# Unsupervised Contrastive Goal-Reaching

**Ahmed Turkman**[1] **Raj Ghugare**[2] **Benjamin Eysenbach**[2]
[1]African Institute for Mathematical Sciences [2] Princeton University
ahmedt@aims.ac.za

## Abstract

Goal-conditioned reinforcement learning (GCRL) enables agents to learn to achieve different goals. However, it is often limited by the need for a pre-defined goal sampling distribution. Prior works attempted to lift this limitation by training additional components that propose goals for the agent at the frontier of its capabilities, by fitting goal coverage density estimators, or by training additional goal samplers. These approaches are difficult to scale for relatively higher-dimensional goals due to the additional challenge of modeling or sampling high-dimensional variables. To address this problem, we introduce Unsupervised Contrastive Goal Reaching (UCGR), a simple algorithm that enables the agent to propose its own training goals without the need for additional networks or density estimators. UCGR leverages the learned critic in the contrastive reinforcement learning framework [13] as an implicit dynamics-aware model of reachability. Our experiments show that UCGR outperforms strong prior methods in a variety of tasks, particularly when goals are complex and high-dimensional.[1]

## 1 Introduction

Pretraining models on large-scale datasets with self-supervised objectives has become a cornerstone of modern machine learning. While these supervised models have achieved significant successes from natural language processing to scientific discovery, their reliance on human-collected data is a bottleneck for problems that humans aren't capable of solving – from unsolved theorems to unprecedented engineering feats. Reinforcement learning provides us with the machinery to develop agents that can collect their own data and solve novel problems.

Goal-conditioned reinforcement learning (GCRL) [22, 24] promises to combine these two strong paradigms. It enables developing agents that collect their own data and learn to reach goals in a self-supervised manner, without relying on human designed reward functions [35, 12, 13]. In GCRL, agents try to reach a set of pre-specified goals by maximizing a simple objective – the probability of reaching those goals in the future. GCRL algorithms collect data to reach these pre-specified goals, and use this data to improve their policy and collect better data.

There is still one assumption in GCRL which requires human supervision – a set of prespecified goals to reach. This can be limiting in many important problems where good outcomes are unclear or even unimaginable – for example, the synthesis of novel drugs or constructing novel architectures. Several algorithms have been developed to address this limitation, providing an automatic curriculum of goals for the agent. One class of methods suggested generating goals that maximize state space coverage by fitting density estimators over the set of visited states, while another class of methods proposed training additional networks to generate goals with medium difficulty. However, incorporating these additional components into the RL training makes it challenging to scale for environments with high-dimensional spaces or long-horizon tasks.

[1]Our code is publicly available: https://github.com/ahmed-turkman/Unsupervised-Contrastive-Goal-Reaching

2nd Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2025).

We propose Unsupervised Contrastive Goal-Reaching, or UCGR, an algorithm based on a simple autonomous goal-selection strategy enabling the agent to propose its own goals during training without requiring additional components. Our work builds on the contrastive reinforcement learning framework introduced by Eysenbach et al. [13] and makes the realization that this framework learns an implicit dynamics model of the environment. We propose choosing goals from the past achieved goal distribution that are maximally difficult based on the learned contrastive RL representations.

Our contributions include developing a novel algorithm for solving the GCRL problem based on a simple goal-selection strategy utilizing the contrastive RL framework. We compare our algorithm against strong prior methods: the original contrastive RL algorithm and an algorithm based on MEGA (Maximum Entropy Goal Achievement) [32]. We conduct experiments on AntMaze and Navix environments and our results show that for tasks with high-dimensional goals, UCGR is the only method to learn.

## 2 Related Work

Traditionally, RL assumes the existence of a reward function, defined as part of the problem [39]. In contrast, our work builds upon prior research trying to make RL self-supervised. Self-supervised RL can be broadly categorized into pure exploration [4, 6, 19, 28, 31, 40, 42], skill learning [1, 11, 18, 21, 25, 38], or goal conditioned RL [2, 3, 12, 22].

In this paper, we focus on the GCRL problem. GCRL can be thought of as an RL problem with a sparse reward function [20, 36] which is equal to 1 at the goal and 0 otherwise. Although this works for discrete problems [9, 10, 15], one needs to define some metric for defining "closeness" of the agent to the goal in continuous cases [2, 26, 27]. To lift this assumption and bridge the gap between GCRL and self-supervised learning, prior work unifies both the discrete and continuous cases by treating GCRL as the problem of maximizing probability densities instead [12, 23, 30, 37].

Based on this principle of maximizing probability densities, a large amount of progress has been made on the GCRL problem [13, 16, 34, 43]. These papers typically assume the existence of a desired goal distribution, which can be queried to collect data during training, as a part of the GCRL problem. Similar to the reliance on a reward function in standard RL, this assumption limits the applicability of GCRL: the distribution might provide goals that are beyond the agent's reach. This, in turn, hinders progress toward a fully self-supervised RL paradigm.

As a result, prior work has realized the need for self-supervised goal generation [14, 7, 32]. One line of work prioritized exploration-aware goals, maximizing state space coverage. Maximum Entropy Gain Exploration (MEGA) [32] fits a kernel density estimator (KDE) to the set of achieved goals and then selects new targets from the lowest-density areas, maximizing the entropy of the achieved goal distribution. Pong et al. [33] trains a generative model to propose goals that are skewed so that less-visited states gain a higher probability when fitting a generative model. Warde-Farley et al. [41] sampled goals from a diversified replay buffer in a way that stored goals have as far distance as possible. However, these methods do not necessarily account for the environment's temporal dynamics. Our experiments show that maximizing state coverage is not effective in complex long-horizon environments.

Our work is more similar to autonomous goal-selection methods that seek medium difficulty goals. The intuition is that generating goals that are neither difficult nor naive is most effective for learning, and the curriculum should continuously adapt to the agent's growing skills. Building on this intuition, Florensa et al. [14] suggested scoring the difficulty of goals based on rewards received when achieving them. They then trained a GAN [17] to generate goals with increasing difficulty. Campero et al. [7] introduced a teacher-student framework, where the teacher is responsible for proposing goals and is rewarded based on the performance of the student on those goals, while the student is rewarded based on both the external reward and the intrinsic reward provided by the teacher. Zhang et al. [44] used the epistemic uncertainty of the $Q-$function to estimate difficulty. They then fitted a distribution to the sample goals based on these estimates. While these methods might implicitly consider temporal dynamics of the environment, they all involve training additional components. Optimizing all of these components simultaneously can be challenging. We aim to address these limitations by proposing a new method that accounts for the environment's temporal dynamics, and does so without training additional components or networks.

## 3 Preliminaries

### 3.1 Goal-Conditioned Reinforcement Learning

A goal-conditioned MDP is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{G}, \mathcal{T}, \phi, r, \gamma, p_0, p_g \rangle$ [29] where $\mathcal{S}$, $\mathcal{A}$, $\mathcal{G}$, $\gamma$, $p_0$, and $p_g$ denote the state space, action space, goal space, discount factor, initial state distribution, and the desired goal distribution. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ represents the dynamics transition function, $\phi : \mathcal{S} \to \mathcal{G}$ is a function mapping from states to goals. Finally, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$ is the reward function. Unlike the standard MDP, it additionally depends on the desired goal the agent is seeking.

Similar to prior work [5, 8, 12, 35], we define the reward as the probability of reaching the goal at the following time step. However, as we show in 3.2, we adopt the contrastive RL framework which maximizes this reward only implicitly not explicitly.

$$r_g(s_t, a_t) \triangleq (1 - \gamma)\, p(s_{t+1} = s_g \mid s_t, a_t). \tag{1}$$

For a goal-conditioned policy $\pi(a \mid s, g)$, we define $p_t^{\pi(\cdot|\cdot,g)}(s)$ as the distribution of states visited by the agent after $t$ time steps. We define the discounted state visitation distribution or for simplicity, the future state distribution as:

$$p^{\pi(\cdot|\cdot,g)}(s_f = s) \;=\; (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t\, p_t^{\pi(\cdot|\cdot,g)}(s). \tag{2}$$

### 3.2 Contrastive Reinforcement Learning

Our work is built on the contrastive RL framework proposed by Eysenbach et al. [13]. It provided a new perspective for self-supervised RL by demonstrating that contrastive representation learning methods can be directly used as GCRL algorithms. This framework hinges on learning representations whose inner product corresponds to a goal-conditioned value function. The $Q$-function (corresponding to the reward function in Eq. 1) is equivalent to the state occupancy measure:

$$Q_{s_g}^{\pi(\cdot|\cdot,s_g)}(s, a) = p^{\pi(\cdot|\cdot,s_g)}(s_{t+} = s_g|s, a). \tag{3}$$

The framework learns two encoders: First, a state-action encoder $\phi(s, a)$ that maps a state-action pair $(s, a)$ to a $d$-dimensional representation vector. Second, a goal encoder $\psi(s_g)$ that maps a goal state $s_g$ (or a future state $s_f$) to the same $d$-dimensional latent space. A critic function (or similarity score) $f(s, a, s_f)$ is defined as the inner product of these representations: $f(s, a, s_f) = \phi(s, a)^\top \psi(s_f)$. The objective is to train these encoders such that $f(s, a, s_f)$ is high if the future state $s_f$ is likely to be reached after taking action $a$ in state $s$, and low otherwise. The main idea of this framework is that optimizing a contrastive objective based on these encoders, corresponds to maximizing the $Q$-function.

For a future state $s^+$ sampled from the trajectory of a state action pair $(s, a)$, and a batch of $(K - 1)$ random states $s^-$ sampled from the marginal distribution of future states, we define our contrastive objective based on the InfoNCE loss as:

$$\max_f \mathbb{E}_{\substack{(s,a)\sim p(s,a), s^-\sim p(s_f) \\ s^+\sim p^\pi(\cdot|\cdot)(s_f|s,a)}} [-\mathcal{L}_{\text{InfoNCE}}(s, a, s^+, \{s^-\}^{K-1}) - \lambda_{\text{reg}} \mathcal{R}(s, a, s^+, \{s^-\}^{K-1})], \tag{4}$$

where

$$\mathcal{L}_{\text{InfoNCE}}(s, a, s^+, \{s^-\}^{K-1}) = -\log \frac{\exp(f(s, a, s^+))}{\exp(f(s, a, s^+)) + \sum_{i=1}^{K-1} \exp(f(s, a, s_i^-))}, \tag{5}$$

The regularizer $\mathcal{R}$ is defined as

$$\mathcal{R}(s, a, s^+, \{s^-\}^{K-1}) = \log \left( e^{f(s,a,s^+)} + \sum_{i=1}^{K-1} e^{f(s,a,s^-_{(i)})} \right). \tag{6}$$

Finally, we learn an actor that chooses actions whose representations are close to the representations of the desired goals:

$$\max_{\pi(a|s,s_g)} \mathbb{E}_{\pi(a|s,s_g)\,p(s)\,p(s_g)}\big[f\big(s,\,a,\,s_f = s_g\big)\big]. \tag{7}$$

# 4 Unsupervised Contrastive Goal Reaching

In this section, we introduce the main contribution of this paper, UCGR, a simple method for unsupervised goal sampling for GCRL. Our method extends the original contrastive RL framework to the unsupervised setting, where an agent must learn to achieve a wide range of goals without being told which goals to practice. We first explain our Min-LogSumExp (MinLSE) goal-selection strategy in detail, then explain our proposed algorithm.

## 4.1 The MinLSE Goal Selection Strategy

A central challenge in unsupervised GCRL is defining a curriculum of goals for the agent to practice. An effective curriculum should guide the agent towards the frontier of its capabilities, goals that are reachable but not yet mastered. That's exactly the intuition behind UCGR. Unlike prior methods, UCGR does not require training additional networks, nor fitting an external density model. It leverages the contrastive critic function as an implicit density model over the space of reachable goals. For any given candidate goal $g$, we can estimate its reachability from the agent's collected experience stored in the replay buffer $\mathcal{B}$. We form a score for $g$ by aggregating its critic values over a batch of state-action pairs $\{(s_i, a_i)\}_{i=1}^{K} \subset \mathcal{B}$. We define this score using the LogSumExp operator:

$$S(g) = \log \sum_{i=1}^{K} \exp(f(s_i, a_i, g)). \tag{8}$$

Recalling that $f(s_i, a_i, g)$ is the similarity between $\phi(s_i, a_i)$ and $\psi(g)$, which encodes the reachability of the goal $g$ from the state-action pair $(s_i, a_i)$. Therefore, to find goals on the frontier of exploration, we seek those with the lowest reachability within our collected experience. UCGR implements this by selecting the goal $g^*$ that minimizes the MinLSE score over a set of candidate goals $\mathcal{G}_{cand} \subset \mathcal{B}_g$, where $\mathcal{B}_g$ is the set of achieved goals in the replay buffer.

$$g^* = \arg\min_{g \in \mathcal{G}_{cand}} S(g) = \arg\min_{g \in \mathcal{G}_{cand}} \log \sum_{i=1}^{K} \exp(f(s_i, a_i, g)). \tag{9}$$

## 4.2 A Complete GCRL Algorithm

We can now move forward and construct a complete GCRL algorithm. We integrate our proposed goal-selection strategy into the Contrastive RL (CPC) algorithm [13] to build an unsupervised GCRL algorithm, where the agent autonomously proposes its own learning goals during training, instead of relying on an external goal distribution. We provide the pseudocode in Algorithm 1.

**Algorithm 1** Unsupervised Contrastive Goal-Reaching (UCGR)

Initialize actor $\pi_\theta$, state-action encoder $\phi$, goal encoder $\psi$, and replay buffer $\mathcal{B}$.
Collect initial experience with a random policy and add to $\mathcal{B}$.
**for** each training step **do**
  **// Phase 1: Critic and Actor Updates**
  Sample a batch of state-action pairs $\{(s_i, a_i)\}_{i=1}^K$ from $\mathcal{B}$.
  For each $(s_i, a_i)$, sample a positive future goal $g_i^+$ from its trajectory using the state occupancy measure.
  Update critic encoders $\phi$ and $\psi$ by minimizing Eq. 4 using the batch $\{(s_i, a_i, g_i^+)\}_{i=1}^K$.
  Update actor $\pi_\theta$ by minimizing Eq. 7.
  **// Phase 2: MinLSE Goal Selection**
  For each $g_j^+ \in \{(s_i, a_i, g_j^+)\}_{i=1}^K$, compute the score $S(g_j^+) = \log \sum_{i=1}^K \exp(f(s_i, a_i, g_i^+))$.
  Select the exploratory goal $g^* = \arg\min_{g_i^+ \in \{(s_i, a_i, g_j^+)\}} S(g_i^+)$.
  **// Phase 3: Experience Collection**
  Collect a new trajectory $\tau$ by executing policy $\pi_\theta(a \mid s, g^*)$ in the environment.
  Add $\tau$ to the replay buffer $\mathcal{B}$.
**end for**=0

## 5 Experiments

The aim of our experiments is to answer three main questions. First, does incorporating UCGR into a contrastive RL algorithm improve its learning efficacy? Second, how does UCGR compare against prior unsupervised goal sampling strategies? Third, how does UCGR perform in high-dimensional spaces? In this section, we detail the experiments performed to answer these questions. We explain our choices for the baselines, the tasks we experimented on, and the results we found.

**Baselines.** We compared against two baselines. For answering the first question, we compared against the original contrastive RL (CPC) that was proposed by Eysenbach et al. [13]. Let's refer to it as "CRL". In this algorithm, goals are sampled directly from the desired goal distribution, without any unsupervised goal generation strategy. Testing against this baseline shows us the absolute improvement added by UCGR alone. The second baseline is "MEGA", which is an unsupervised goal sampling strategy introduced by Pitis et al. [32]. It suggests sampling goals from low-density areas of the previously achieved goal distribution. This method is a representative of a large class of prior work on unsupervised goal sampling. Testing against this baseline helps us answer our second question about the performance of UCGR against prior unsupervised goal sampling methods.

Since we only compare different goal-sampling methods, we use contrastive RL for all experiments as the backbone algorithm, while only changing the goal sampling procedure. We also ensure that common hyperparameters have the same value for different algorithms.

**Tasks.** We used two main types of environments to perform our experiments. Two examples are shown in Fig. 1. The AntMaze environment involves an ant with four legs that should learn how to navigate a maze and reach a commanded goal. There are three increasingly challenging mazes, all with easy and hard evaluation settings: AntMaze-U, AntMaze-Big, and AntMaze-Hardest. This is a standard locomotion task in the literature that was widely used to test GCRL algorithms. Its 29-dimensional observation space helps us observe how UCGR performs in high-dimensional spaces and answer our third research question. The second environment is Navix environment, which is a reimplementation of the MiniGrid environment suite in JAX. It is a standard environment for testing exploration for reinforcement learning algorithms, and it involves a triangle-like agent that needs to solve goal-oriented tasks. We used

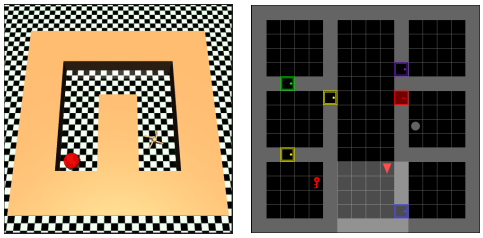
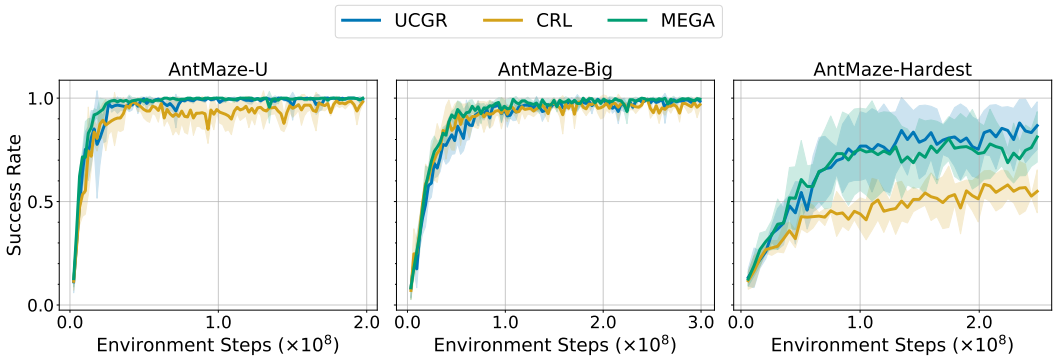
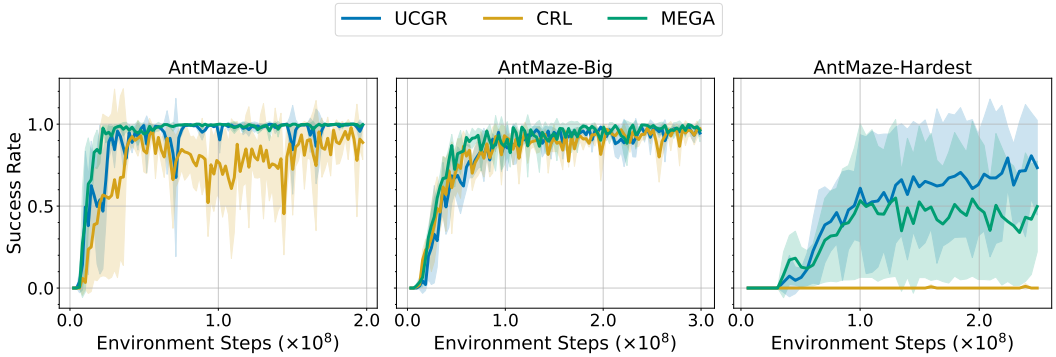


Figure 1: The AntMaze-U (left) and Navix-KeyCorridor (right) environments. Visuals adapted from the Farama Foundation's documentation at robotics.farama.org and minigrid.farama.org, respectively.




(a) Easy goal evaluation.




(b) Hard goal evaluation.

Figure 2: Performance comparison of UCGR, CRL, and MEGA in Ant Maze environments using **substate goals**. The shaded regions represent one standard deviation over three random seeds.

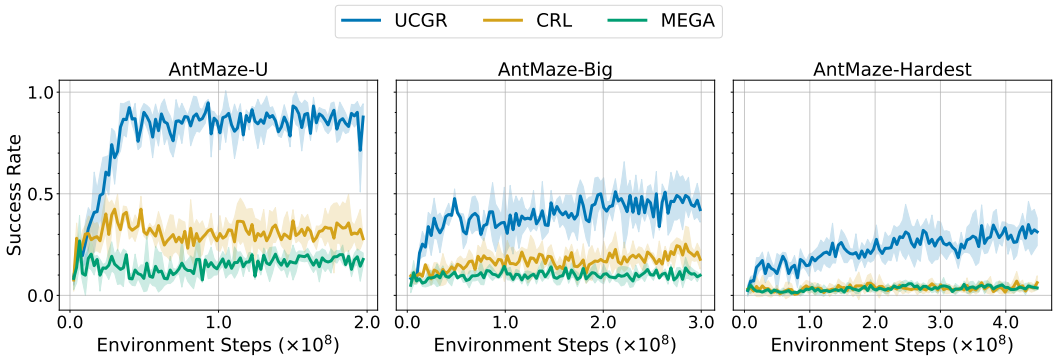
five different tasks in this environment: Empty, Four Rooms, DoorKey, KeyCorrridor-S3R2, and KeyCorridor-S3R3.

For AntMaze environments, we performed two types of experiments based on two different ways of defining goals. "Sub-state goals" are defined by the x-y coordinates components of the 29-dimensional state space, testing the ability of the agent in only reaching a target location. This case is the most common in the literature. To test the algorithms' ability in understanding the temporal relationships between goals, we also defined the goals to be full states in "Full-state goals". In this case, the agent receives a full description of a state that it should match, including a target location, a target orientation, target joint angles, and target velocities.

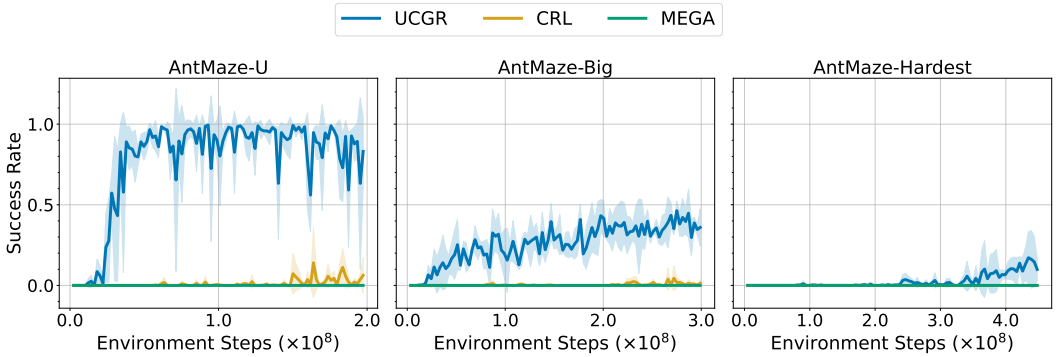
**Evaluation Metrics.** To evaluate the performance of the three algorithms in the different tasks, we used three evaluation metrics. Success Rate is the proportion of parallel agents that succeed in reaching the goal at least once during an episode. Goal Maintenance is the proportion of the episode's time that was spent by the agent at the goal, averaged across all parallel agents. Average Distance is the average distance from the goal during an episode, averaged across all parallel agents.

### 5.1 Results and Discussion

**AntMaze - Sub-state Goals.** In the AntMaze-U environment with sub-state goals as shown in Fig. 2, both the unsupervised goal sampling methods, UCGR, and MEGA, achieved better sample efficiency than the supervised CRL, especially in the hard evaluation setting (Fig. 2b), showing the effectiveness of unsupervised goal sampling. However, the performance of both of them is very comparable. In the AntMaze-Big, the three algorithms achieved similar performance with negligible differences. In the AntMaze-Hardest, the most difficult maze, our method clearly outperformed the two baselines in both the easy and hard evaluation settings. CRL completely failed in the hard evaluation setting, showing its inability to learn in long-horizon environments.




(a) Easy goal evaluation.




(b) Hard goal evaluation.

Figure 3: Performance comparison of UCGR, CRL, and MEGA in Ant Maze environments using **full-state goals**. The shaded regions represent one standard deviation over three random seeds.

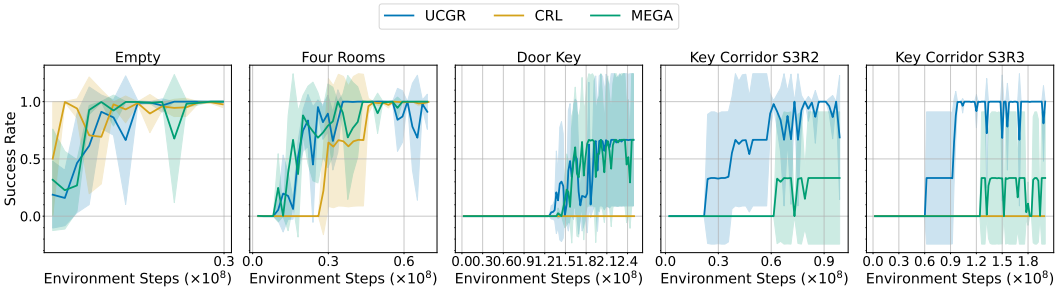



Figure 4: Performance comparison of UCGR, CRL, and MEGA across five Navix environments. The shaded regions represent one standard deviation over three random seeds.

**AntMaze - Full-state Goals.** The superiority of our method is clear in the case of Full-state goals (Fig. 3), where it significantly outperformed the two baselines by large margins. In the AntMaze-U environment, our method was able to achieve almost a perfect success rate after only about 50M environment steps, while baselines were stuck below 0.4 success rate in the easy evaluation setting (Fig. 3a) and were not able to show any progress in the hard evaluation setting (Fig. 3b). Similarly, our method was the only algorithm to show success in the hard evaluation of AntMaze-Big and AntMaze-Hardest (Fig. 3b). We believe this strong performance is due to UCGR's ability to account for the temporal structure of the environment.

**Navix.** Similar pattern appeared with the Navix environments (Fig. 4). Only in the easiest Navix-Empty task, CRL was able to learn faster than UCGR and MEGA, as it was trained directly on achieving the evaluation goal, which is simple in this task. The performance of UCGR and Mega is comparable in this task. In the Navix-Four-Rooms and Navix-Door-Key, the performance of

UCGR and MEGA is comparable without significant differences, while both of them achieved higher sample efficiency than the supervised CRL. The differences between the three algorithms appeared clearly in the Navix-Key-Corridor tasks that involve long-horizon exploration challenges. UCGR significantly outperformed the two baselines, showing its ability to understand the temporal structure of the environment.

## 6 Conclusion

In this paper, we introduced UCGR, a novel algorithm for unsupervised GCRL. Utilizing the learned critic by Contrastive RL algorithms, UCGR empowers the agent with an autonomous goal-generation ability, based on its own understanding of the environment. We showed that prior methods that seek to maximize goal-space coverage fail in tasks with complex temporal structure, highlighting UCGR as a strong GCRL algorithm.

One limitation of UCGR is that goal generation is restricted to previously achieved goals from the buffer. This makes UCGR undefined in the case of dynamical environments that are changing over time. Future research should tackle this challenge by designing algorithms that are simple and are not restricted to generating goals from the buffer.

**Acknowledgments** This paper is based on the master's thesis of Ahmed Turkman, completed at the African Institute for Mathematical Sciences (AIMS) South Africa. We would like to thank the AIMS South Africa family for providing an excellent learning environment, and Google DeepMind for generously supporting his studies and research. We additionally acknowledge the use of Princeton Research Computing resources at Princeton University which is a consortium of groups led by the Princeton Institute for Computational Science and Engineering (PIC- SciE) and Office of Information Technology's Research Computing.

## References

[1] Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

[2] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[3] Bram Bakker, Jürgen Schmidhuber, et al. Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In *Proc. of the 8-th Conf. on Intelligent Autonomous Systems*, pages 438–445, 2004.

[4] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[5] Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.

[6] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[7] Andres Campero, Roberta Raileanu, Heinrich Küttler, Joshua B Tenenbaum, Tim Rocktäschel, and Edward Grefenstette. Learning with amigo: Adversarially motivated intrinsic goals. *arXiv preprint arXiv:2006.12122*, 2020.

[8] Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *International conference on machine learning*, pages 1430–1440. PMLR, 2021.

[9] Peter Dayan. Navigating through temporal difference. *Advances in neural information processing systems*, 3, 1990.

[10] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

[11] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[12] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*, 2020.

[13] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

[14] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pages 1515–1528. PMLR, 2018.

[15] David Foster and Peter Dayan. Structure in the space of value functions. *Machine Learning*, 49 (2):325–346, 2002.

[16] Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. *arXiv preprint arXiv:1912.06088*, 2019.

[17] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[18] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

[19] Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR, 2019.

[20] Frank S He, Yang Liu, Alexander G Schwing, and Jian Peng. Learning to play in a day: Faster deep reinforcement learning by optimality tightening. *arXiv preprint arXiv:1611.01606*, 2016.

[21] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

[22] Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pages 1094–8, 1993.

[23] Hilbert J Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and experiment*, 2005(11):P11011, 2005.

[24] John E Laird, Allen Newell, and Paul S Rosenbloom. Soar: An architecture for general intelligence. *Artificial intelligence*, 33(1):1–64, 1987.

[25] Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191*, 2021.

[26] Charline Le Lan, Marc G Bellemare, and Pablo Samuel Castro. Metrics and continuity in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8261–8269, 2021.

[27] Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. *AI&M*, 1(2):3, 2006.

[28] Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021.

[29] Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

[30] Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in neural information processing systems*, 31, 2018.

[31] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

[32] Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *International conference on machine learning*, pages 7750–7761. PMLR, 2020.

[33] Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

[34] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[35] Tim GJ Rudner, Vitchyr Pong, Rowan McAllister, Yarin Gal, and Sergey Levine. Outcome-driven reinforcement learning via variational inference. *Advances in Neural Information Processing Systems*, 34:13045–13058, 2021.

[36] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.

[37] Yannick Schroecker and Charles Isbell. Universal value density estimation for imitation learning and goal-conditioned reinforcement learning. *arXiv preprint arXiv:2002.06473*, 2020.

[38] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

[39] Richard S Sutton and Andrew G Barto. Reinforcement learning: an introduction, 2nd edn. adaptive computation and machine learning, 2018.

[40] Sebastian B Thrun. *Efficient exploration in reinforcement learning*. Carnegie Mellon University, 1992.

[41] David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *arXiv preprint arXiv:1811.11359*, 2018.

[42] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 11920–11931. PMLR, 2021.

[43] Zilai Zeng, Ce Zhang, Shijie Wang, and Chen Sun. Goal-conditioned predictive coding for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:25528–25548, 2023.

[44] Yunzhi Zhang, Pieter Abbeel, and Lerrel Pinto. Automatic curriculum learning through value disagreement. *Advances in Neural Information Processing Systems*, 33:7648–7659, 2020.