly estimate difficulty by formulating the training process as a student-item response process and model the process with IRT.

Given a teacher model that is able to estimate both ability of the student and difficulty of an item, we are able to present a sequence of environments that are within the student’s zone of proximal development. By providing environments within the zone, we are unlikely to generate infeasible environments that are impossible or too difficult for the current student, while also avoiding trivial environments that provide little to no learning value. As our method does not rely on non-stationary objectives such as regret, we are able to train PERM offline and transfer knowledge to any student agent, including human students. Lastly, because our method only relies on environment parameters and student response, it works in virtually any environment without requiring any expert knowledge. We show in later sections that PERM serves as a good exploration mechanism to understand the parameter-response relation given any environment.

PERM can be separated into two components: (i) learning latent representations of ability and difficulty; (ii) generating a curriculum during agent training. The algorithm is summarized in Algorithm 1.

3.1 Preliminaries

We draw parallels from UED to IRT by characterizing each environment parameter λ as an item which the student agent with a policy π_t ‘responds’ to by interacting and maximizes its own reward r . Specifically, each student interaction with the parameterized environment yields a tuple (π_t, λ_t, r_t) , where π_t represents the student policy at t -th interaction, and achieves reward r_t during its interaction with the environment parameterized by λ_t . We then use a history of such interac-

Algorithm 1 Curriculum Generation for RL Agents with PERM

Input: Environment E , Environment parameters λ , Student Agent π

Parameter: k episode frequency before update

Output: Trained Student Agent, Trained PERM

```

1: Let  $t = 0, \lambda_0 \sim \text{Uniform}(\lambda)$ 
2: while not converged do
3:   for  $k$  episodes do
4:     Collect Reward  $r_t$  from agent  $\pi$  playthrough of  $E(\lambda_t)$ .
5:     Estimate current ability  $\mu_{a_t}, \sigma_{a_t}$  by computing  $q_\phi(a|d, r, \lambda)$ 
6:     Sample current ability  $a_t \sim N(\mu_{a_t}, \sigma_{a_t})$ 
7:     Get next difficulty  $d_{t+1} \leftarrow a_t$ 
8:     Generate next parameters  $\lambda_{t+1} \leftarrow p_\theta(\lambda|d_{t+1})$ 
9:      $t \leftarrow t + 1$ 
10:  end for
11:  Update PERM with  $\mathcal{L}_{PERM}$ 
12:  RL Update on Student Agent
13: end while
14: return trained student agent  $\pi$ , trained PERM
    
```

tions to learn latent representations of student ability $a \in \mathbb{R}^n$ and item difficulty $d \in \mathbb{R}^n$, where $a \propto r$ and $d \propto \frac{1}{r}$. In this formulation, π_t at different timesteps are seen as students independent of each other.

3.2 Learning Latent Representations of Ability and Difficulty

Following from Wu et al’s Variational Item Response Theory (VIBO, [2020]), we use a Variational Inference problem [Kingma and Welling, 2013] formulation to learn latent representation of any student interaction with the environment. More critically, VIBO proposes the amortization of the item and student space, which allows it to scale from discrete observations of items, to a continuous parameter space such as the UED. From here, we drop the subscript for the notations a, d , and r to indicate our move away from discretized items and students.

Enabling Generation of Environment Parameters

The objective of VIBO is to learn the latent representation of student ability and difficulty of items. In order for us to generate the next set of environment parameters λ_{t+1} for the student to train on, we modify VIBO to include an additional decoder to generate λ given a desired difficulty estimate d . The graphical form of PERM can be seen in Figure 1b.

We state and prove the revised PERM objective based on Variational Inference in the following theorem. We use notation consistent with the Variational Inference literature, and refer the motivated reader to [Kingma and Welling, 2013] for further reading.

Theorem 1. *Let a be the ability for any student, and d be the difficulty of any environment parameterized by λ . Let r be the*

continuous response from the student on the environment. If we define the PERM objective as

$$\mathcal{L}_{PERM} \triangleq \mathcal{L}_{recon_r} + \mathcal{L}_{recon_\lambda} + \mathcal{L}_A + \mathcal{L}_D \quad (3)$$

where

$$\begin{aligned} \mathcal{L}_{recon_r} &= \mathbb{E}_{q_\phi(a,d|r,\lambda)}[\log p_\theta(r|a, d)] \\ \mathcal{L}_{recon_\lambda} &= \mathbb{E}_{q_\phi(a,d|r,\lambda)}[\log p_\theta(\lambda|d)] \\ \mathcal{L}_A &= \mathbb{E}_{q_\phi(a,d|r,\lambda)}\left[\log \frac{p(a)}{q_\phi(a|d, r, \lambda)}\right] \\ &= \mathbb{E}_{q_\phi(d|r,\lambda)}[D_{KL}((q_\phi(a|d, r))\|p(a))] \\ \mathcal{L}_D &= \mathbb{E}_{q_\phi(a,d|r,\lambda)}\left[\log \frac{p(d)}{q_\phi(d|r, \lambda)}\right] \\ &= \mathbb{E}_{q_\phi(d|r,\lambda)}\left[\log \frac{p(d)}{q_\phi(d|r, \lambda)}\right] \\ &= D_{KL}((q_\phi(d|r, \lambda))\|p(d)) \end{aligned} \quad (4)$$

and assume the joint posterior factorizes as follows:

$$q_\phi(a, d|r, \lambda) = q_\phi(a|d, r, \lambda)q_\phi(d|r, \lambda) \quad (5)$$

then $\log p(r) + (\lambda) \geq \mathcal{L}_{PERM}$; \mathcal{L}_{PERM} is a lower bound of the log marginal probability of a response r .

Proof. Expand the marginal and apply Jensen’s inequality:

$$\begin{aligned} \log p_\theta(r) + \log p_\theta(\lambda) &\geq \mathbb{E}_{q_\phi(a,d|r)}\left[\log \frac{p_\theta(r, a, d, \lambda)}{q_\phi(a, d|r, \lambda)}\right] \\ &= \mathbb{E}_{q_\phi(a,d|r)}[\log p_\theta(r|a, d)] \\ &\quad + \mathbb{E}_{q_\phi(a,d|r)}[\log p_\theta(\lambda|d)] \\ &\quad + \mathbb{E}_{q_\phi(a,d|r)}\left[\log \frac{p(a)}{q_\phi(a|d, r, \lambda)}\right] \\ &\quad + \mathbb{E}_{q_\phi(a,d|r)}\left[\log \frac{p(d)}{q_\phi(d|r, \lambda)}\right] \\ &= \mathcal{L}_{recon_r} + \mathcal{L}_{recon_\lambda} + \mathcal{L}_A + \mathcal{L}_D \end{aligned}$$

Since $\mathcal{L}_{PERM} = \mathcal{L}_{recon_r} + \mathcal{L}_{recon_\lambda} + \mathcal{L}_A + \mathcal{L}_D$ and KL divergences are non-negative, we have shown that \mathcal{L}_{PERM} is a lower bound on $\log p_\theta(r) + \log p_\theta(\lambda)$.

For easy reparameterization, all distributions $q_\phi(\cdot)$ are defined as Normal distributions with diagonal covariance. \square

3.3 Generating Environments for Curricula

Our method makes a core assumption that optimal learning takes place when the difficulty of the environment matches the ability of the student. In the continuous response model given in Eq. 2, we see that when ability and difficulty is matched (i.e. $a_i = d_j$), the probability which the student achieves a normalized average score $r_{i,j} = 0$ is 0.5. This is a useful property to operationalize the zone of proximal development, as we can see that the model estimates an equal probability of the student overperforming or underperforming.

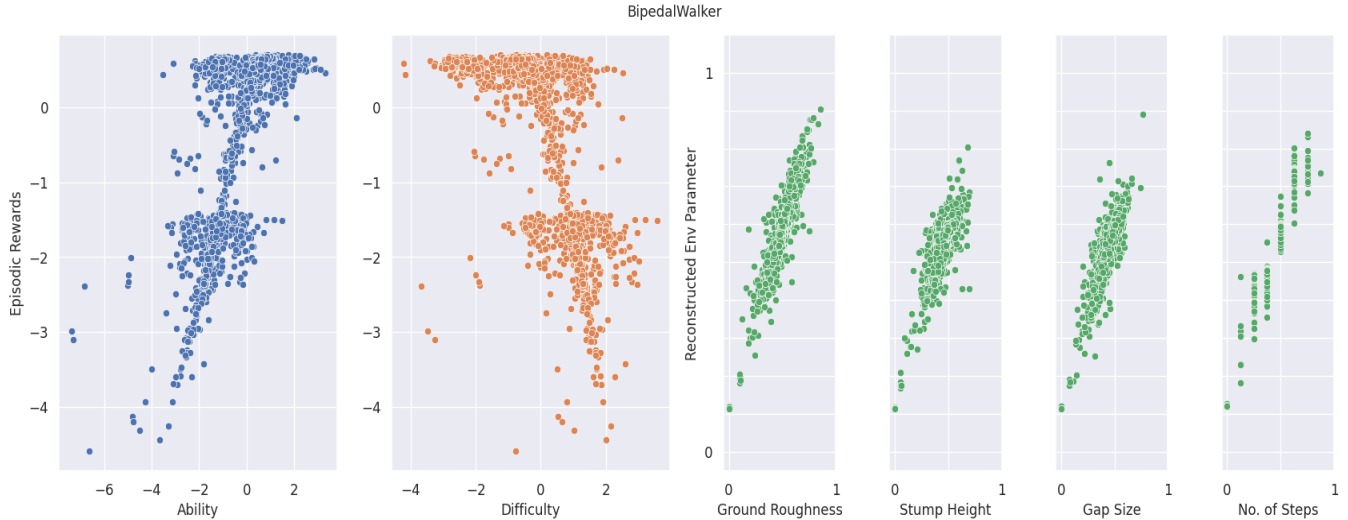


Figure 3: Analysis of PERM’s reconstruction capabilities on BipedalWalker. Blue and orange plots represent ability and difficulty estimates against actual rewards achieved by agent; latent variables learned by PERM correspond to actual reward accordingly. Green plots visualizes the real environment parameters against parameters recovered by PERM. PERM is able to reconstruct the environment parameters from difficulty. Values presented are normalized.

Training is initialized by uniformly sampling across the range of environment parameters. After each interaction between the student and the environment, PERM estimates the ability a_t of the student given the episodic rewards and parameters of the environment. PERM then generates the parameters of the next environment $\lambda_{t+1} \sim p_{\theta}(\lambda|d_{t+1})$ where $d_{t+1} = a_t$.

4 Experiments

In our experiments, we seek to answer the following research questions (RQ): **RQ1**: How well does PERM represent the environment parameter space with ability and difficulty measures? **RQ2**: How do RL Agents trained by PERM compare to other UED baselines?

We compare two variants of PERM, PERM-Online and PERM-Offline, with the following baselines: PLR[⊥] (Robust Prioritized Replay, [Jiang et al., 2021a]), PAIRED [Dennis et al., 2020], Domain Randomization(DR, [Tobin et al., 2017]). PERM-Online is our method that is randomly initialized and trained concurrently with the student agent, as described in Algorithm 1; PERM-Offline is trained separately from the student agent, and remains fixed throughout the student training. PERM-Offline is used to investigate its performance when used in an offline manner, similar to how we propose to use it for human training.

For all experiments, we train a student PPO agent [Schulman et al., 2017] in OpenAI Gym’s *LunarLander* and *BipedalWalker* [Brockman et al., 2016].

We first evaluate PERM’s effectiveness in representing the parameter space on both OpenAI environments. Specifically, we evaluate how the latent variables ability a and difficulty d correlate to the rewards obtained in each interaction, as well as its capability in generating environment parameters. We then provide a proof-of-concept of PERM’s curricula gen-

Env	Response MSE	λ MSE	R-Squared
LunarLander	7.8×10^{-5}	0.001	1.00
BipedalWalker	2.5×10^{-4}	0.001	0.986

Table 1: Analysis of PERM’s recovery capabilities. PERM is able to reconstruct the response and environment parameters with great accuracy. R-squared is obtained by regressing ability and difficulty on response.

eration on the *LunarLander* environment, which has only two environment parameters to tune. Lastly, we scale to the more complex *BipedalWalker* environment that has eight environment parameters, and compare the performance of the trained agent against other methods using the same evaluation environment parameters as in Parker-Holder et al [2022].

4.1 Analyzing PERM’s Representation of Environment Parameters

We begin by investigating PERM’s capabilities in representing and generating the environment parameters. In order to establish PERM’s capabilities for curricula generation purposes, PERM needs to demonstrate the following: i) the latent representations ability a and difficulty d needs to conform to proposed relationship with response r (i.e. $a \propto r$ and $d \propto \frac{1}{r}$); ii) given input environment parameters λ and response r , the reconstructed environment parameters λ' and response r' need to match its inputs. For both analyses, we rely on correlation metrics and mean-squared error (MSE) to establish PERM’s capabilities.

We first train PERM by collecting agent-environment interactions from training a PPO agent under a DR framework until convergence. We then train an offline version of PERM using a subset of data collected and Equation 3. We use the remaining data collected as a holdout set to evaluate PERM’s

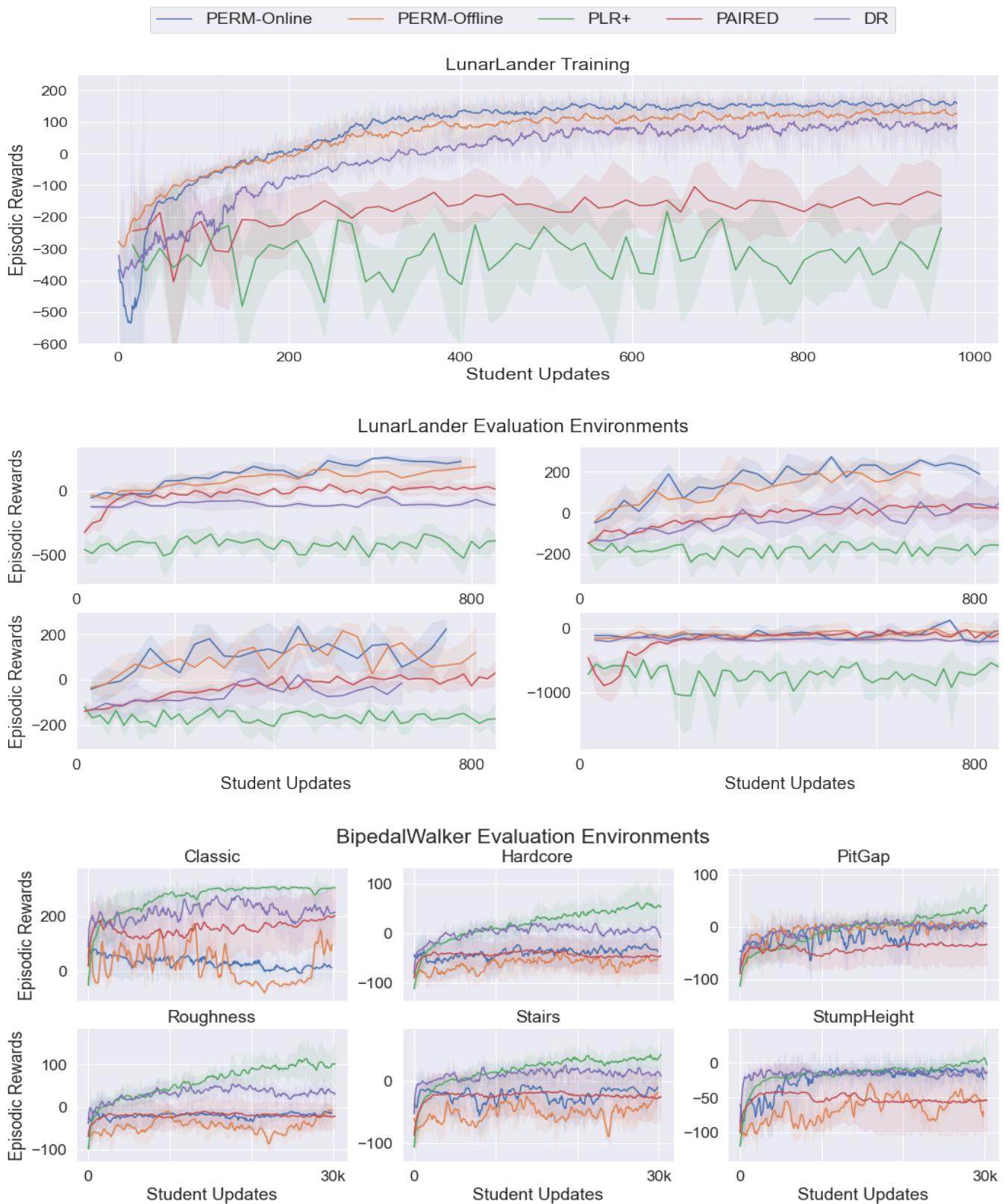


Figure 4: Agents trained by PERM-Online and Offline outperform other methods on LunarLander in both training and evaluation environments. Top: Performance on LunarLander during training; Middle: Performance on selected LunarLander evaluation environments; Bottom: Performance on BipedalWalker evaluation environments.

performance. The results are visualized in Figure 2 and Figure 3, and summary statistics are provided in Table 1. As we see in both plots, the latent representations a (blue) and d (orange) largely correlates with our expectations of its respective relationships with the response variable r . When both ability and difficulty are regressed against the response variable, we achieve a R-squared of 1.00 and 0.986 for LunarLander and BipedalWalker respectively, indicating that both latent representations are perfect predictors of reward achieved by an agent in a given parameterized environment. Turning to PERM’s capability in generating environment parameters (Figure 2 & 3, green), we see that PERM achieves near perfect recovery of all environment parameters on the test set, as indicated by the MSE between input parameters and recovered parameters. Taking the strong results of PERM in recovering environment parameters from the latent variables, we proceed to generate curricula to train RL Agents.

4.2 Training RL Agents with PERM

LunarLander

We next apply PERM’s environment generation capabilities to train an agent in LunarLander. In this domain, student agents control the engine of a rocket-like structure and is tasked to land the vehicle safely. Before each episode, teacher algorithms determine the gravity and wind power present in a given playthrough, which directly effects the difficulty of landing the vehicle safely. We train student agents for $1e6$ environment timesteps, and periodically evaluate the agent on test environments. The parameters for the test environments are randomly generated and fixed across all evaluations, and are provided in the Appendix. We report the training and evaluation results in Figure 4 top and middle plots respectively. As we see, student agents trained with PERM achieves stronger performance over all other methods, both during training and evaluation environments. More importantly, we note that despite training PERM-Offline on a different student, the RL agent under PERM-Offline still maintains its training effectiveness over other methods.

We note that despite a reasonably strong performance of an agent trained under DR, DR has a greater possibility of generating environments that are out of the student’s range of ability. We observe that episode lengths for students trained under DR are shorter (mean of 244 timesteps vs 311 timesteps for PERM), indicating a larger proportion of levels where the student agent immediately fails. PERM, by providing environments that are constantly within the student’s capabilities, is more sample efficient than DR.

BipedalWalker

Finally, we evaluate PERM in the modified BipedalWalker from Parker-Holder et al. [2022]. In this domain, student agents are required to control a bipedal vehicle and navigate across a terrain. The teacher agent is tasked to select the range of level properties in the terrain, such as the minimum and maximum size of a pit. The environment is then generated by uniformly sampling from the parameters. We train agents for about 3 billion environment steps, and periodically evaluate the agents for about 30 episodes per evaluation environment. The evaluation results are provided in Figure 4, bottom.

In the BipedalWalker’s evaluation environments, student agent trained by PERM produced mixed results, notably achieving comparable performance to PLR^+ in the `StumpHeight` and `PitGap` environment, and comparable performance to PAIRED in others. As BipedalWalker environment properties are sampled from the environment parameters generated by the teacher, it is likely that the buffer-based PLR^+ that tracks seeds of environments had a superior effect in training our student agents. PERM, on the other hand, is trained to only generate the ranges of the environment properties, which results in non-deterministic environment generation despite the same set of parameters.

5 Conclusion and Future Work

We have introduced PERM, a new method that characterizes the agent-environment interaction as a student-item response paradigm. Inspired by Item Response Theory, we provide a method to directly assess the ability of a student agent, and the difficulty associated with parameters of a simulated environment. We proposed to generate curricula by evaluating the ability of the student agent, then generating environments that match the ability of the student. Since PERM does not rely on non-stationary measures of ability such as Regret, our method allows us to predict ability and difficulty directly across different students. Hence, our approach is transferable and is able to adapt to learning trajectories of different students. Theoretically, we could use PERM to train humans in similarly parameterized environments.

We have demonstrated that PERM produces strong representation of both parameterized environments, and is a suitable approach in generating environment parameters with desired difficulties. Finally, we trained RL agents with PERM in our selected environments, and found that our method outperformed the other methods in the deterministic environment, LunarLander.

Most recently, Zhuang et al. [2022] proposed to use a IRT-based model in Computerized Adaptive Testing (CAT) on humans, to some success. The objective of CAT is to accurately predict the student’s response to a set of future questions, based on her response to prior questions. We look forward to deploying PERM or IRT-based models in real world settings for training purposes. We hope that our results inspire research into methods that are able to train both humans and RL Agents effectively.

Acknowledgements

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-PhD/2022-01-025).

References

- [Arulkumaran et al., 2019] Kai Arulkumaran, Antoine Cully, and Julian Togelius. Alphastar: An evolutionary computation perspective. In *Proceedings of the genetic and evolutionary computation conference companion*, pages 314–315, 2019.

- [Berner *et al.*, 2019] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.
- [Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [Dennis *et al.*, 2020] Michael Dennis, Natasha Jaques, Eugene Vinitzky, 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.
- [Du *et al.*, 2022] Yuqing Du, Pieter Abbeel, and Aditya Grover. It takes four to tango: Multiagent selfplay for automatic curriculum generation. *arXiv preprint arXiv:2202.10608*, 2022.
- [Embretson and Reise, 2013] Susan E Embretson and Steven P Reise. *Item response theory*. Psychology Press, 2013.
- [Fogolino *et al.*, 2019] Francesco Fogolino, Christiano Coletto Christakou, and Matteo Leonetti. An optimization framework for task sequencing in curriculum learning. In *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 207–214. IEEE, 2019.
- [Harlen, 2001] Wynne Harlen. *The Assessment of Scientific Literacy in the OECD/PISA Project*, pages 49–60. Springer Netherlands, Dordrecht, 2001.
- [Hartshorne *et al.*, 2018] Joshua K Hartshorne, Joshua B Tenenbaum, and Steven Pinker. A critical period for second language acquisition: Evidence from 2/3 million english speakers. *Cognition*, 177:263–277, 2018.
- [Jiang *et al.*, 2021a] 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, 2021.
- [Jiang *et al.*, 2021b] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pages 4940–4950. PMLR, 2021.
- [Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [Li *et al.*, 2023a] Dexun Li, Wenjun Li, and Pradeep Varakantham. Diversity induced environment design via self-play. *arXiv preprint arXiv:2302.02119*, 2023.
- [Li *et al.*, 2023b] Wenjun Li, Pradeep Varakantham, and Dexun Li. Effective diversity in unsupervised environment design. *arXiv preprint arXiv:2301.08025*, 2023.
- [Łuniewska *et al.*, 2016] Magdalena Łuniewska, Ewa Haman, Sharon Armon-Lotem, Bartłomiej Etenkowski, Frenette Southwood, Darinka Anelković, Elma Blom, Tessel Boerma, Shula Chiat, Pascale Engel de Abreu, et al. Ratings of age of acquisition of 299 words across 25 languages: Is there a cross-linguistic order of words? *Behavior research methods*, 48(3):1154–1177, 2016.
- [Moravec, 1988] Hans Moravec. *The mind children: The future of robot and human intelligence*. Harvard University Press, 1988.
- [Narvekar *et al.*, 2016] Sanmit Narvekar, Jivko Sinapov, Matteo Leonetti, and Peter Stone. Source task creation for curriculum learning. In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems*, pages 566–574, 2016.
- [Narvekar *et al.*, 2017] Sanmit Narvekar, Jivko Sinapov, and Peter Stone. Autonomous task sequencing for customized curriculum design in reinforcement learning. In *IJCAI*, pages 2536–2542, 2017.
- [Narvekar *et al.*, 2020] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *The Journal of Machine Learning Research*, 21(1):7382–7431, 2020.
- [Parker-Holder *et al.*, 2022] Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. *arXiv preprint arXiv:2203.01302*, 2022.
- [Racaniere *et al.*, 2019] Sebastien Racaniere, Andrew K Lampinen, Adam Santoro, David P Reichert, Vlad Firoiu, and Timothy P Lillicrap. Automated curricula through setter-solver interactions. *arXiv preprint arXiv:1909.12892*, 2019.
- [Rasch, 1993] Georg Rasch. *Probabilistic models for some intelligence and attainment tests*. ERIC, 1993.
- [Samejima, 1974] Fumiko Samejima. Normal ogive model on the continuous response level in the multidimensional latent space. *Psychometrika*, 39(1):111–121, 1974.
- [Schulman *et al.*, 2015] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.
- [Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [Silver *et al.*, 2016] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
- [Silver *et al.*, 2017] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by

self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

- [Sutton *et al.*, 1998] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.
- [Tobin *et al.*, 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 *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.
- [Vygotsky and Cole, 1978] Lev Semenovich Vygotsky and Michael Cole. *Mind in society: Development of higher psychological processes*. Harvard university press, 1978.
- [Wu *et al.*, 2020] Mike Wu, Richard L Davis, Benjamin W Domingue, Chris Piech, and Noah Goodman. Variational item response theory: Fast, accurate, and expressive. *arXiv preprint arXiv:2002.00276*, 2020.
- [Zhuang *et al.*, 2022] Yan Zhuang, Qi Liu, Zhenya Huang, Zhi Li, Shuanghong Shen, and Haiping Ma. Fully adaptive framework: Neural computerized adaptive testing for online education. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4734–4742, 2022.