Adversarial Environment Design via Regret-Guided Diffusion Models

Hojun Chung, Junseo Lee, Minsoo Kim, Dohyeong Kim, and Songhwai Oh

Robot Learning Laboratory, Seoul National University

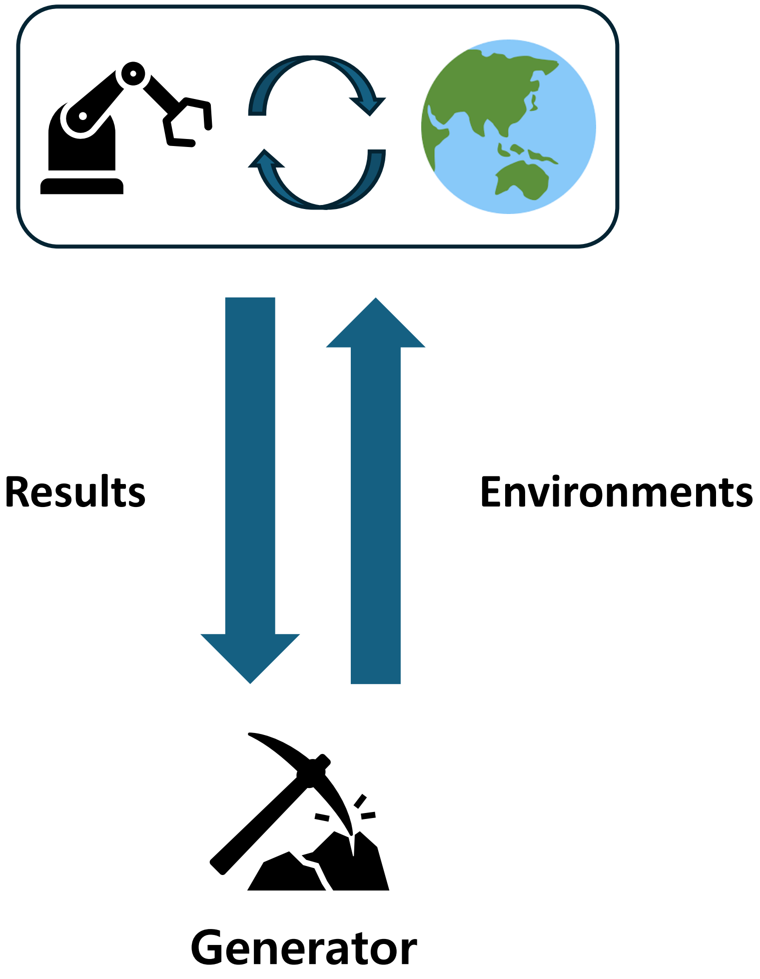
RLLAB http://rllab.snu.ac.kr

NEURAL INFORMATION PROCESSING SYSTEMS

Problem Definition

- Autonomous agents often fail in unseen environments.
- To train an agent robust to environmental changes, we focus on generating adversarial environments in which the agent will be trained.

Unsupervised Environment Design (UED)



- UED is a framework designed to find a minimax regret policy π^* that is robust to the variations in the environment θ.

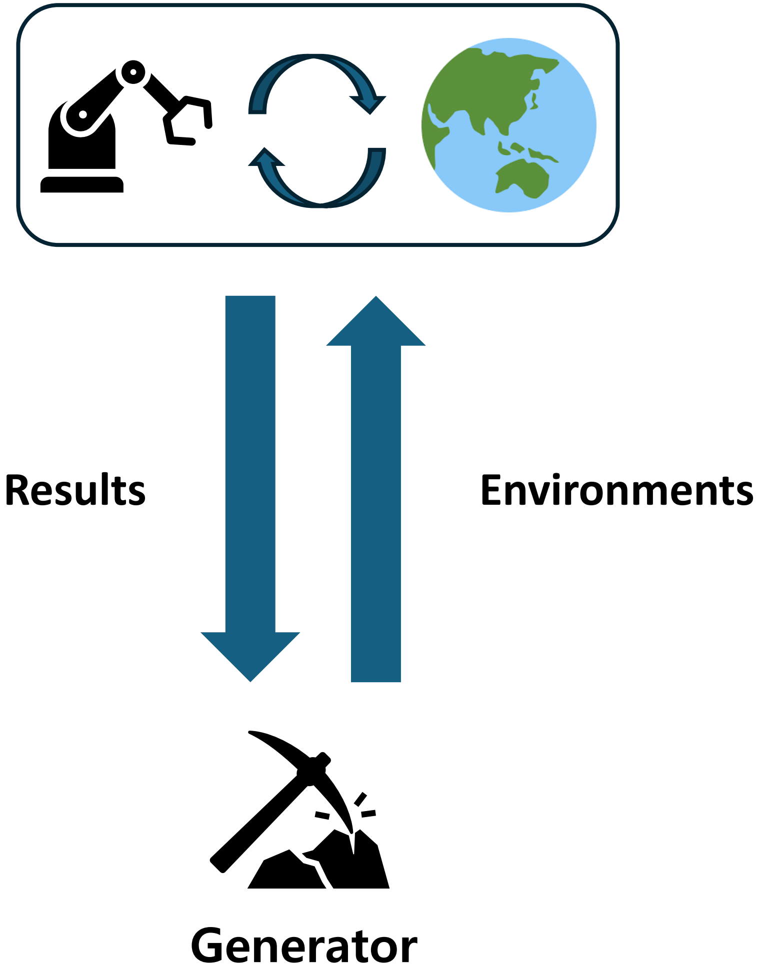
$$\pi^* \in \operatorname*{argmin}_{\pi \in \Pi} \max_{\theta \in \Theta} \textsc{Regret}(\pi, \theta)$$

$$\textsc{Regret}(\pi, \theta) := -V(\pi, \theta) + \max_{\pi' \in \Pi} V(\pi', \theta)$$

- To obtain π^*, UED solves the following min-max problem:

$$\min_{\pi \in \Pi} \max_{\theta \in \Theta} \textsc{Regret}(\pi, \theta)$$

Unsupervised Environment Design (UED)



- Prior works on UED
  - train an environment generator via reinforcement learning.
    - ➢ Stability ↓ / Sample efficiency ↑
  - replay among randomly generated environments with high regrets.
    - ➢ Stability ↑ / Sample efficiency ↓

- We propose a method which takes the advantages of two approaches by **leveraging the power of the diffusion model**.

Soft UED

- We augment the UED objective to ensure the diversity of the training environments and enhance the stability.

$$\min_{\pi \in \Pi} \max_{\Lambda \in \mathcal{D}_\Lambda} \mathbb{E}_{\theta \sim \Lambda} \left[\textsc{Regret}(\pi, \theta)\right] + \frac{1}{\omega} H(\Lambda)$$

Λ: distribution over a set of environment parameter

- The modified min-max problem has a valid optimal point.

Proposition 4.1. *Let* $L(\pi, \Lambda) := \mathbb{E}_{\theta \sim \Lambda} \left[\textsc{Regret}(\pi, \theta)\right] + \frac{1}{\omega} H(\Lambda)$ *and assume that* S, A, *and* Θ *are finite. Then,* $\min_{\pi \in \Pi} \max_{\Lambda \in \mathcal{D}_\Lambda} L(\pi, \Lambda) = \max_{\Lambda \in \mathcal{D}_\Lambda} \min_{\pi \in \Pi} L(\pi, \Lambda)$.

Regret-Guided Diffusion Models

- The soft UED converts the problem of finding regret-maximizing θ into the problem of sampling θ from the following distribution:

$$\Lambda^{\pi}(\theta) = \frac{u(\theta)\exp(\omega\textsc{Regret}(\pi,\theta))}{C^{\pi}}$$

$u(\cdot)$: uniform distribution
C^{π}: normalizing constant
ω: guidance weight

- Then, we solve this sampling problem using guided diffusion:

$$\nabla_{\theta_t}\log\Lambda_t^{\pi}(\theta_t) = \nabla_{\theta_t}\log u_t(\theta_t) + \omega\nabla_{\theta_t}\textsc{Regret}_t(\pi,\theta_t)$$

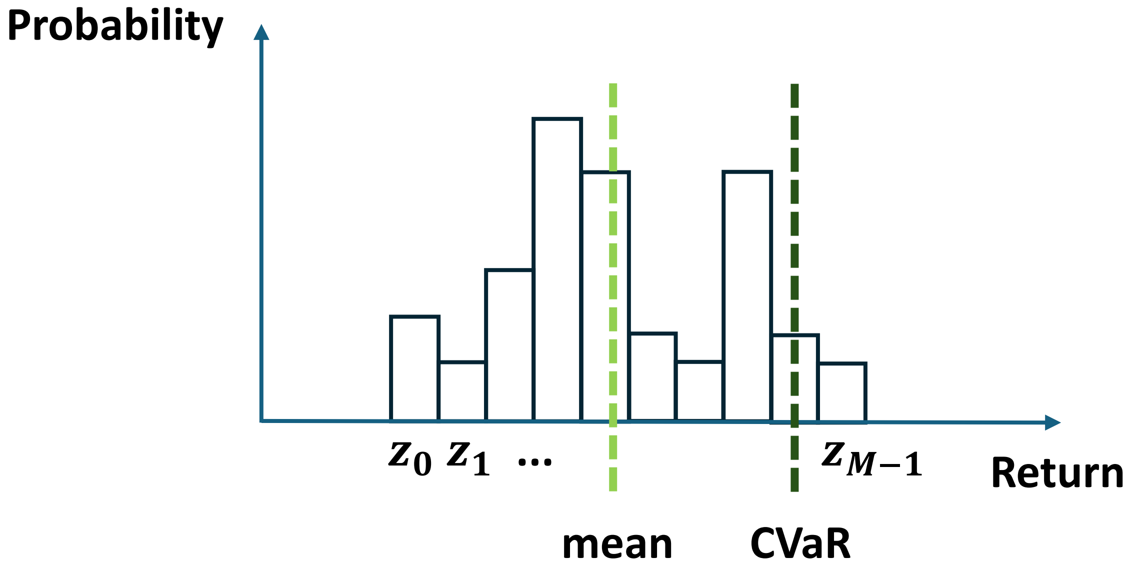
$$s_{\phi}^{\pi}(\theta_t,t) = \boxed{s_{\phi}(\theta_t,t)} + \omega\boxed{\nabla_{\theta_t}\textsc{Regret}_t(\pi,\theta_t)},$$

$$d\theta_t = -\beta_t\left[\frac{1}{2}\theta_t + s_{\phi}^{\pi}(\theta_t,t)\right]dt + \sqrt{\beta_t}dW_t.$$

- $\boxed{}$: pre-train a diffusion model, $\boxed{}$: estimate regret in a differentiable form

A Differentiable Regret Estimator

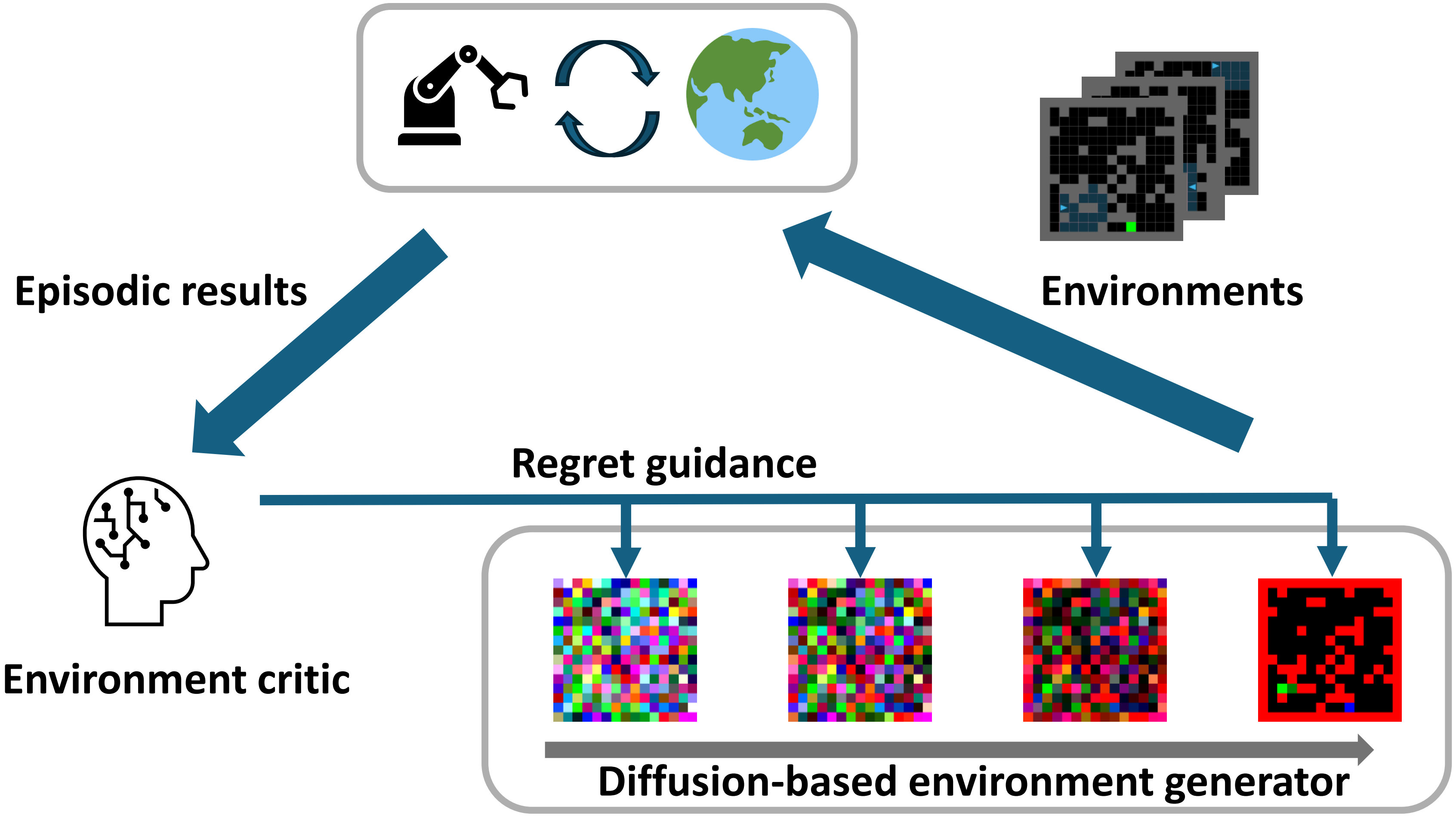
- Prior works estimate the regret in a non-differentiable form.
- We utilize an environment critic τ_ψ, which predicts a distribution of return.



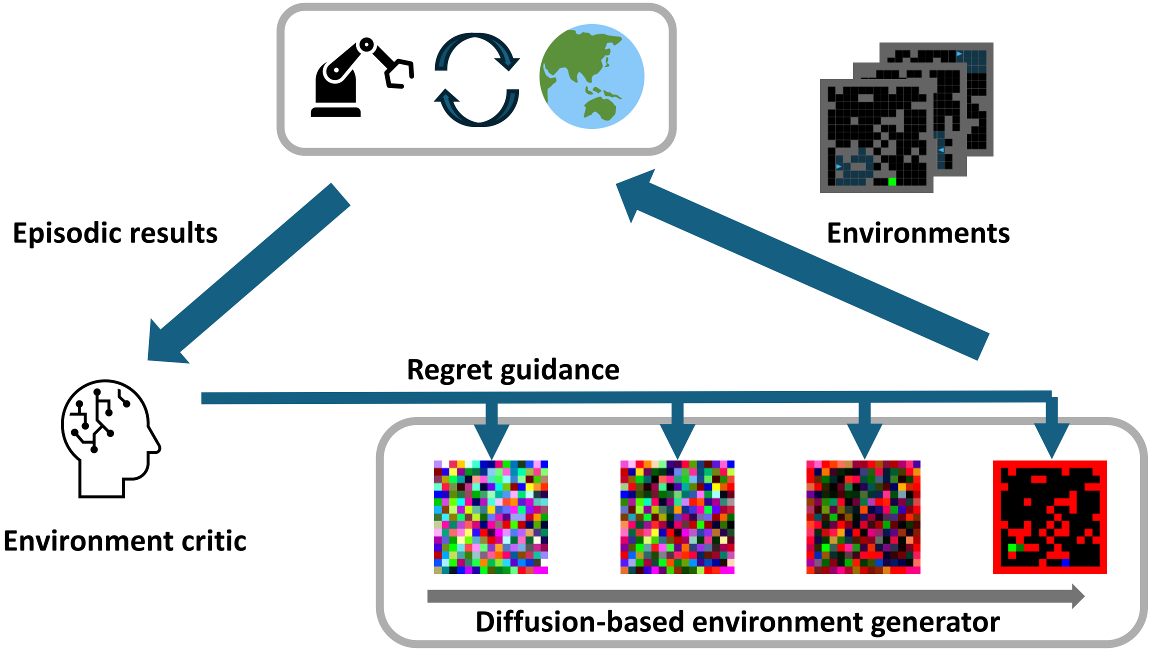
$$\hat{\mathcal{Z}}_\pi(\theta_t, t) = \sum_{i=0}^{M-1} \mathrm{softmax}_i(\tau_\psi(\theta_t, t))\delta_{z_i}$$

$$\textsc{Regret}_t(\theta_t, t) \approx \mathrm{CVaR}_\alpha(\hat{\mathcal{Z}}_\pi(\theta_t, t)) - \mathbb{E}(\hat{\mathcal{Z}}_\pi(\theta_t, t))$$

Overview: ADD



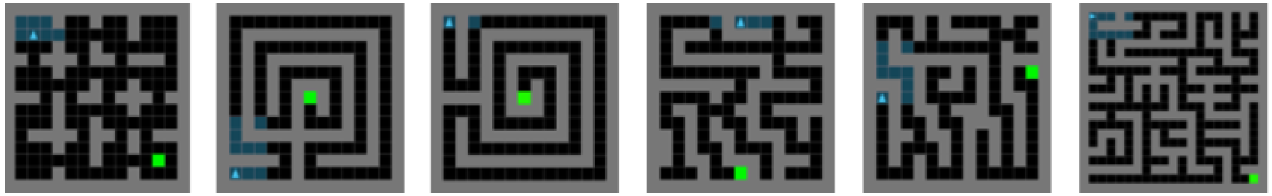
Overview: ADD



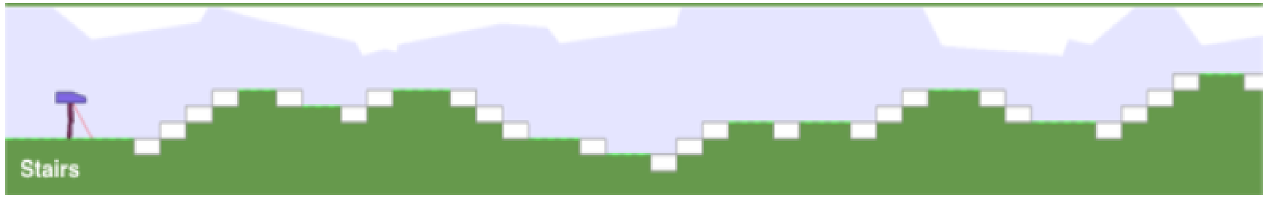
- No additional training of the generator
  - ➢ Stability ↑
- Directly generates training environments
  - ➢ Sample efficiency ↑
- Effectively combines the strengths of previous UED methods!

Experiments

- **Tasks**
  - Minigrid



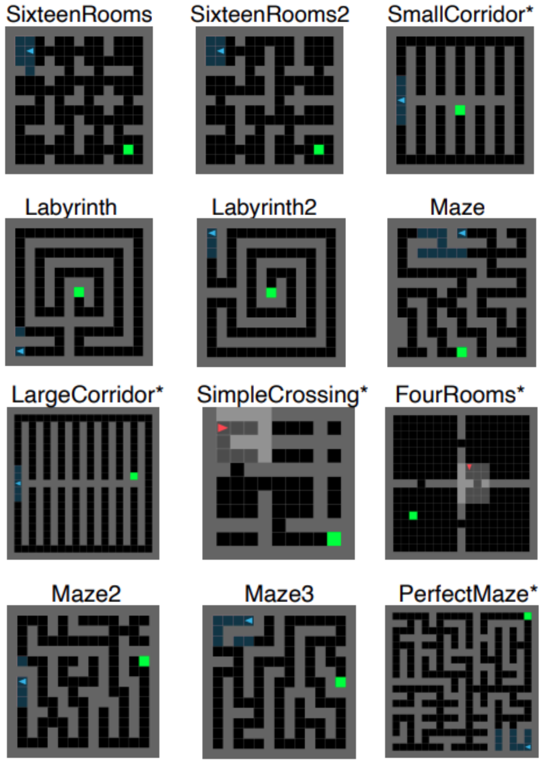
  - BipedalWalker



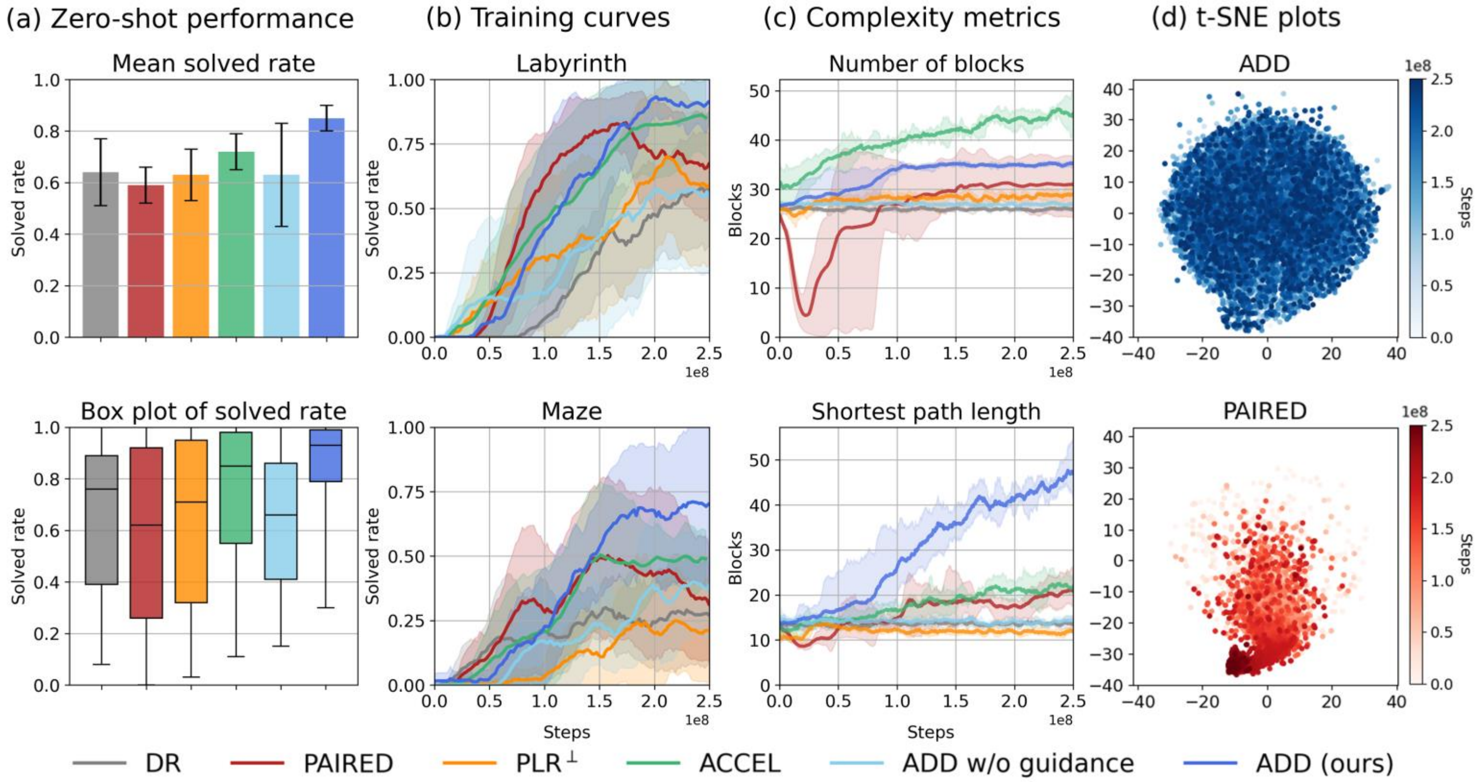
- **Evaluation**
  - Zero-shot transfer performance
  - Generated curriculum

Minigrid Results

Test environments

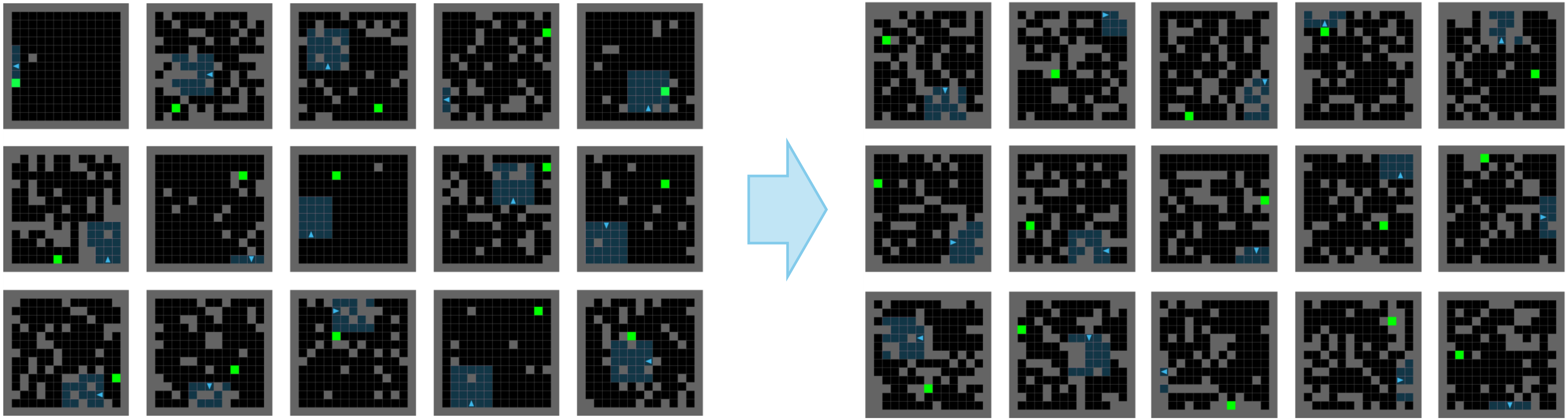


Results



Minigrid Results

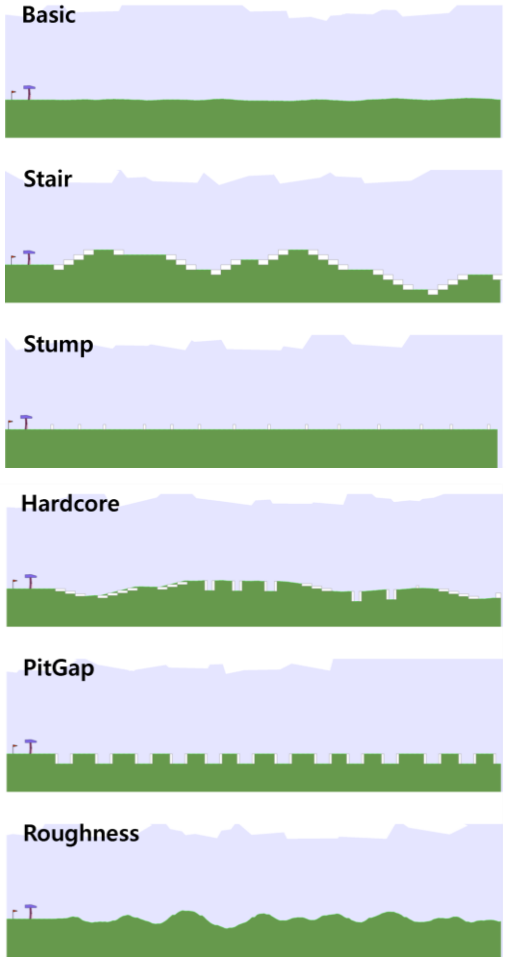
Generated training environments



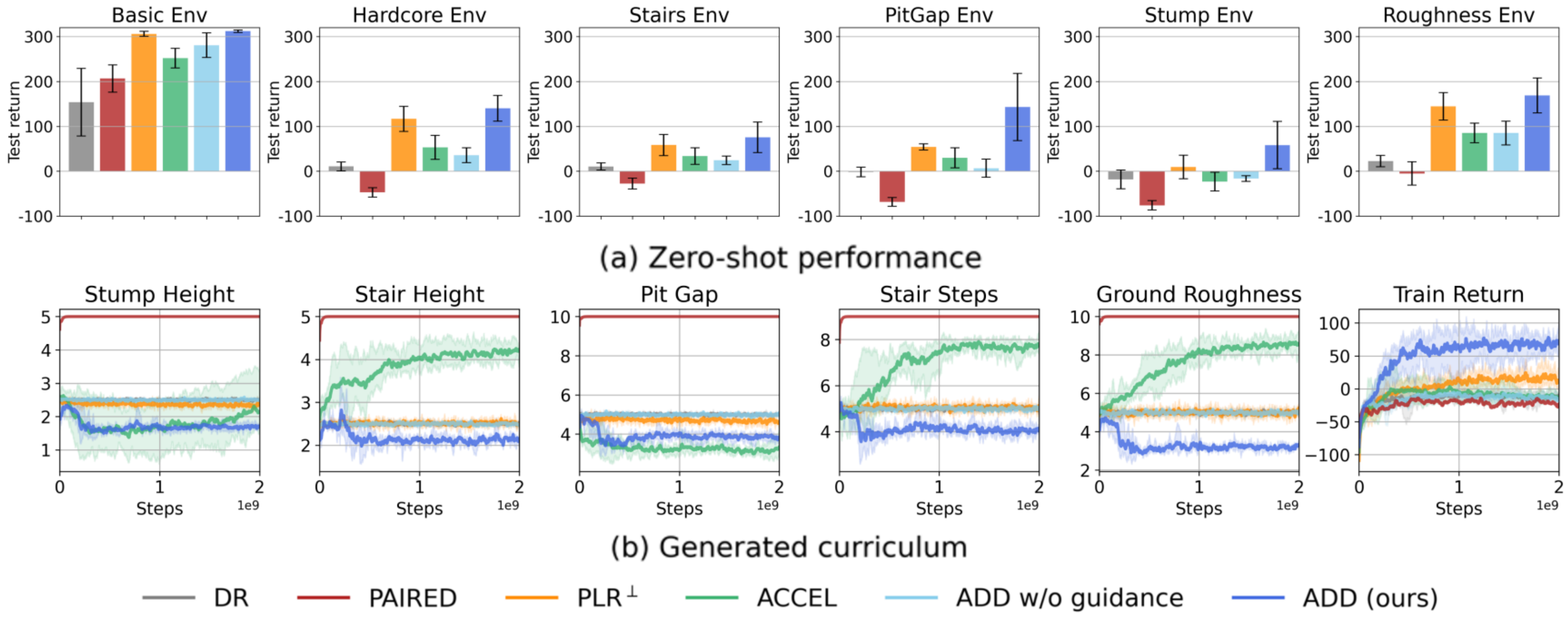
http://rllab.snu.ac.kr

BipedalWalker Results

Test environments

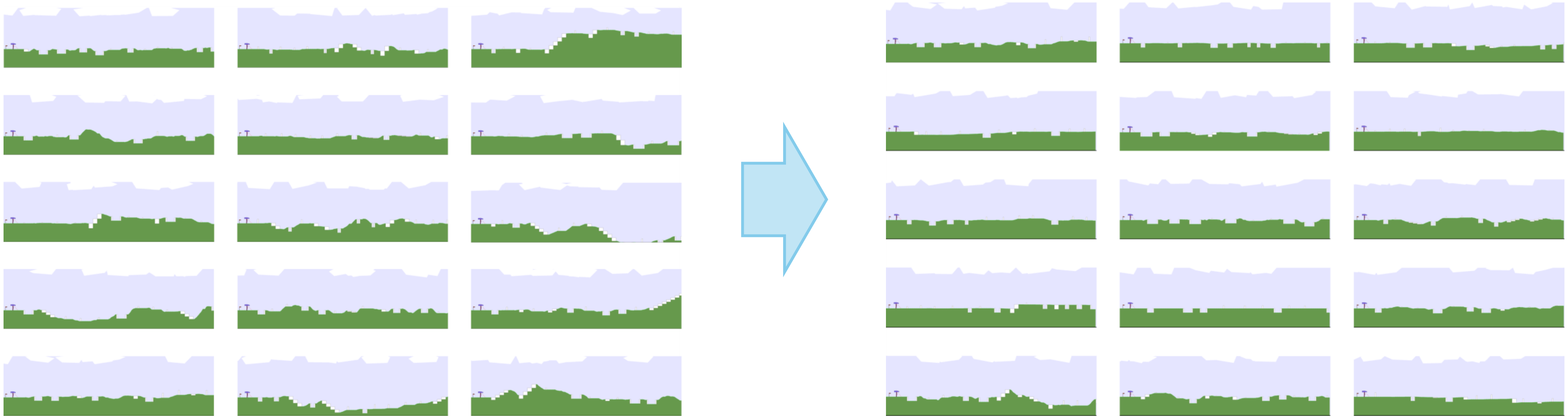


Results

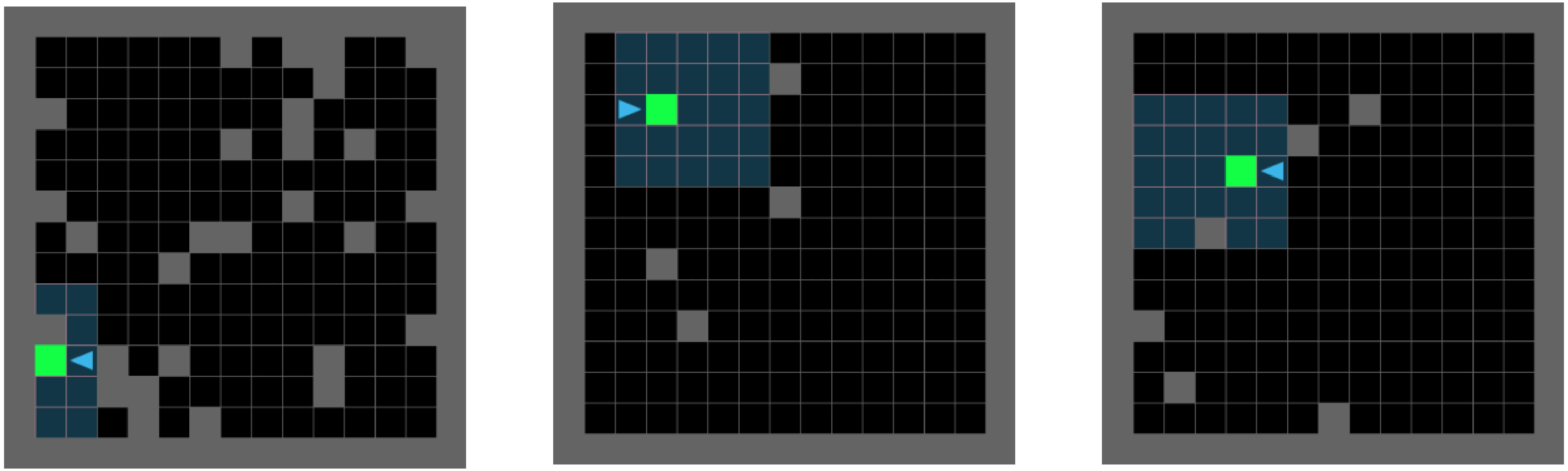


BipedalWalker Results

Generated training environments



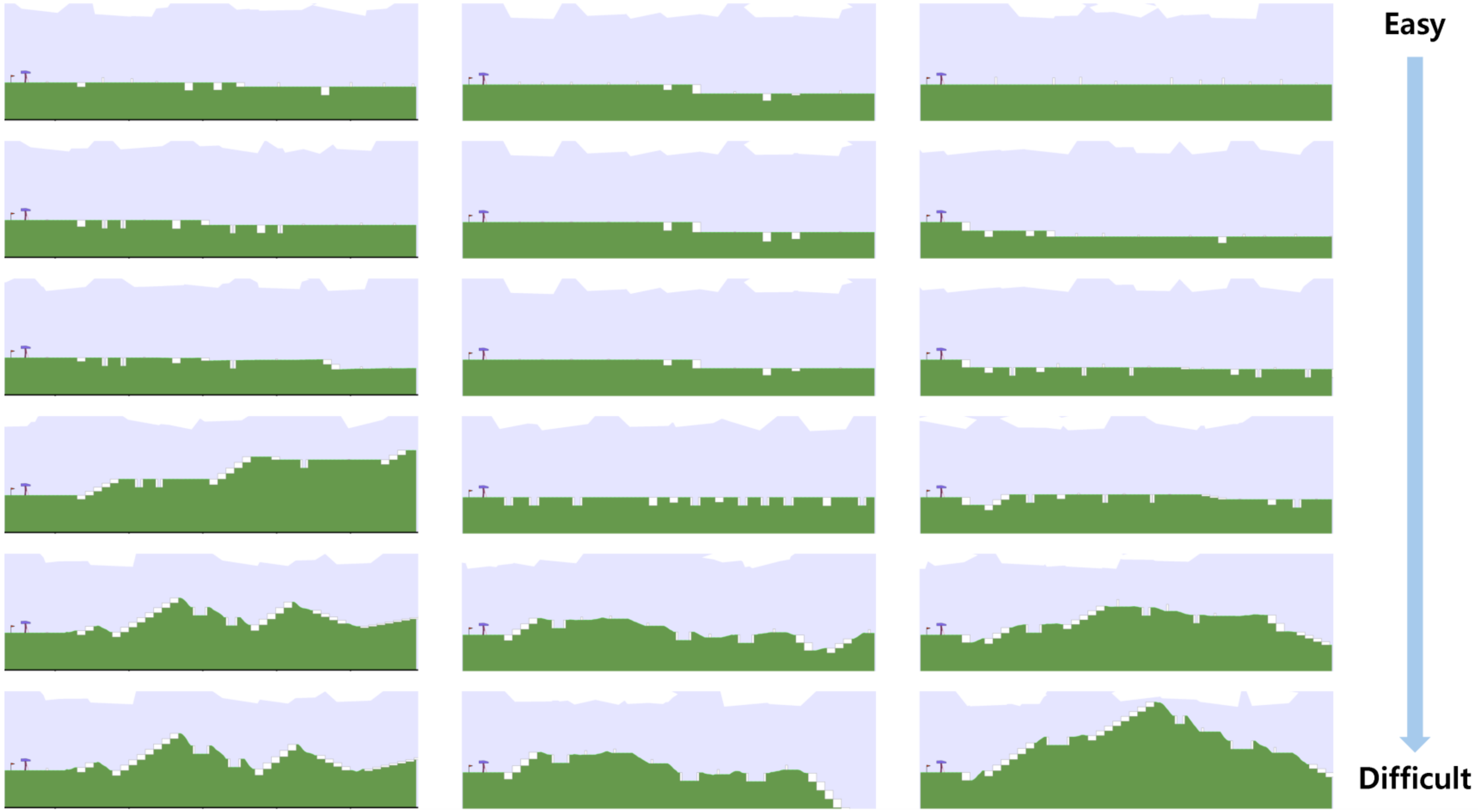
Controlling Difficulty Levels



- Additionally, our method can control the difficulty level of environments it generates

$$s'_\phi(\theta_t, t) = s_\phi(\theta_t, t) + \omega \nabla_{\theta_t} \log \Pr(\hat{\mathcal{Z}}_\pi(\theta_t, t) = z_{M-k}),$$
$$d\theta_t = -\beta_t \left[\frac{1}{2}\theta_t + s'_\phi(\theta_t, t) \right] dt + \sqrt{\beta_t} dW_t.$$

Controlling Difficulty Levels



http://rllab.snu.ac.kr

Thank you for your attention

If you have any questions, please contact hojun.chung@rllab.snu.ac.kr