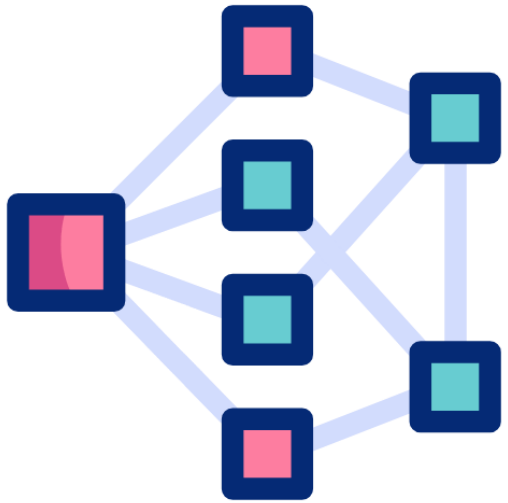
Diffusion-based Curriculum Reinforcement Learning

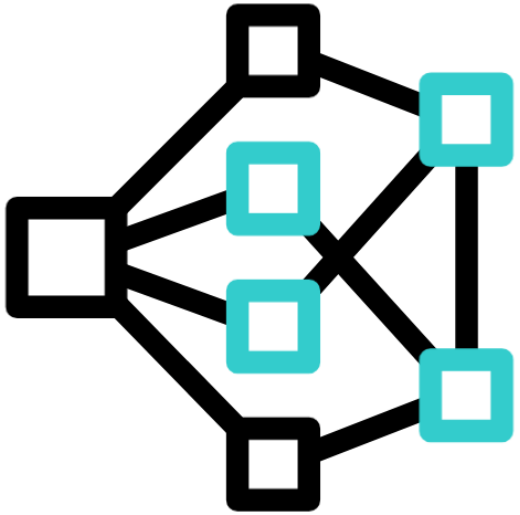
Erdi Sayar[1] Giovanni Iacca[2] Ozgur S. Oguz[3] Alois Knoll[1]

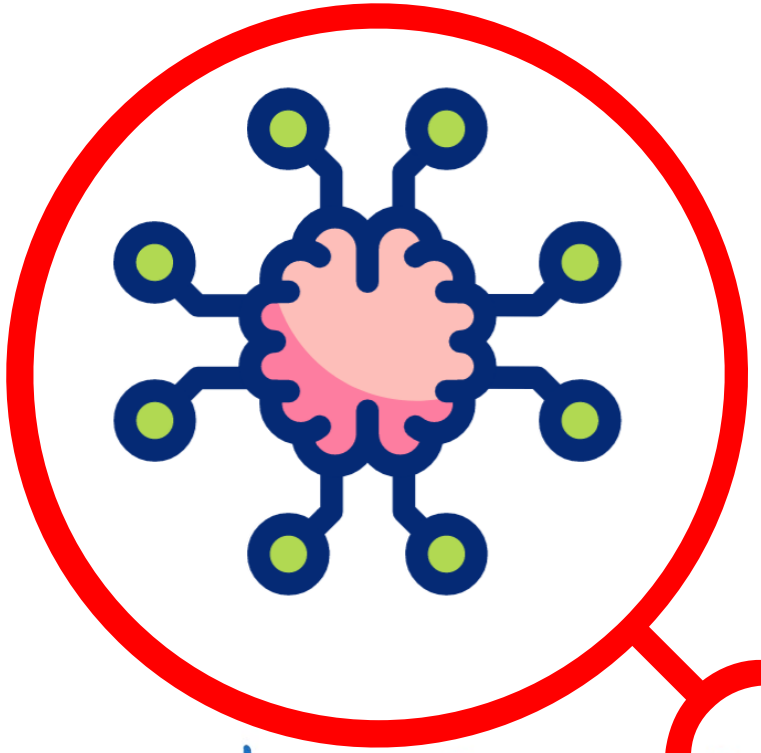
[1]Technical University of Munich, Germany
[2]University of Trento, Italy
[3]Bilkent University, Türkiye

NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS 2024

Curriculum generation with Diffusion Models







Soft Actor Critic

SAC

AIM Reward

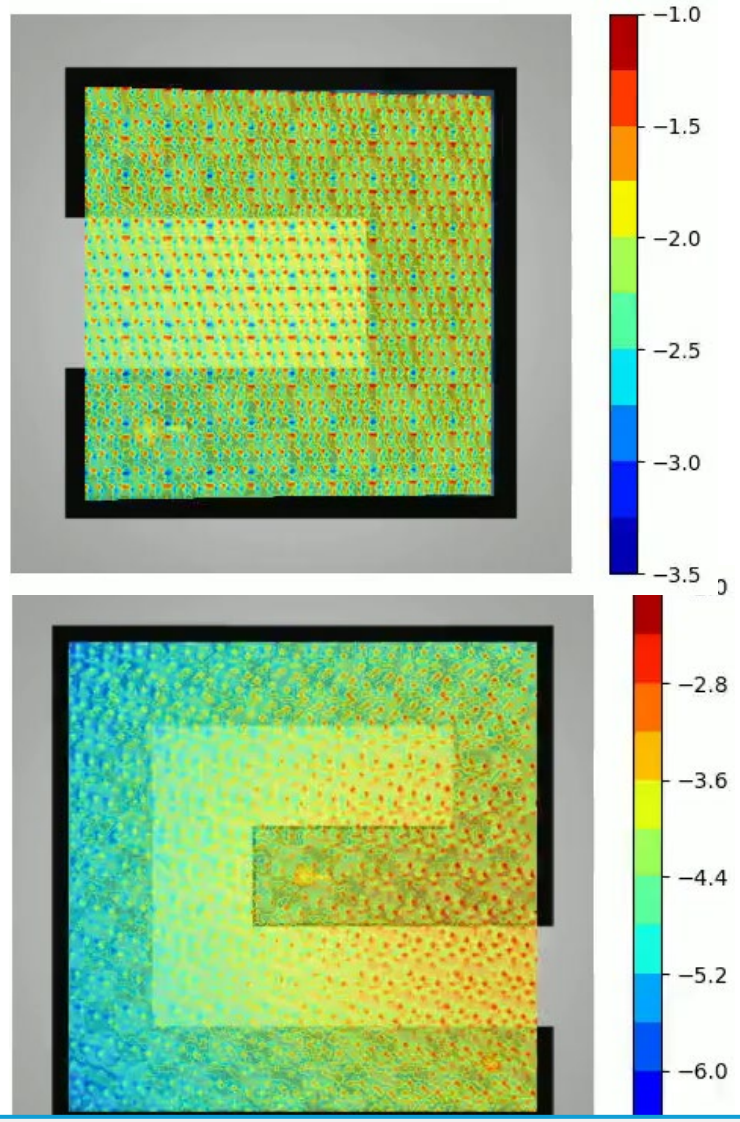
Trainable reward

Curriculum Generation

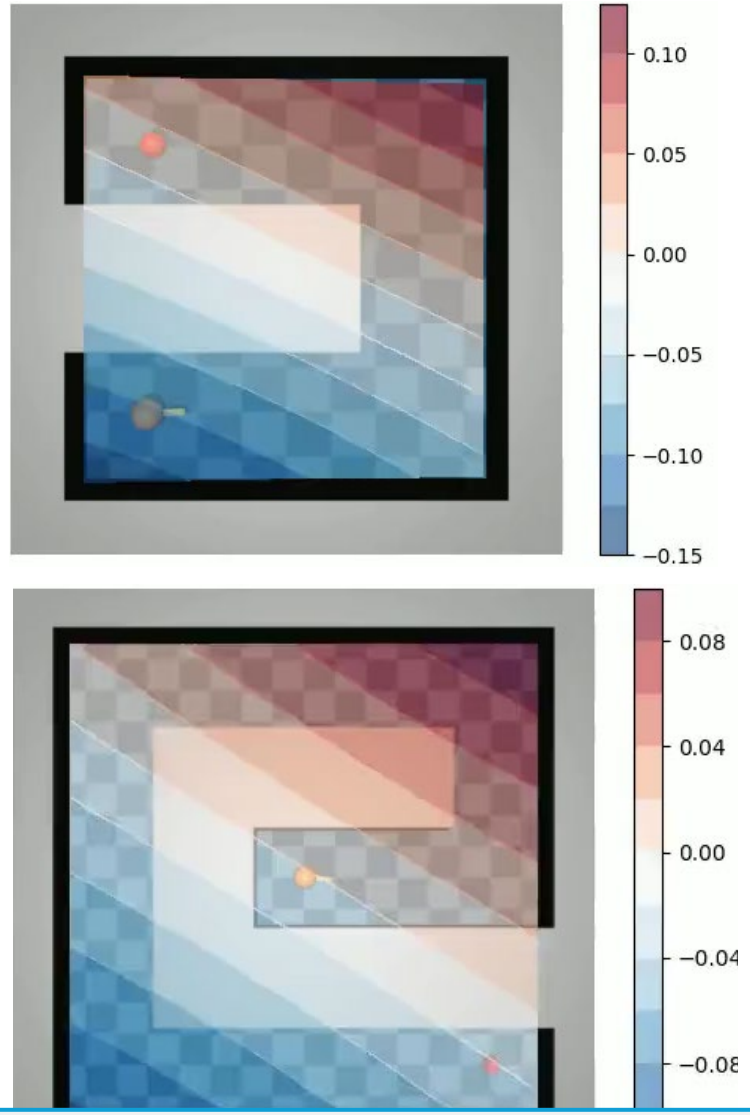
Diffusion model

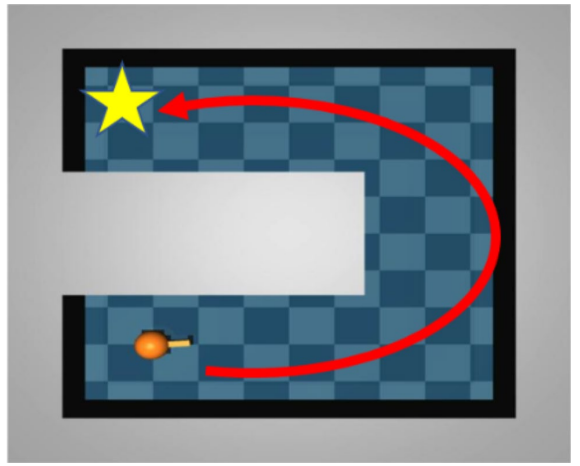
How Q and reward function changes during training

$Q_\phi(s, x, a)$

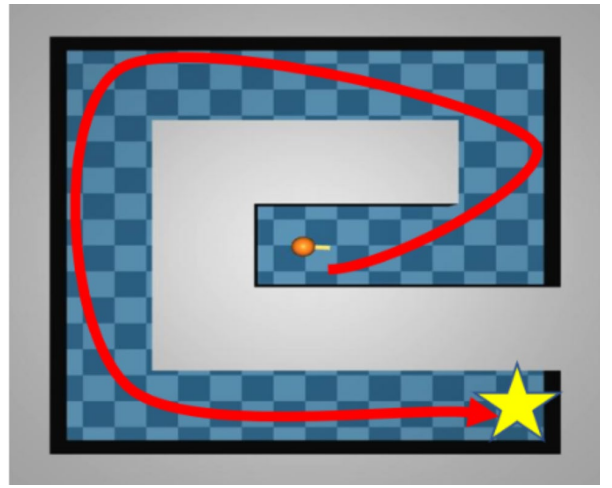


$r_\varphi(x, g_d)$





U-Maze Environment



Spiral-Maze Environment

Methodology

- We used a diffusion model to generate curriculum for the agent.

Algorithm 2 RL Training and Evaluation

Input: no. of episodes E, timesteps T
Select an off-policy algorithm $\mathbb{A}$ ▷ In our case, $\mathbb{A}$ is SAC
Initialize replay buffer $\mathcal{B} \leftarrow \emptyset$, $g_c \leftarrow \{g_d\}$ and networks Q_ϕ, π_ψ, r_φ
for episode $= 0 \ldots E$ **do**
  Sample initial state s_0
  for $t = 0 \ldots T$ **do**
    $a_t = \pi(s_t, g_c)$
    Execute a_t, obtain next state s_{t+1}
    Store transition $(s_t, a_t, r_t, s_{t+1}, g_c)$ in $\mathcal{B}$
  Sample a minibatch b from replay buffer $\mathcal{B}$
  $\mathcal{G}_c \leftarrow$ `DiffusionCurriculumGenerator`(b)
  Find g_c that maximizes w in Eq. (12)
  Update Q and π with b to minimize $\mathcal{L}_Q$ and $\mathcal{L}_\pi$ in Eq. (1) and in Eq. (2)
  Update the AIM reward function r_φ
$success \leftarrow 0$ ▷ Success rate
Sample a desired goal $g_d \sim \mathcal{G}$
for i $= 1 \ldots n_{testrollout}$ **do**
  $a_t = \pi(s_t, g_d)$
  Execute a_t, obtain next state s_{t+1} and reward r_t
  if $|\phi(s_{t+1}) - g_d| \leq \kappa$ **then**
    $success =\leftarrow success + 1/n_{testrollout}$

Methodology

- We used a diffusion model to generate curriculum for the agent.

Algorithm 1 Diffusion Curriculum Goal Generator

Input: state s, no. of reverse diffusion timestep N, training step M
Obtain states s from the minibatch b
for i $= 1, \ldots, M$ **do** ▷ Training iterations
  $\mathbf{g}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  for k $= N, \ldots, 1$ **do** ▷ Reverse diffusion process
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    $g_{k-1} = \frac{1}{\sqrt{\alpha_k}} \left(g_k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}} \epsilon_\theta(g_k, k, s) \right) + \sqrt{\beta_k}\epsilon$ ▷ Using Eq. (9)
  $k \sim \text{Uniform}(\{1, \ldots, N\})$
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  Calculate diffusion $\mathcal{L}_d(\theta), Q(s, \mathbf{g}_0, \pi(s)), r(g_d, \mathbf{g}_0)$ ▷ Calculate with the generated goal $\mathbf{g}_0$
  Calculate total loss $\mathcal{L} = \xi_d \mathcal{L}_d(\theta) - \xi_q Q(s, \mathbf{g}_0, \pi(s)) - \xi_r r(\mathbf{g}_0, g_d)$ ▷ Eq 11
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$ ▷ Take gradient descent step and update the diffusion weights
return $\mathbf{g}_0$

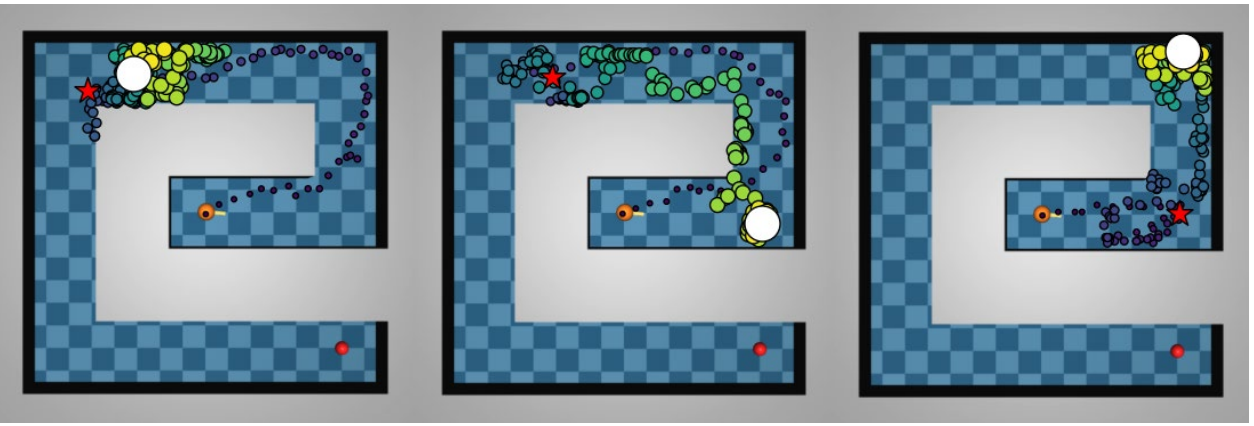
How Diffusion model is learning to generate curriculum goals.

Different Strategies to select curriculum goals

Selecting Curriculum Goals

Strategy s_T

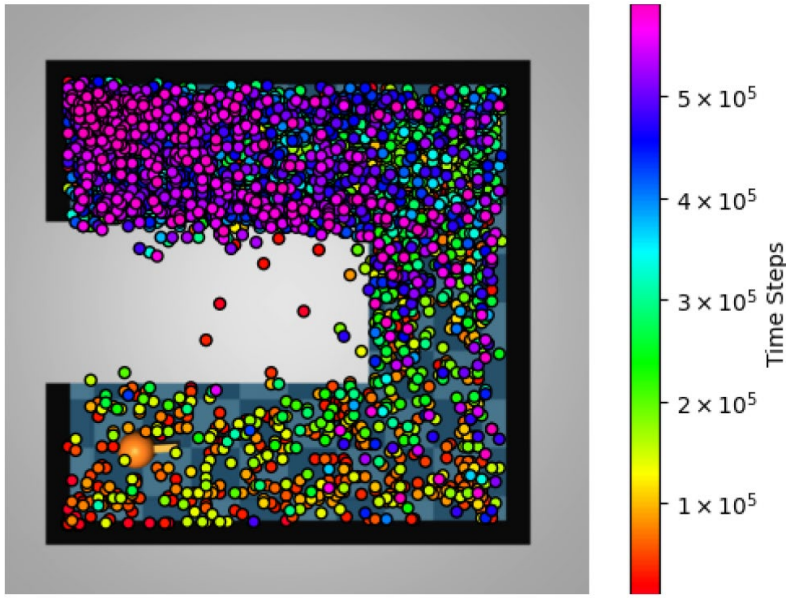
Strategy Optimization Algorithm



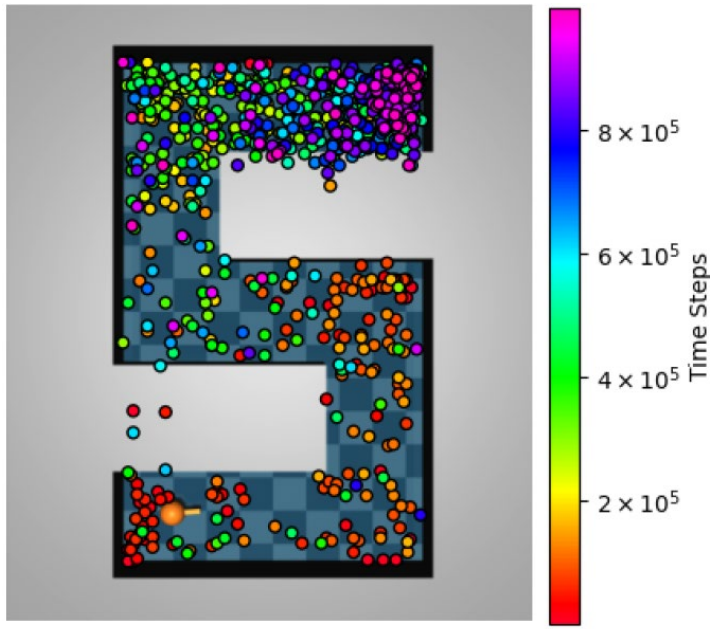
Episode **525** Episode **526** Episode **527**

$$\max_{\hat{\mathcal{G}}^c:|\hat{\mathcal{G}}^c|=K} \sum_{g^0_{t=0,..T} \in \hat{\mathcal{G}}^d, g^i_+ \in \hat{\mathcal{G}}^+} w \quad \text{where:} \quad w = \sqrt{\frac{(g_i - \bar{g})^2}{N}}$$

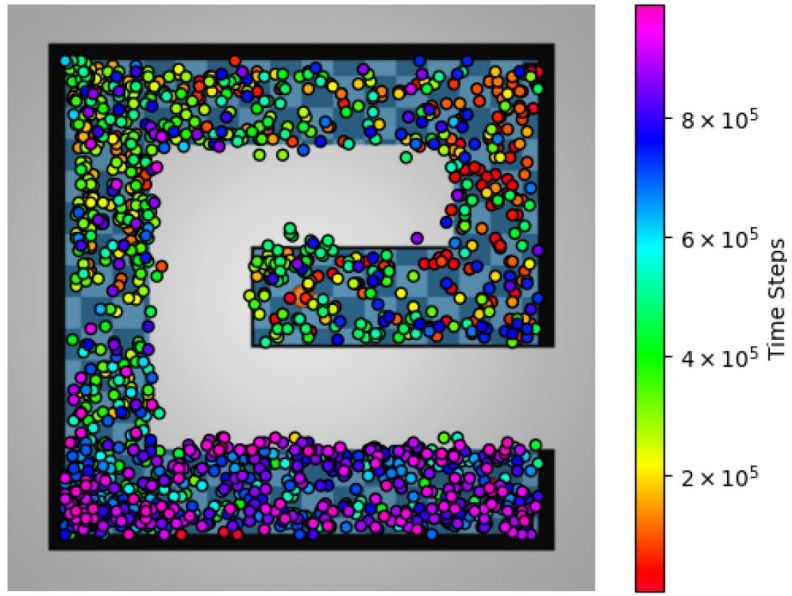
Generated curriculum goals during training



PointUMaze

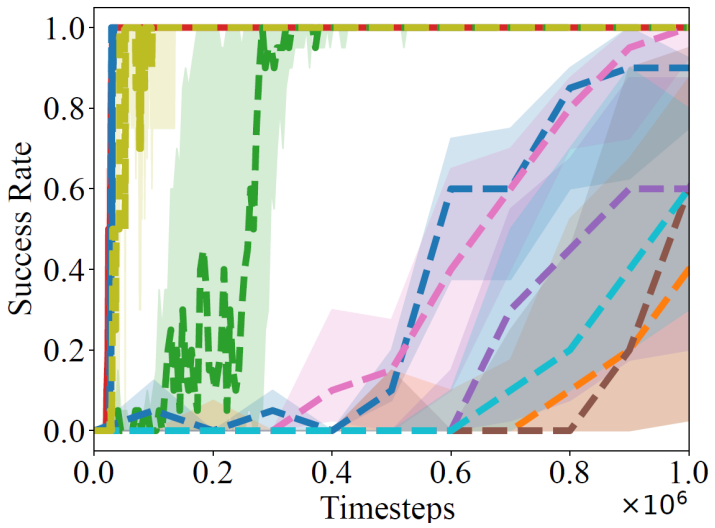


PointNMaze

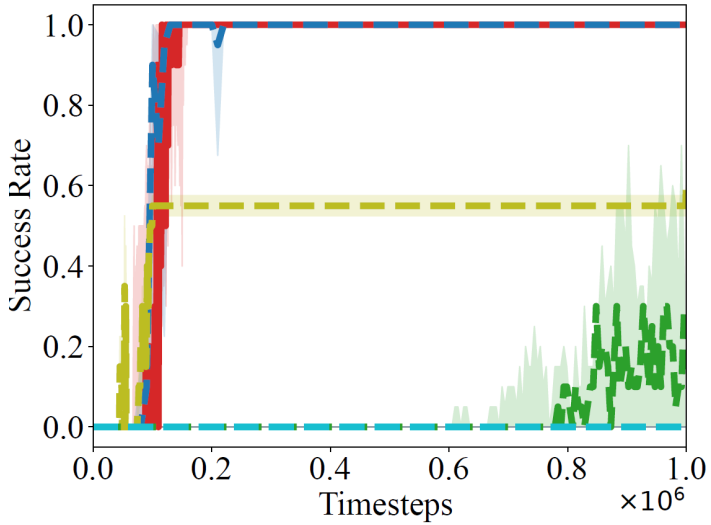


PointSpiralMaze

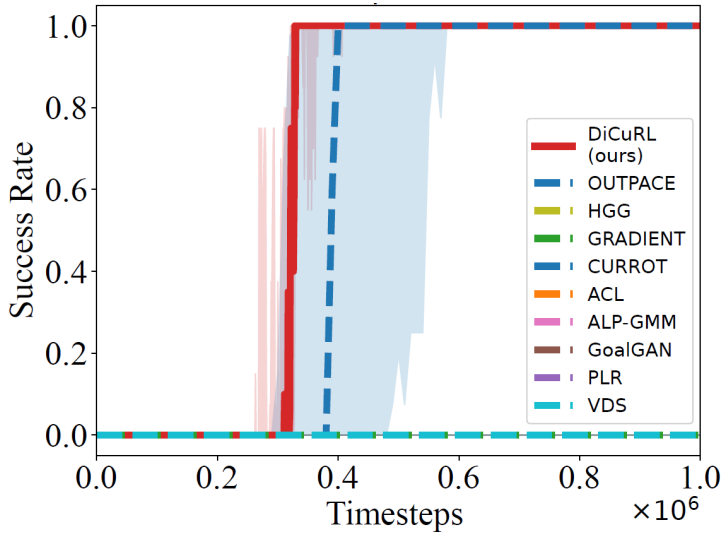
Results and Success Rate



PointUMaze



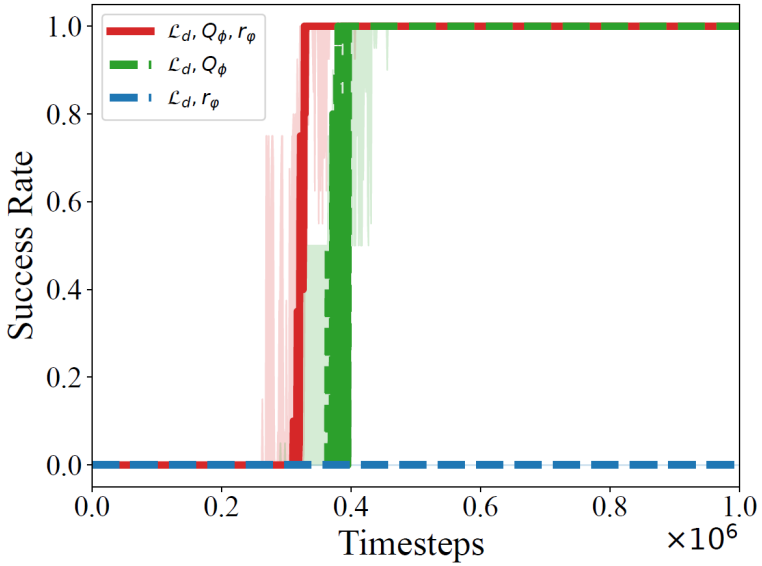
PointNMaze



PointSpiralMaze

Algorithm	PointUMaze	PointNMaze	PointSpiralMaze
DiCuRL (Ours)	28333 ± 3036	108428 ± 34718	305833 ± 43225
OUTPACE	29800 ± 4166	113333 ± 24267	396875 ± 111451
HGG	48750 ± 24314	NaN	NaN
GRADIENT	263431 ± 114795	NaN	NaN

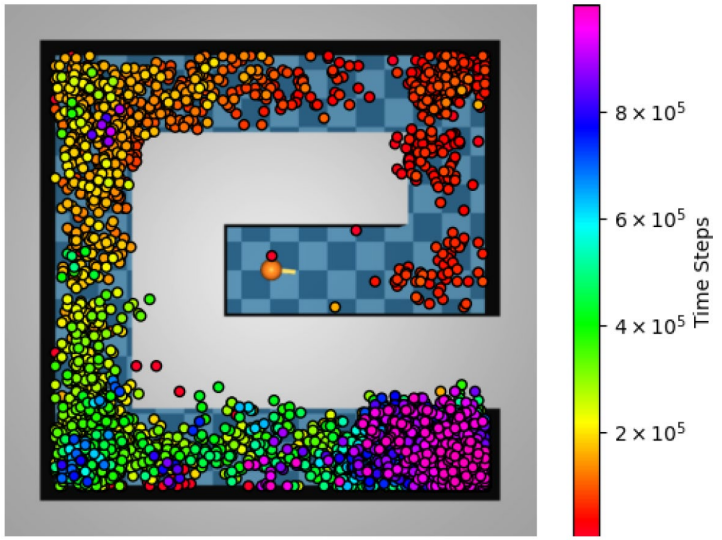
Ablation Study



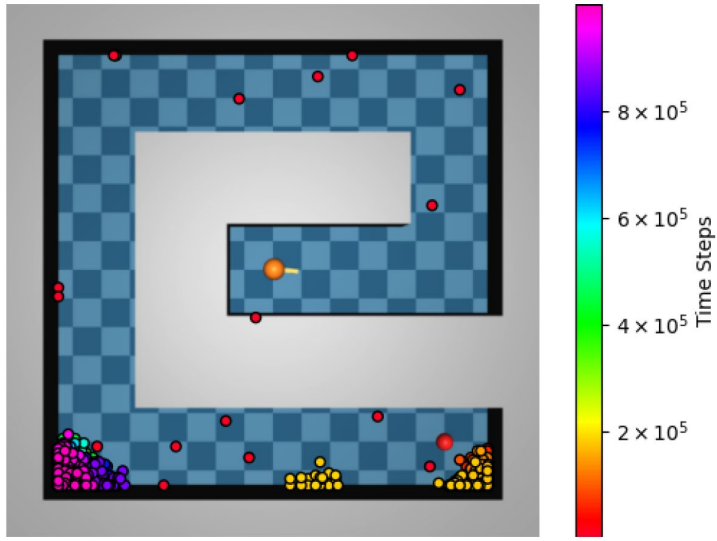
$$\mathcal{L} = \xi_d \cdot \mathcal{L}_d(\theta) - \xi_q \cdot Q_\phi(s, g_0, \pi(s, g_0)) - \xi_r \cdot r_\varphi(g_0, g_d)$$

$$\mathcal{L} = \xi_d \cdot \mathcal{L}_d(\theta) - \xi_q \cdot Q_\phi(s, g_0, \pi(s, g_0)) - \xi_r \cdot r_\varphi(g_0, g_d)$$

$$\mathcal{L} = \xi_d \cdot \mathcal{L}_d(\theta) - \xi_q \cdot Q_\phi(s, g_0, \pi(s, g_0)) - \xi_r \cdot r_\varphi(g_0, g_d)$$



Curriculum goals with Q



Curriculum goals with reward

Conclusions

- We introduced the **Di**ffusion Based **Cu**rriculum **R**einforcement **L**earning (DiCuRL) that utilizes diffusion models to generate curriculum goals.
- The diffusion model is trained to minimize its loss while simultaneously maximizing Q and Reward function.
- The generated goals promote exploration due to the inherent noising and denoising mechanism of the diffusion model.
- Additionally, we tested DiCuRL on two robot manipulation tasks, FetchPush and FetchPickAndPlace (see the results in the paper.)