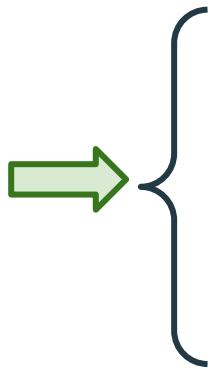


Evolving Curricula with Regret-Based Environment Design

Jack Parker-Holder*, Minqi Jiang*, Michael Dennis, Mikayel Samvelyan,
Jakob Foerster, Edward Grefenstette, Tim Rocktäschel



Agents should be robust to variations in the environment



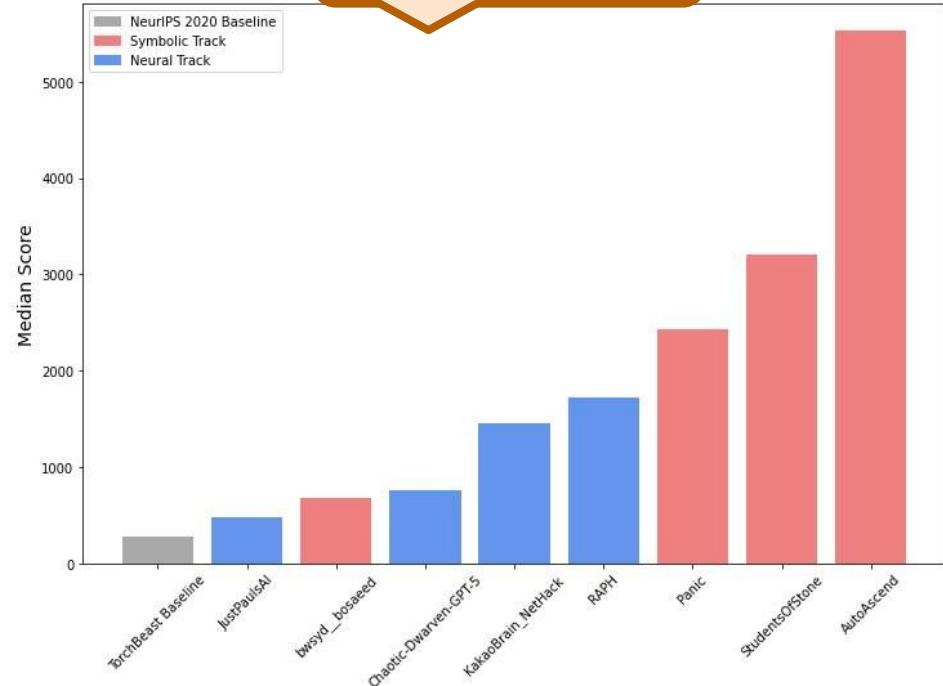
The “Problem Problem”

Too easy for RL



Zhang et al. 2021

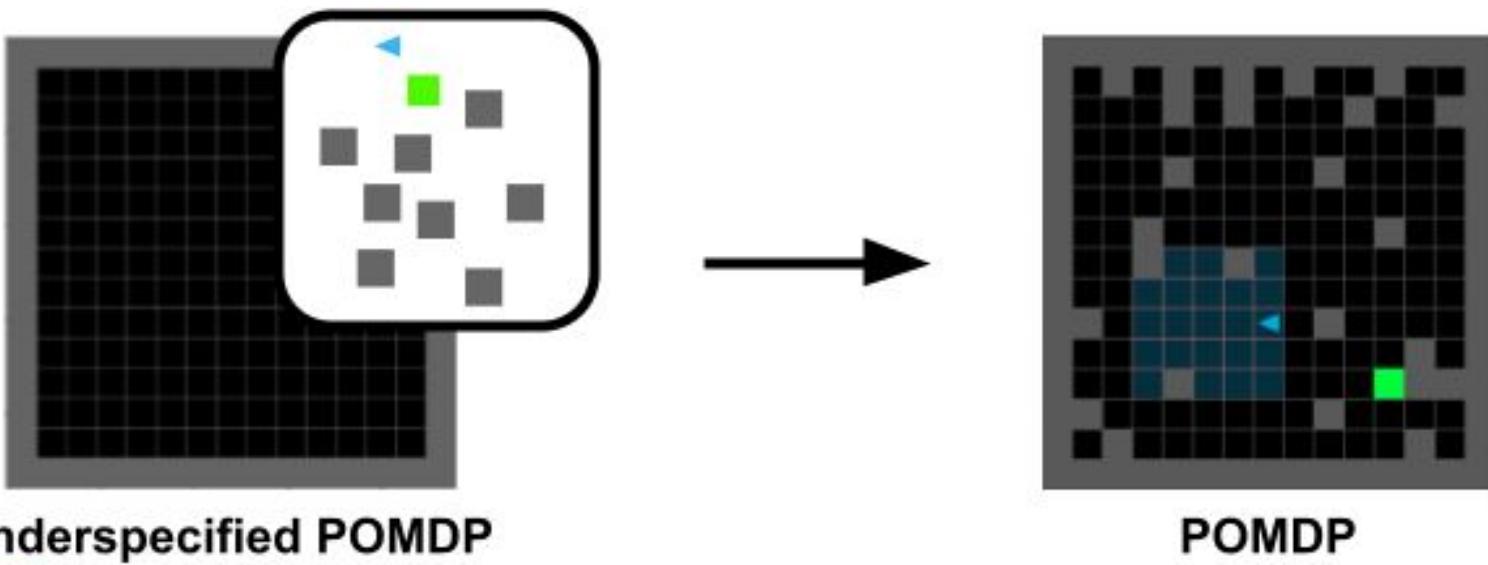
Too hard for RL



Hambro et al. 2021

Unsupervised Environment Design

We can represent parameterized environments as Underspecified POMDPs (UPOMDPs).



Underspecified POMDP

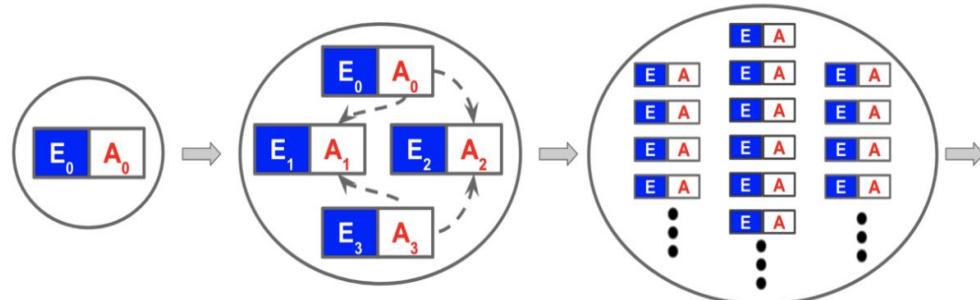
$$\langle A, O, \Theta, S^M, \mathcal{T}^M, \mathcal{I}^M, \mathcal{R}^M, \gamma \rangle$$

POMDP

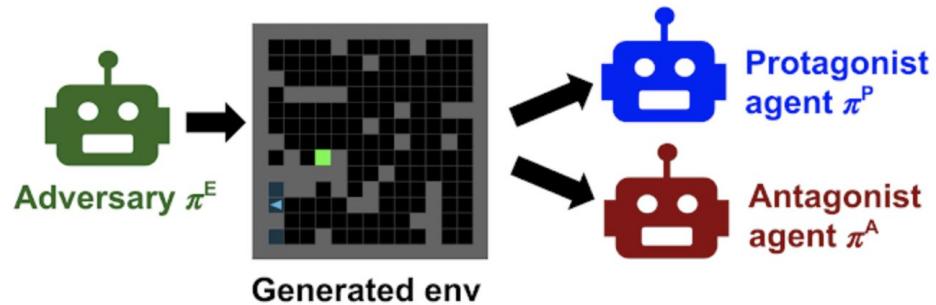
$$\langle A, O, S, \mathcal{T}, \mathcal{I}, \mathcal{R}, \gamma \rangle$$

Previous Work

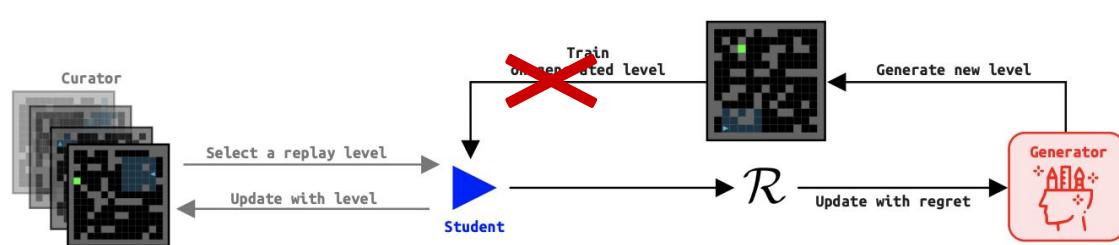
POET (Wang et al. 2019)



PAIRED (Dennis et al. 2020)



Robust PLR (Jiang et al. 2021)

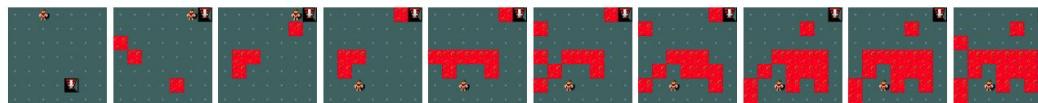
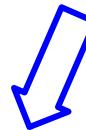


Q: *How can we harness the power of evolution in a principled regret-based curriculum?*

A: *Rather than generate levels from scratch, leverage the PLR buffer to produce new levels by editing existing ones.*

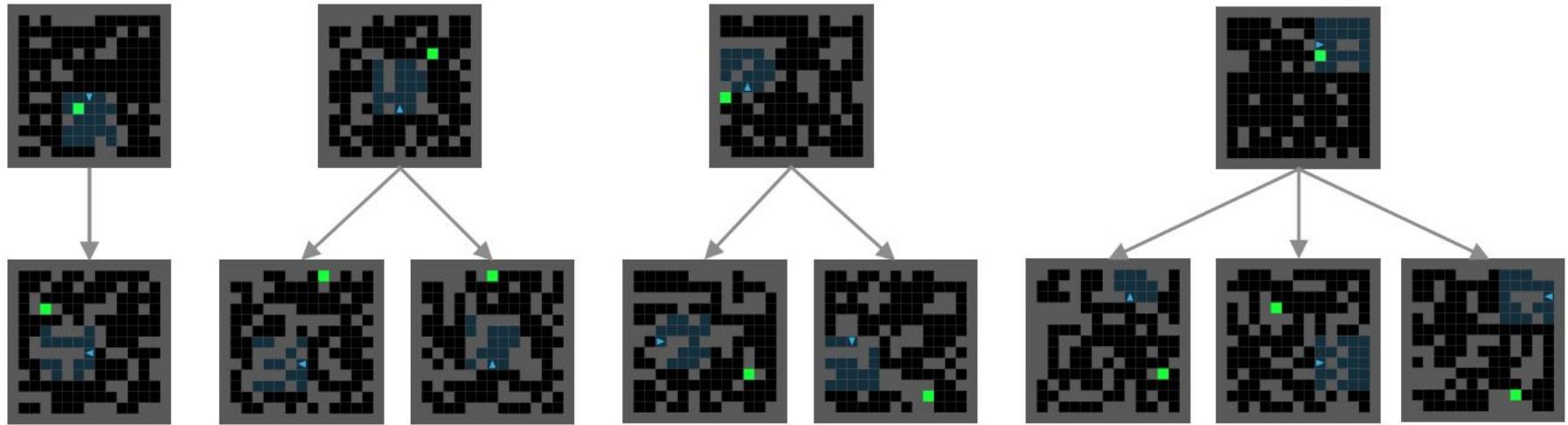
Adversarially Compounding Complexity by Editing Levels, or ACCEL

Algorithm	Generation Strategy	Generator Obj	Curation Obj	Setting
POET (Wang et al., 2019)	Evolution	Minimax	MCC	Population-Based
PAIRED (Dennis et al., 2020)	Reinforcement Learning	Minimax Regret	None	Single Agent
PLR (Jiang et al., 2021b;a)	Random	None	Minimax Regret	Single Agent
ACCEL	Random + Evolution	None	Minimax Regret	Single Agent



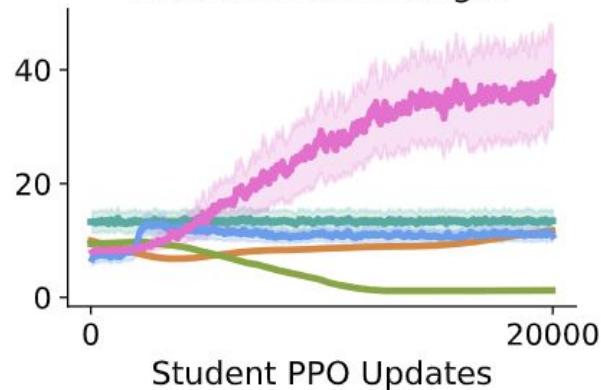
Remark 1. If ACCEL reaches a Nash equilibrium, then the student policy is following a minimax regret strategy.

So... Does it Work?

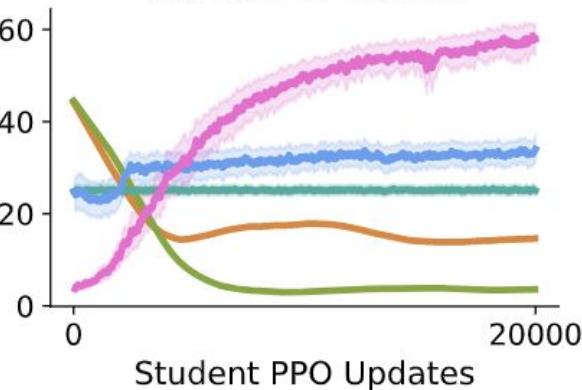


— PAIRED
 — Minimax
 — DR
 — PLR
 — ACCEL

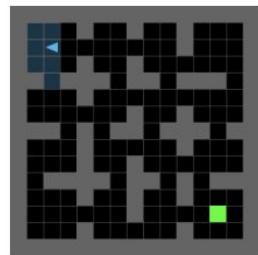
Shortest Path Length



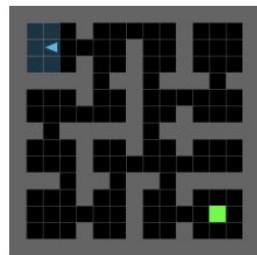
Number of Blocks



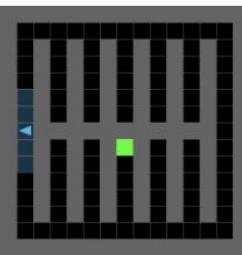
SixteenRooms



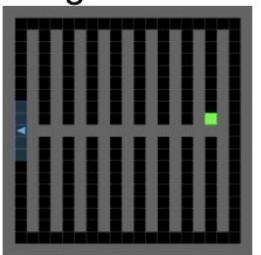
SixteenRooms2



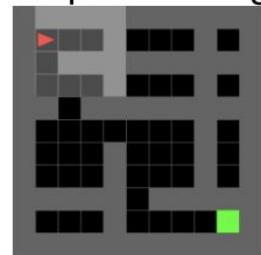
SmallCorridor*



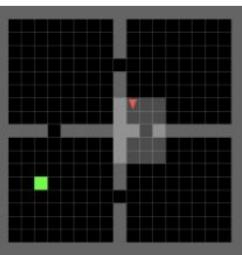
LargeCorridor*



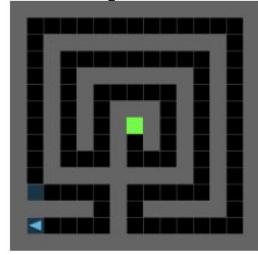
SimpleCrossing*



FourRooms*



Labyrinth



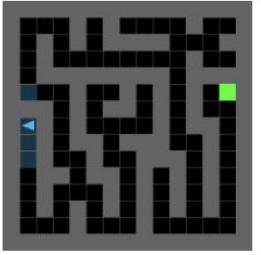
Labyrinth2



Maze



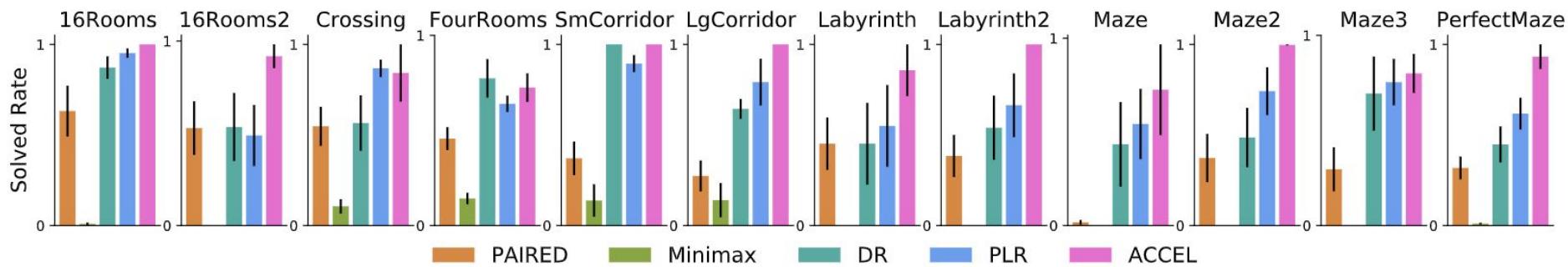
Maze2

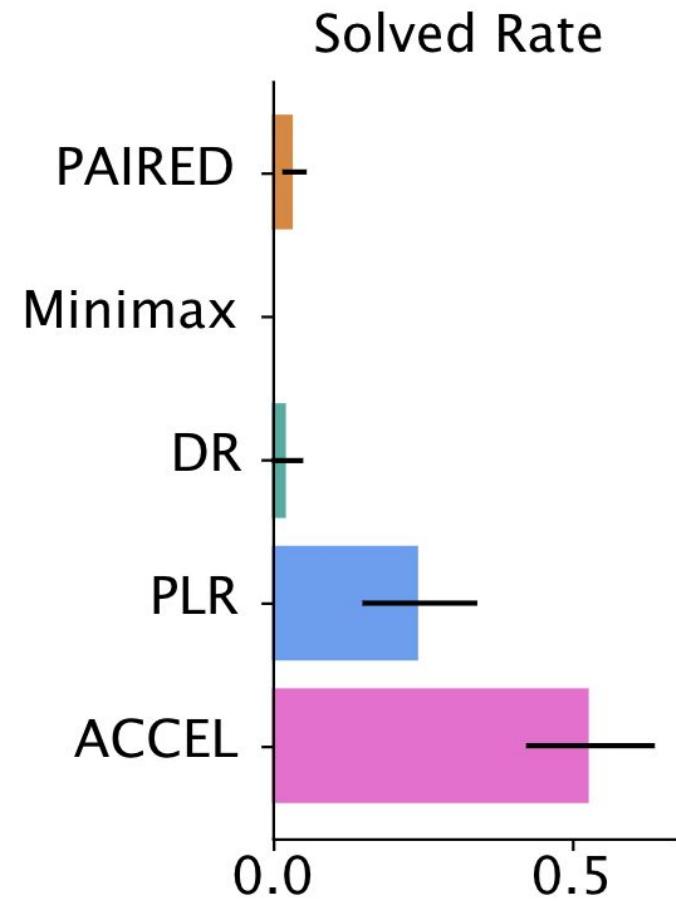


Maze3



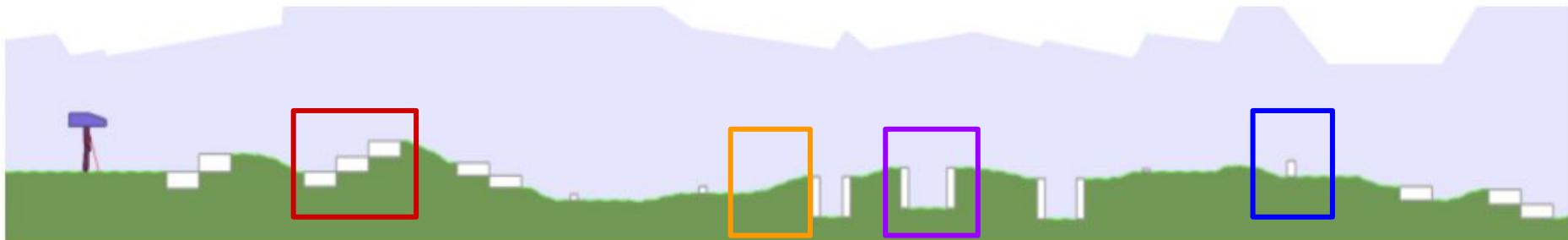
PerfectMaze*

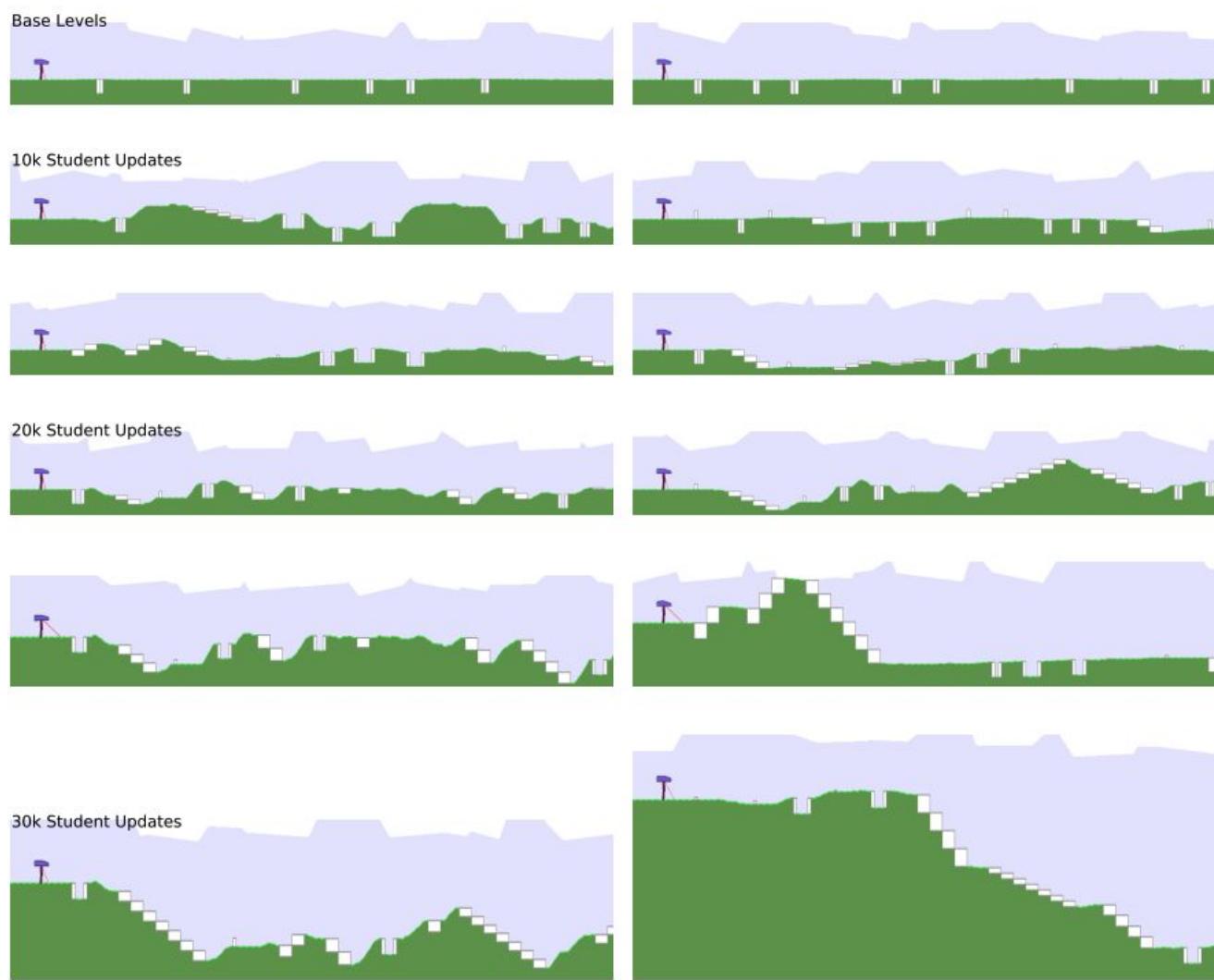




Walking in Challenging Terrain

	Stump Height	Stair Height	Stair Steps	Roughness	Pit Gap
Easy Init	[0,0.4]	[0,0.4]	1	Unif(0.6)	[0,0.8]
Edit Size	0.2	0.2	1	Unif(0.6)	0.4
Max Value	[5,5]	[5,5]	9	10	[10,10]





Stairs



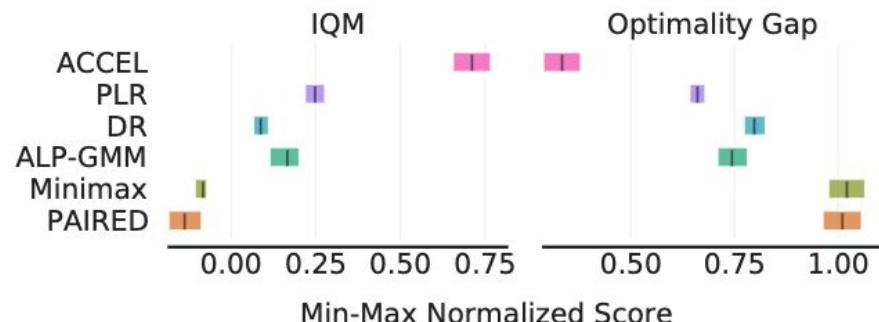
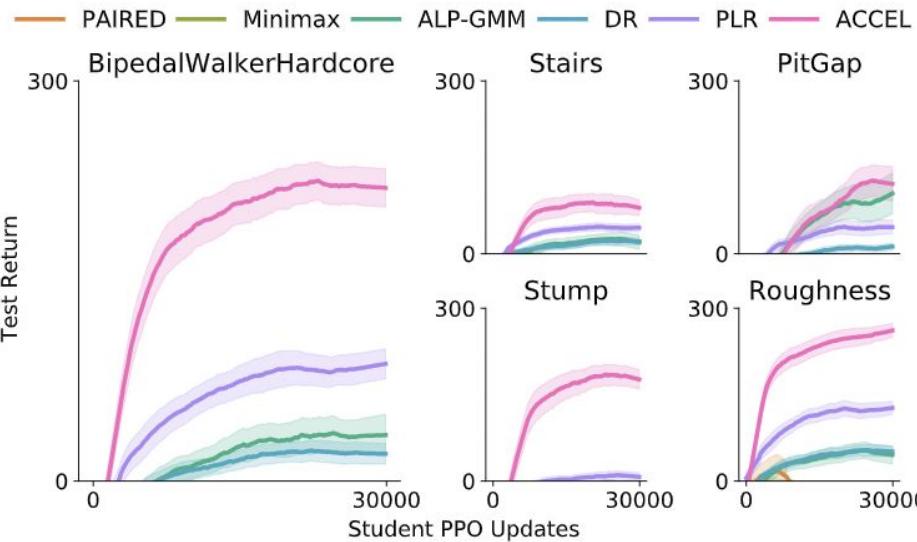
PitGap

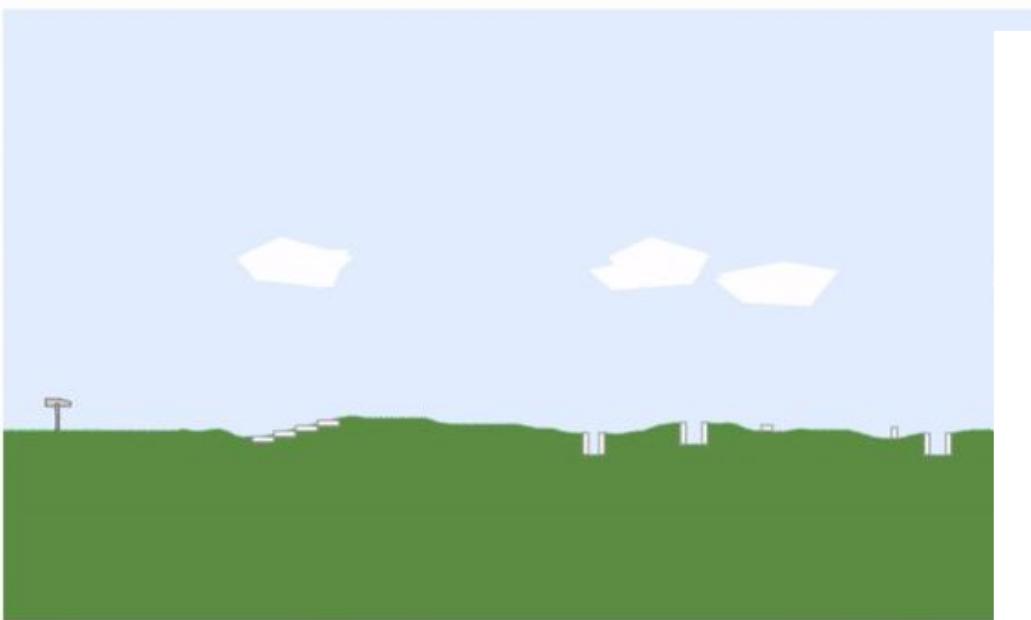


Stump

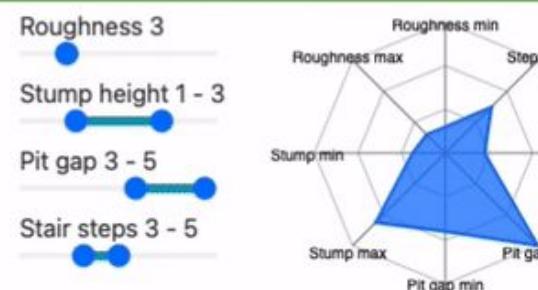


Roughness





	ACCEL	Show <input checked="" type="checkbox"/>	Follow <input checked="" type="checkbox"/>
	PLR	Show <input checked="" type="checkbox"/>	Follow <input type="checkbox"/>
	ALP-GMM	Show <input checked="" type="checkbox"/>	Follow <input type="checkbox"/>
	Domain Randomization	Show <input checked="" type="checkbox"/>	Follow <input type="checkbox"/>



Run

Reset

Show all seeds

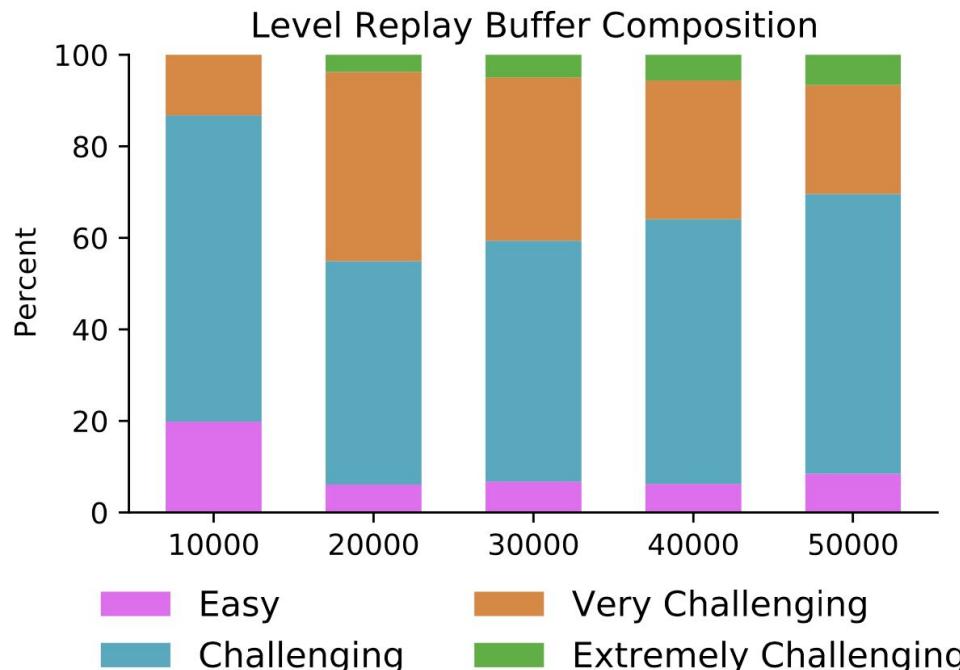
accelagent.github.io



Reaching the Complexity of POET

Table 1. Environment encoding thresholds.

Stump Height (High)	Pit Gap (High)	Ground Roughness
≥ 2.4	≥ 6	≥ 4.5



Summary

ACCEL is a new UED algorithm, capable of producing agents that can:

1. Produce and solve complex levels.
2. Transfer to challenging human-designed environments.
3. Maintain strong performance on a diverse range of challenges

Arxiv paper



Code



Twitter Thread

