Variational Automatic Curriculum Learning for Sparse-Reward Cooperative Multi-Agent Problems

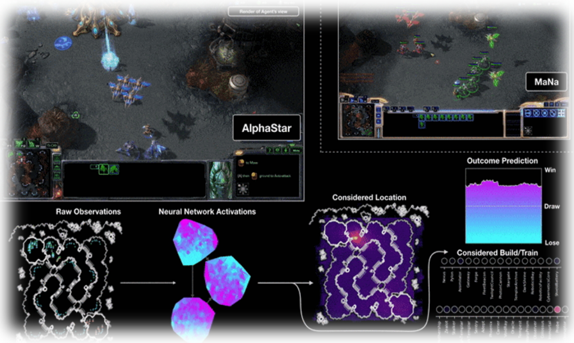
Jiayu Chen, Yuanxin Zhang, Yuanfan Xu, Huimin Ma,
Huazhong Yang, Jiaming Song, Yu Wang, Yi Wu

Contents

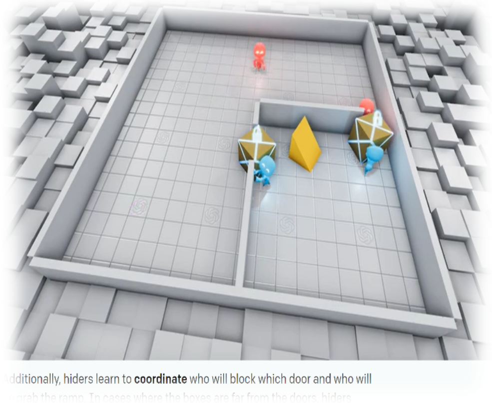
Introduction

- **Multi-agent reinforcement learning (MARL) is applied to solve challenging multi-agent games**

OpenAI Five Dota 2



AlphaStar StarCraft



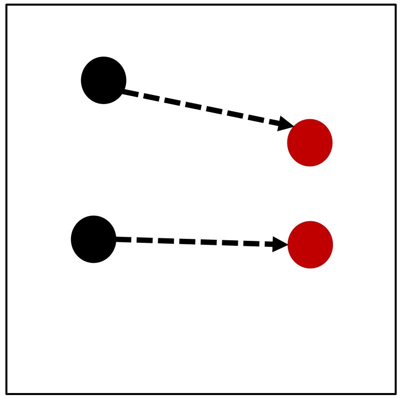
Hide-and-seek

- **learning intelligent multi-agent policies in general still remains a great RL challenge:**

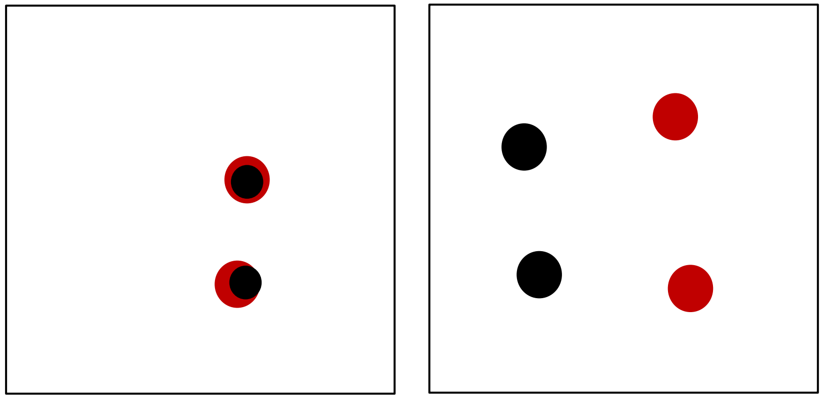
  - ✓ Massive compute → Sample-efficient
  - ✓ Require shaped rewards → Sparse-reward
  - ✓ Only handle a limit number of agents → A large number of agents

Introduction

- **We focus on goal-conditioned cooperative problems**
  - ✓ **Sparse reward problems**
  - ✓ **Massive agents**

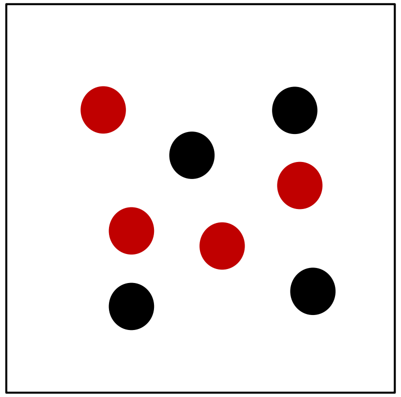


Simple-Spread with $n = 2$



reward = 1 **reward = 0**

Sparse-reward

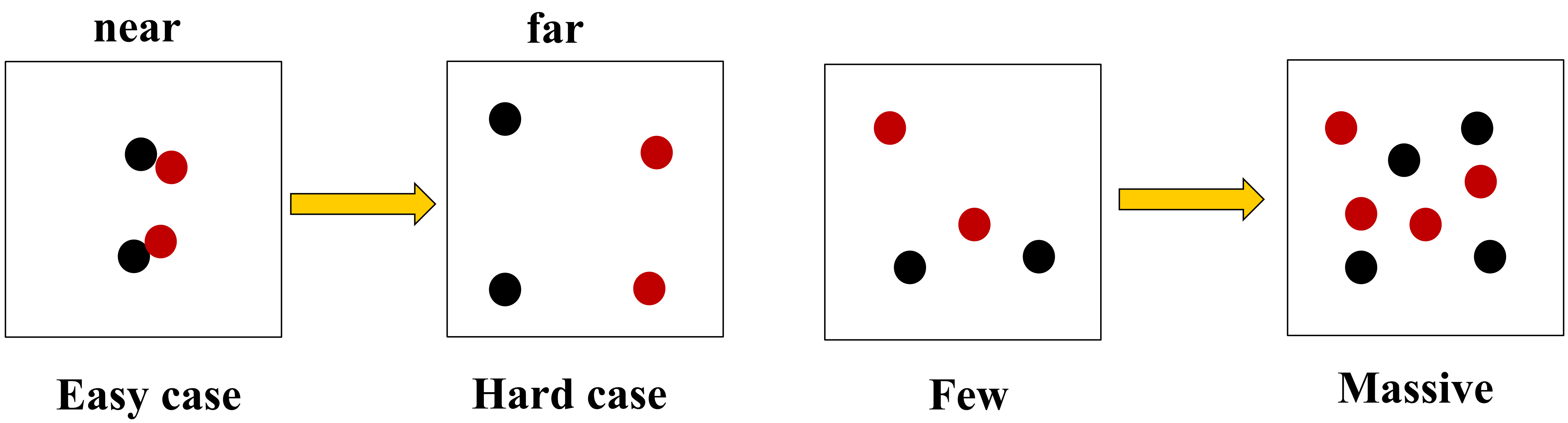


$n = 4$

More?

Introduction

- **Solution: Curriculum Learning**



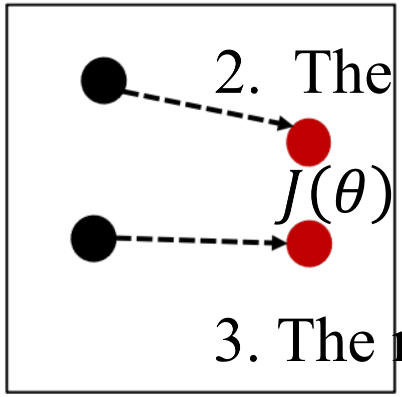
Variational Automatic Curriculum Learning

Preliminary

n: e.g. number of agents

1. Define a multi-agent Markov decision process (MDP) :

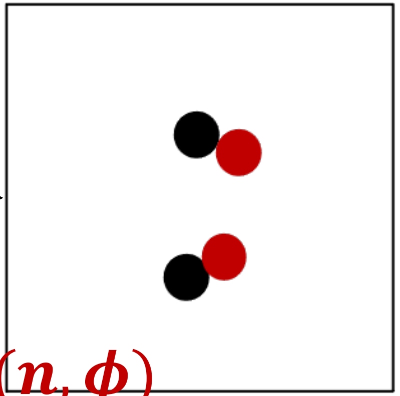
$$M(n,\phi) =< \boxed{n}, \boxed{\phi}, S, \mathcal{A}, O, O, P, R, \boxed{s_\phi^0}, g_\phi, \gamma >$$



ϕ: positions of agents and landmarks

2. The final objective is to maximize:

$$J(\theta) = E_{n,\phi,a_i^t,s^t}\left[\sum_t \gamma^t R(s^t, A^t; g_\phi)\right] = E_{n,\phi}\left[\boxed{V(n,\phi,\pi_\theta)}\right]$$



s_ϕ^0: initial states

3. The main idea of curriculum learning is to construct **a task sampler $q(n,\phi)$**

$V(n,\phi,\pi_\theta)$: the value function for π_θ over task $M(n,\phi)$, **0/1**

$q(n,\phi)$ → entire task space

$q(n,\phi)$ $\begin{cases} \text{Easy to hard} \\ \text{Few to massive} \end{cases}$

generate

$M(n,\phi)$

efficiently maximize

$J(\theta)$

Variational Automatic Curriculum Learning

➢ Variational Inference

$$\mathcal{L} = \mathbb{E}_{\phi\sim p}[V(\phi,\pi)] \overset{\text{Variational inference}}{=} \mathbb{E}_{\phi\sim q}\left[\frac{p(\phi)}{q(\phi)}V(\phi,\pi)\right] \overset{\text{identical transformation}}{=} \mathbb{E}_{\phi\sim q}\left[V(\phi,\pi) + \left(\frac{p(\phi)}{q(\phi)} - 1\right)V(\phi,\pi)\right]$$

$$\geq \underbrace{\underbrace{\mathbb{E}_{\phi\sim q}[V(\phi,\pi)]}_{\mathcal{L}_1:\, policy\ update} + \underbrace{\mathbb{E}_{\phi\sim q(\phi)}\left[V(\phi,\pi)\log\frac{p(\phi)}{q(\phi)}\right]^{*}}_{\mathcal{L}_2:\, curriculum\ update}}_{\text{iterative}}$$

$x - 1 \geq logx$

Maximize $\mathcal{L}_1$ with π

Maximize $\mathcal{L}_2$ with $q(\phi)$

*We prove that if we can perfectly optimize the RL procedure for $\mathcal{L}_1$ under $q(\phi)$, $\mathcal{L}_2$ encourages $q(\phi)$ to converge to $p(\phi)$

Variational Automatic Curriculum Learning

L_1: $\mathbb{E}_{\phi\sim q(\phi)}[V(\phi,\pi)]$, **standard RL procedure**

L_2: $\mathbb{E}_{\phi\sim q(\phi)}[V(\phi,\pi)\log(\frac{p(\phi)}{q(\phi)})]$ **How to represent $q(\phi)$?**

Neural network ? →

- **Stein variational inference**

expensive

Use particles to approximate $q(\phi)$

Q : the particle set

Variational Automatic Curriculum Learning

L_2: $\mathbb{E}_{\phi \sim q(\phi)}[V(\phi, \pi)\log(\frac{p(\phi)}{q(\phi)})]$ **How to update $q(\phi)$?**

- **Stein variational gradient descent**

$$\phi' = \phi + \epsilon f(\phi)$$

We prove that $f^*(\cdot) = E_{\phi' \in Q}[V(\phi', \pi) \cdot \nabla_{\phi'} k(\phi', \cdot)]$

Respelling force

Scale

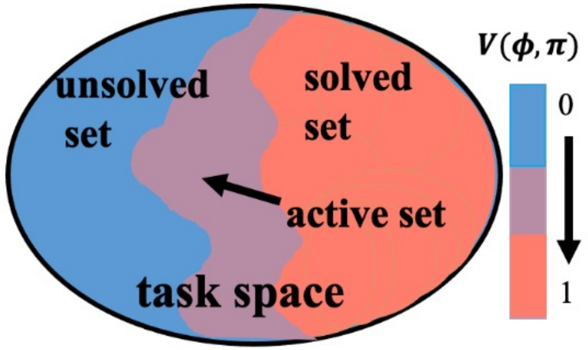
Kernel function

Task Expansion

➢ Implementation

✓ Value Quantization

$$V(\phi,\pi) \xrightarrow{\text{Efficient}} \begin{aligned} \mathcal{Q}_{sol} &= \{\phi \mid V(\phi,\pi) > \sigma_{max}\} \\ \mathcal{Q}_{act} &= \{\phi \mid \sigma_{min} \le V(\phi,\pi) \le \sigma_{max}\} \end{aligned}$$

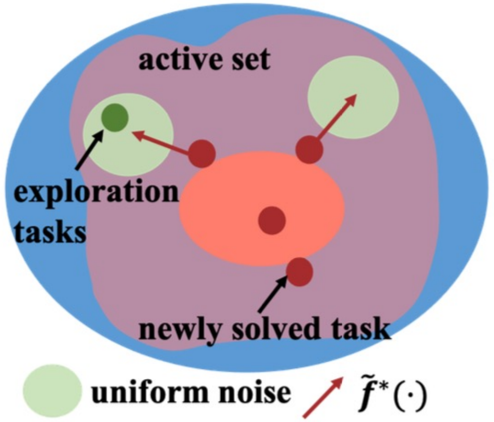


✓ Sampling-Based Particle Exploration

$$f^*(\cdot) = E_{\phi' \in Q}[V(\phi',\pi) \cdot \nabla_{\phi'} k(\phi',\cdot)]$$

$$\Big\downarrow \ \text{Simplify}$$

$$\tilde{f}^*(\cdot) \propto \mathbb{E}_{\phi' \in \mathcal{Q}_{sol}}[\nabla_{\phi'} k(\phi',\cdot)]$$



$$\boldsymbol{\phi_{exp} \leftarrow \phi_{seed} + \epsilon \tilde{f}^*(\phi_{seed}) + Unif(-\delta,\delta)}$$

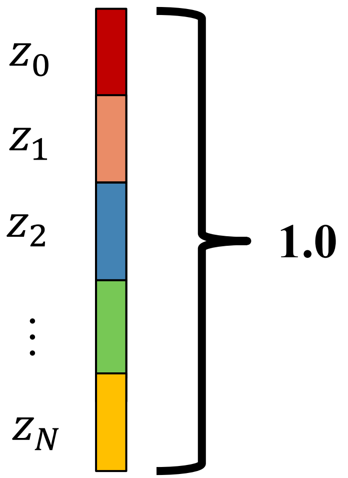
Explore novel tasks in the **boundary region** between $\mathcal{Q}_{act}$ and $\mathcal{Q}_{sol}$

Entity Progression

$$L_2: \mathbb{E}_{\phi \sim q(\phi)}[V(\phi, \pi)\log(\frac{p(\phi)}{q(\phi)})]$$

How to represent $q(\phi)$

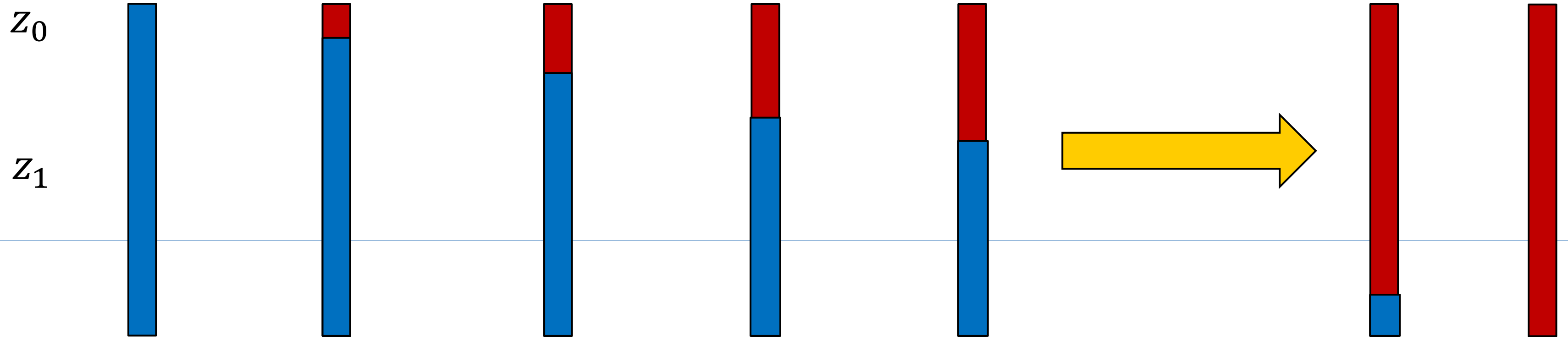
How to update $q(\phi)$



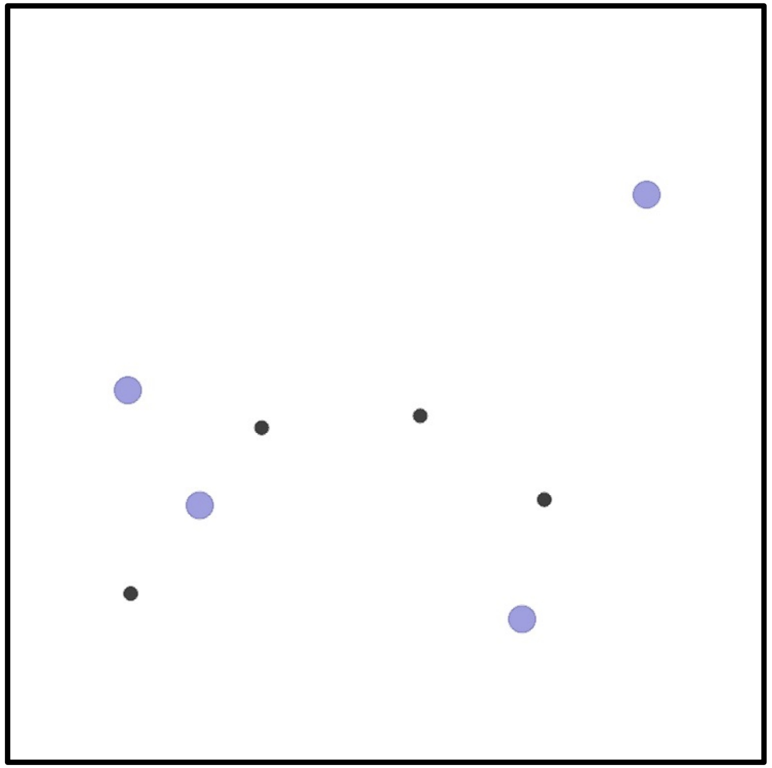
How to handle discrete variables ?

✓ Continuous Relaxation for Discrete Parameter

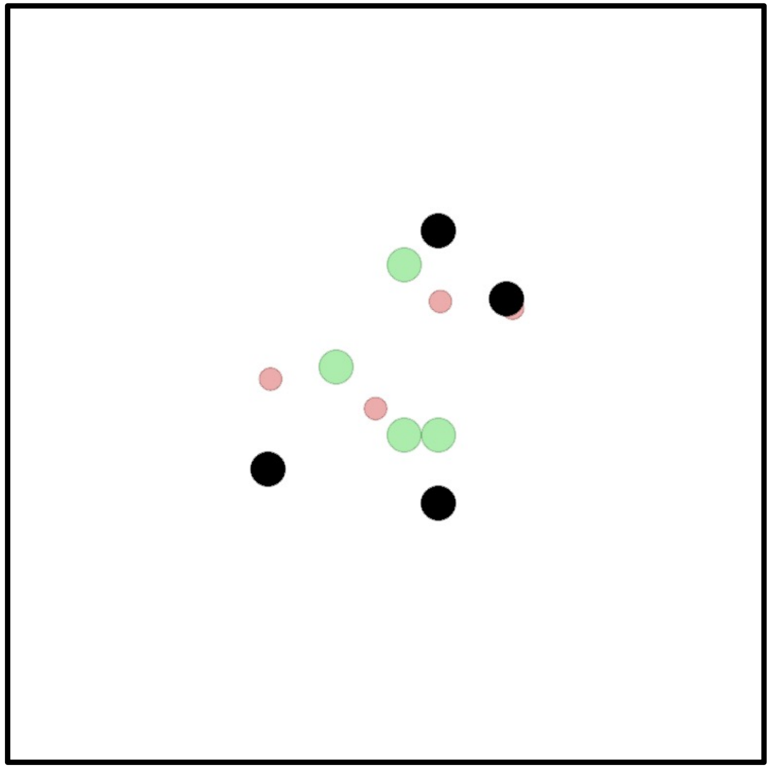
- $p(n; z) =$ Categorical $(z_1, z_2, \ldots, z_N)$ denotes the distribution which generates n agents with probability z_n
- start with $z_{n_0} = 1$ and gradually increase z_k for larger k



Multi-agent Particle-World Environments



Simple-Ball



Push-Ball

Multi-agent Particle-World Environments

Baselines :

(1) multi-agent PPO with uniform task sampling **(Uniform)**

(2) naïve population curriculum **(PC-Unif)**

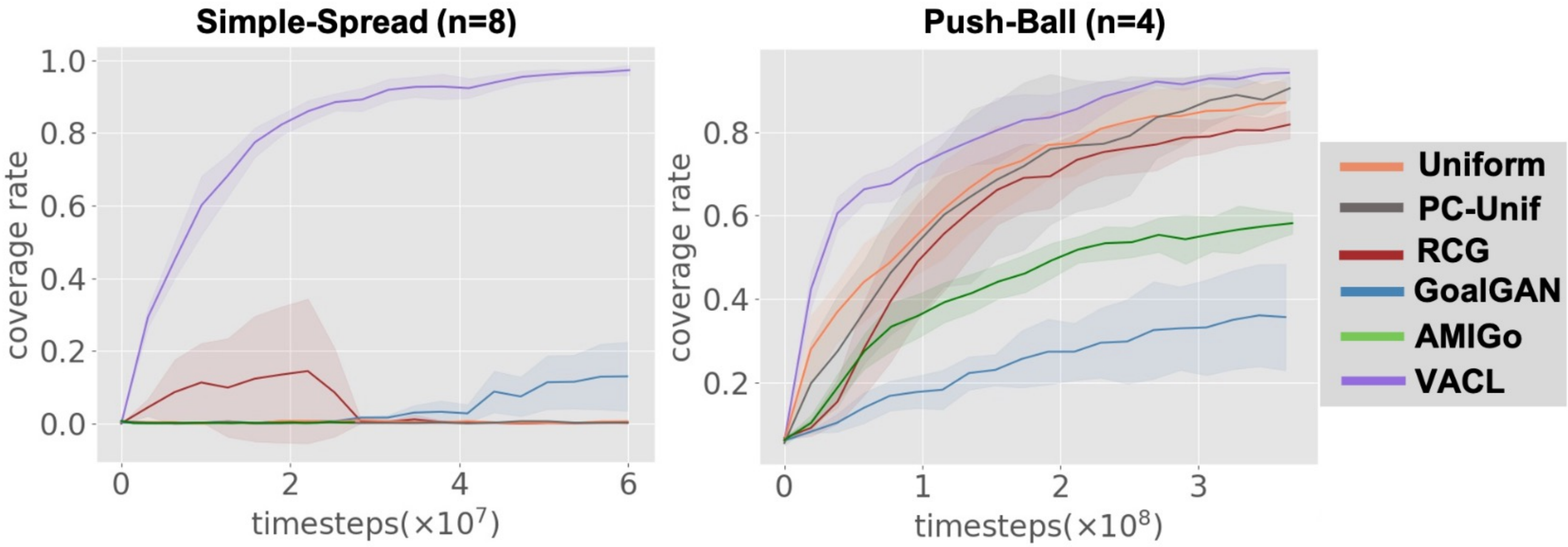
(3) reverse curriculum generation **(RCG)**

(4) automatic goal generation **(GoalGAN)**

(5) adversarially motivated intrinsic goals **(AMIGo)**

Multi-agent Particle-World Environments

✓ **Main results**



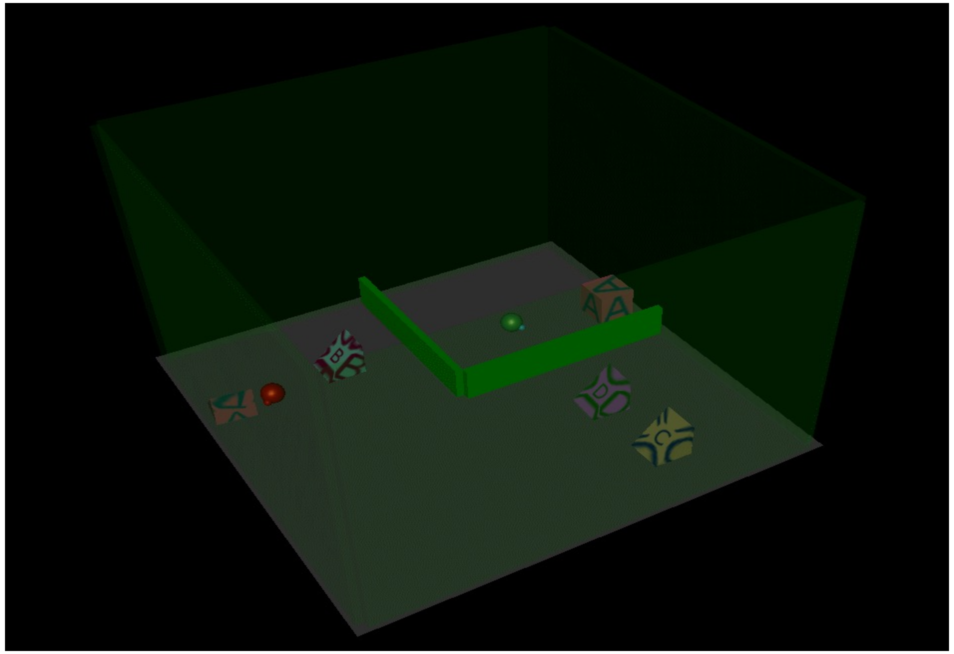
Multi-agent Particle-World Environments

✓ **The results of massive agents**

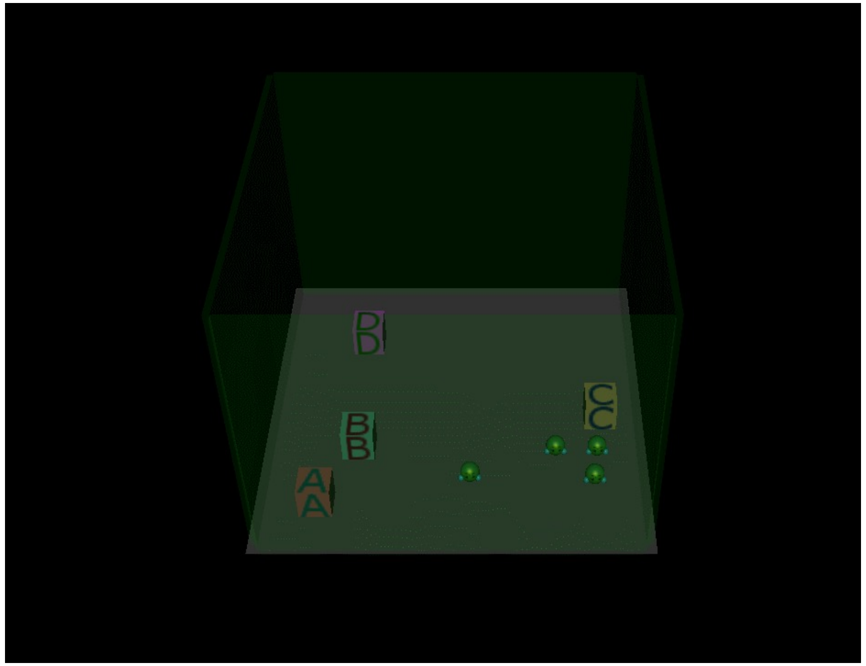
Table 1: The best coverage rate ever reported on *Simple-Spread*.

n	EPC	ATOC	VACL
24	56.8%	/	**97.6%**
50	/	92%	**98.5%**
100	/	89%	**98%**

The Hide-and-Seek Environment



Ramp-Use



Lock-and-Return

The Hide-and-Seek Environment

✓ Main results

Table 2: Results of VACL and baselines in HnS tasks.

		Uniform	RCG	GoalGAN	AMIGo	VACL
Ramp-Use	$n = 1$	$42.8\% \pm 35.4\%$	$31.5\% \pm 33.7\%$	$1.0\% \pm 0.8\%$	$47.2\% \pm 10.3\%$	$93.3\% \pm 5.4\%$
Lock-and-Return	$n = (2, 2)$	$<1\%$	$5.0\% \pm 5.1\%$	$<1\%$	$< 2\%$	$97.3\% \pm 0.1\%$
	$n = (4, 4)$	/	/	/	/	$97.0\% \pm 1.6\%$

Conclusion

- **Variational Automatic Curriculum Learning (VACL)**
  - efficiently solves a collection of **sparse-reward** multi-agent **cooperative** problems
  - achieves **over 98% coverage rate with 100 agents** in the simple-spread testbed **using sparse rewards**
  - achieves over 90% success rates on both two games in the HnS scenarios, including **reproducing the ramp use behavior**.

Thanks！

Visit our website for more information
https://sites.google.com/view/vacl-neurips-2021

Jiayu Chen
jiayu-ch19@mails.tsinghua.edu.cn