Extended Abstract

Motivation Offline goal-conditioned RL (GCRL) can leverage large amount of weakly-labeled or reward-free trajectories to learn goal-conditioned policies, but the resulting value functions are sometimes noisy. Hierarchical Implicit Q-Learning (HIQL) mitigates this by treating the value function as a latent "planner" and extracting hierarchical policies from it. However, we find HIQL also fails in more complex environments because its state–goal embedding is co-trained with the same noisy value estimates, letting representation errors propagate into the sub-goal space and negatively affect downstream control. In this project, we ask: Can we improve HIQL by replacing its embedding with an independently-learned state–goal encoder?

Method In HIQL, a value function is first learned through GCIVL, after which a pair of hierarchical policies are extracted with an AWR objective: a high-level policy $\pi^h\big(\phi(s_{t+k}) \,\big|\, s, g\big)$ that produces a sub-goal embedding and a low-level policy $\pi^l\big(a \,\big|\, s, \phi(s_{t+k})\big)$ that chooses the action to complete that sub-goal, where ϕ is the embedding co-trained with the value function. We instead pre-train a pair of state- and goal-encoders ϕ' and ψ' with a contrastive objective on offline data, replace the original encoder, and keep them frozen during subsequent value learning and policy extraction; the resulting policies become $\pi^h\big(\psi'(s_{t+k}) \,\big|\, \phi'(s), \psi'(g)\big)$ and $\pi^l\big(a \,\big|\, \phi'(s), \psi'(s_{t+k})\big)$, trained in the same way as in the original framework, and we expect these pre-trained contrastive encoders to give helpful signals that improve value estimation and downstream hierarchical learning.

Implementation We implement contrastive pre-training by sampling mini-batches of triples (u, v^+, v^-) from the offline data, where v^+ is a state reachable from u within H steps and v^- is a state that lies beyond H steps; both are selected without using rewards and (u, v^+, v^-) are from the same trajectory. Training maximizes the contrastive objective $\log \sigma\big(\phi'(u)^\top \psi'(v^+)/\tau\big) + \log\big(1 - \sigma(\phi'(u)^\top \psi'(v^-)/\tau)\big)$, i.e. it explicitly increases the dot product between positive pairs while reducing it for negative pairs; We tested our methods on OGBench, an offline goal-conditioned RL benchmark which consists of different tasks and data types. For each task, the pretraining takes about 40 minutes on 20k steps on an AWS g4.2xlarge instance. After pretraining, we replace the original encoder in HIQL with the state goal encoders and keep them frozon during the value function learning and policy extraction. Each single task result takes 2-4 hours to be evaluated.

Results Our method improves HIQL's success rate across PointMaze, AntMaze, and Cube environments, with gains of up to 20–40% over GCBC and 10–15% over HIQL. In particular, we observe strong performance in both dense (navigate) and sparse (explore, stitch) goal-reachability settings. Our method have relative equivalent performance in HumanoidMaze, but high-DoF makes the task difficult to learn through our method. Moreover, performance drops in multi-skill tasks such as AntSoccer, where short, non-Markovian trajectories make contrastive supervision less effective.

Discussion The results demonstrate the effectiveness and benefits of decoupling representation learning from noisy value signals in the original HIQL structure. However, current contrastive pretraining is trajectory-local and limited in stitching across diverse behaviors. This may explain weaker performance in complex tasks with discontinuous dynamics or multiple skills. Future improvements could explore cross-trajectory pretraining and objectives that handle non-Markovian noise or richer behavior structure.

Conclusion In conclusion, we show that pretraining goal representations using contrastive learning improves downstream value learning and hierarchical policy extraction in offline GCRL. Our approach achieves higher success rates across several benchmark tasks, highlighting the importance of robust embeddings in hierarchical RL. This work opens up new directions for leveraging self-supervised objectives to enhance policy learning in static, complex environments.

Better Goals, Better Policies: Goal Representation for Offline Hierarchical RL

Yongce Li
Department of ICME
Stanford University
yongceli@stanford.edu

Xiaoyue Wang
Department of Computer Science
Stanford University
wxy0115@stanford.edu

Abstract

Offline goal-conditioned reinforcement learning (GCRL) can utilize large amounts of unlabeled trajectories, yet the performance is often limited by the noisy value estimates. Prior work (HIQL) extracts hierarchical policies by inferring sub-goal embeddings directly from the value function, but this approach still fails on tasks with more complex environments. In this project, We revisit HIQL and strengthen its state-goal representation by decoupling embedding learning from value learning. Specifically, we pre-train paired state and goal encoders with a horizon-aware contrastive loss: states reachable within H steps are pulled together, while those farther apart are pushed away. These frozen encoders replace the original embedding in HIQL. Across diverse OGBench tasks, our pretrained encoders improve navigation and exploration success rates by about 10 %. However, it's still limited on high-DoF humanoid mazes and skill-composition tasks, where trajectories are either short, non-Markovian, or require stitching different behaviors. Overall, our findings highlight both the promise of contrastive pretraining for offline GCRL and the need for methods that better exploit non-Markovian or multi-skill data.

1 Introduction

Recent advances in large language and vision models demonstrate the power of self-supervised pre-training on unlabelled or weakly supervised data (Achiam et al., 2023; Radford et al., 2021). Analogously, offline goal-conditioned reinforcement learning (GCRL) exploits large stores of reward-free trajectories to acquire goal-conditioned policies. However, value functions learned from such data can be noisy, hindering the extraction of reliable low-level controllers (Park et al., 2023). Hierarchical Implicit Q-Learning (HIQL) (Park et al., 2023) addresses this limitation by first training a high-level policy that predicts a sub-goal embedding, which is then supplied to a low-level policy to produce actions. Although HIQL achieves strong results on many tasks, we observe its failures in purely state-based environments where the inputs to the high-level policy are raw states rather than pre-computed feature embeddings (Park et al., 2025). Prior work shows that contrastive representation learning can uncover latent structure in large amount of data and yield high-quality embeddings for downstream tasks (Radford et al., 2021). Building on this insight, we propose to enhance HIQL by pre-training a state and goal encoders with a contrastive objective on the same unsupervised trajectories, thereby providing more informative sub-goal embeddings and improving performance on tasks where standard HIQL underperforms.

A core limitation of HIQL is that its state–goal embedding is learned jointly with the value function. Although an embedding co-trained with the optimal value function V^* would be optimal theoretically, the value estimates obtained from offline trajectories are typically noisy and not optimal. These errors distort the learned state embedding space, leading to inaccurate sub-goal representations and degraded policies. We mitigate this issue by decoupling representation learning from value

learning: before any value updates, we pre-train a state–goal encoder with a contrastive objective that maximizes the dot product between a state s_t and future states reachable within H steps, while minimizing its similarity to states lying beyond this horizon. The resulting encoder replaces HIQL's original representation module and is kept frozen during subsequent value-function training and policy extraction. By injecting a horizon-aware signal, the contrastive encoder is expected to provide more coherent sub-goal embeddings and improve overall policy performance on tasks where standard HIQL underperforms.

We evaluated our approach on OGBench, an offline goal-conditioned RL benchmark that spans navigation, locomotion, data-stitching, and robotic-control tasks. Across navigation and exploration domains, the contrastive pre-trained encoder extracts informative signals from offline datasets, improving HIQL's success rate by roughly 10%. It's limited, however, on tasks that need stitching short trajectories or composing distinct skills on complex environment/controls, highlighting a promising direction for future work.

2 Related Work

Offline RL aims to learn effective policies from pre-collected datasets without training on additional environment interactions (Lange et al., 2012; Levine et al., 2020). Previous works like Batch-Constrained deep Q-learning (BCQ) (Fujimoto et al., 2019) and Conservative Q-Learning (CQL) (Kumar et al., 2020) address the distribution shift between behavior policies and learned policies by constraining policy updates or regularizing Q-values in offline RL. Implicit Q-Learning (IQL) (Kostrikov et al., 2021) relies on value-based learning with implicit behavior modeling for offline learning.

Goal-Conditioned RL (GCRL) methods learn policies conditioned on specific goals, which enable agents to learn and perform multiple tasks with specified goals and policies. Hindsight Experience Replay (HER) (Andrychowicz et al., 2017; Chebotar et al., 2021; Fang et al., [n. d.]; Levy et al., 2017; Li et al., 2020; Pong et al., 2018; Yang et al., 2022) relabels failed trajectories with alternative goals to enable sample-efficient learning. Universal Value Function Approximators (UVFA) (Schaul et al., 2015) incorporates goal information into value functions to learn goal-aware representations. Other works also introduce algorithms based on various techniques like contrastive learning (Eysenbach et al., 2022, 2020; Zhang et al., 2021), state occupancy matching (Durugkar et al., 2021; Ma et al., 2022a), and Goal-conditioned behavioral cloning (GCBC) (Lynch et al., 2020; Ghosh et al., 2019a).

Offline Goal-Conditioned RL combines offline learning with goal-conditioned RL in order to address the challenge of sparse rewards, distributional shifts, and invalid goal relableing in the static datasets. Works like GoFAR (Ma et al., 2022b) frames goal-reaching as a state-distribution matching problem, GCSL (Ghosh et al., 2019b) treats GCRL as supervised-learning, and GOPlan (Ghosh et al., 2019c) generates trajectories conditioned on goals through learned diffusion models to have better learned agents. The HIQL methods (Park et al., 2023), introduces a hierarchical approach that learns a single goal-conditioned value function offline using IQL and builds a two level policy: a high-level sub-goal generator and a low level goal-conditioned controller. However, it shows that the state-based methods underperform compared to pixel-based ones.

Relevant Evaluation Benchmark To support and evaluate these methods, there are multiple works evaluated methods on single tasks or behaviors (Myers et al., 2024; Yang et al., 2023), but they lack the comprehensive, standardized benchmark that exhaustively assesses various properties with diverse tasks. recent benchmarks such as OGBench (Park et al., 2025) provide a diverse set of offline datasets designed specifically for evaluating GCRL algorithms. These datasets vary by task type (e.g., navigation, manipulation), data quality (e.g., expert, exploratory), and structure (e.g., stitched sub-trajectories vs. full trajectories), enabling more systematic analysis of goal-reaching under offline constraints. Our method builds on this foundation and aims to improve sub-goal representations by learning more stable and informative embeddings, thereby enhancing the state and goal encoding in the high-level policy.

3 Method

3.1 Preliminaries

3.1.1 Goal-conditioned Implicit V-learning

GCIVL is a goal-conditioned RL algorithm using the idea of implicit Q-learning (IQL) to learn a value function using expectile regression. It fits a value function $V(s, g)$ by minimizing the value loss:

$$\mathcal{L}(V) = \mathbb{E}_{s,g,s' \sim \tau}[l_k^2(r(s, g) + \gamma \bar{V}(s', g) - V(s, g))],$$

where $\bar{V}$ is the target value function, $r(s, g) = 1_g(s) - 1$ is the sparse goal-conditioned reward function, and $l_k^2(x) = |k - 1_{\{x:x<0\}}(x)|x^2$ is the expectile loss used in IQL. After learning the value function, it extracts a goal-conditioned policy with the AWR objective:

$$\mathcal{J}_{\text{AWR}}(\pi) = \mathbb{E}_{s,a,s',g \sim \tau}[e^{\alpha(V(s',g)-V(s,g))} \log \pi(a|s, g)].$$

3.1.2 Hierarchical Implicit Q-learning

HIQL is a hierarchical goal-conditioned RL method which contains two level policies: a high-level policy $\pi^h(\phi(s_{t+k})|s, g)$ which produces a subgoal embedding and a low-level policy $\pi^l(a|s, \phi(s_{t+k}))$ which generates the next action to achieve the subgoal. It first learns a value function using GCIVL, and then extract the policies with the following AWR objective:

$$\mathcal{J}_{\pi^h}(\theta_h) = \mathbb{E}_{(s_t,s_{t+k},g)}[\exp(\beta * (V_{\theta_V}(s_{t+k}, g) - V_{\theta_V}(s_t, g))) \log \pi^h_{\theta_h}(\phi(s_{t+k})|s_t, g)],$$

$$\mathcal{J}_{\pi^l}(\theta_l) = \mathbb{E}_{(s_t,a_t,s_{t+1},s_{t+k})}[\exp(\beta * (V_{\theta_V}(s_{t+1}, s_{t+k}) - V_{\theta_V}(s_t, s_{t+k}))) \log \pi^l_{\theta_l}(a_t|s_t, \phi(s_{t+k}))].$$

Note that the extraction of the two policies are decoupled and independent from each other, while they all depends on the value function. In the above equations, ϕ is the subgoal representation model learned through the value function loss.

3.1.3 Contrastive representation learning

Recent studies show that contrastive learning excels at extracting rich representations from large-scale data in both computer vision and natural language processing. The core idea is to learn two encoders that pull positive (similar) input pairs together in representation space while pushing negative (dissimilar) pairs apart. For a positive pair (u, v^+) and a negative pair (u, v^-), we seek encoders ϕ and ψ that maximize the similarity $\phi(u)^\top \psi(v^+)$ and minimize $\phi(u)^\top \psi(v^-)$. We will train the representation models using the binary Noise-Contrastive Estimation (NCE) objective:

$$\max_{\phi,\psi} \mathbb{E}_{(u,v^+,v^-)\sim D}\Big[\log \sigma\big(\phi(u)^\top \psi(v^+)\big) + \log\big(1 - \sigma\big(\phi(u)^\top \psi(v^-)\big)\big)\Big], \tag{1}$$

where $\sigma(\cdot)$ denotes the sigmoid function.

3.2 Contrastive Goal Representation Pretraining

In HIQL, the goal representation ϕ is usually learned alongside the value function and then refined and updated when extracting the actor. However, any over- or under-estimation in the value function can bleed into ϕ, distorting the goal space and hindering policy learning. Motivated by the recent success of contrastive representation learning, we propose pretraining ϕ with a contrastive objective on trajectory data to break this dependency, completely decoupled from the value-loss signal. We expect a value-agnostic embedding to deliver richer, more stable goal features for subsequent value estimation and actor extraction.

We train the state encoder ϕ and the goal encoder ψ by constructing contrastive state–goal pairs (u, v) from the offline dataset $\mathcal{D}$. An anchor state is drawn uniformly from all stored states, $u = s_t \sim p_{\mathcal{D}}(s)$, while a positive goal state is sampled from the same trajectory within a short horizon, $v = s_g \sim p_{\mathcal{D}}\big(s_{t+\tau} \mid s_t,\ 1 \le \tau \le H\big)$, and negative ones are sampled from states at least $H + 1$ time steps ahead, $v = s_g \sim p_{\mathcal{D}}\big(s_{t+\tau} \mid s_t,\ \tau > H\big)$. Optimizing a contrastive objective that pulls positive pairs closer in representation space while pushing negative pairs apart produces goal embeddings that encode short-horizon reachability without inheriting bias from value-function estimation. In

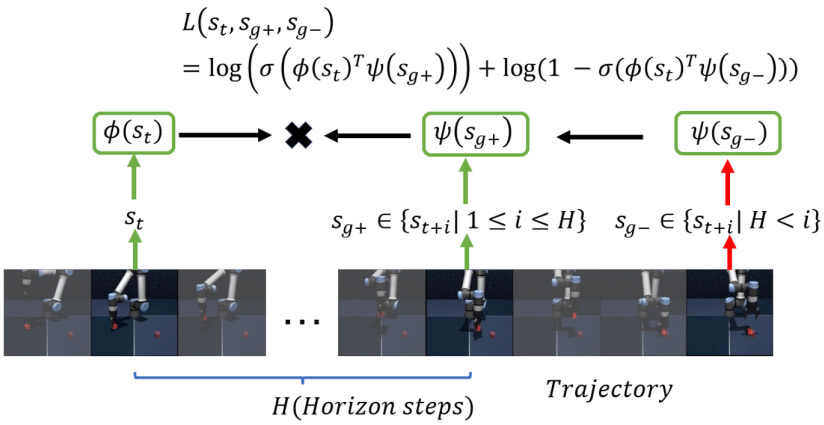

Figure 1: We use contrastive learning to learn a state encoding model ϕ and a goal encoding model ψ, so that the representation of a H-reachable goal state is closer to the current anchor state than a non H-reachable goal state.

summary, our contrastive objective is simply asking if the future state is H steps reachable from the current state:

$$\max_f \mathbb{E}_{\substack{s\sim p_{\mathcal{D}}(s) \\ s_{g+}\sim p_{\mathcal{D}}\left(s_{t+\tau}|s_t,\ 1\le\tau\le H\right) \\ s_{g-}\sim p_{\mathcal{D}}\left(s_{t+\tau}|s_t,\ \tau>H\right)}} \left[\log\sigma(\phi(s)^\top\psi(s_{g+})) \ + \ \log(1-\sigma(\phi(s)^\top\psi(s_{g-})))\right].$$

After training the encoding models, we replace the encoding model ϕ in the original HIQL implementation with our trained state and goal encoding and keep them frozon during the value learning and policy extraction process.

4 Experimental Setup

4.1 Dataset

We use OGBench benchmark, which contains multiple tasks for offline goal-conditioned RL evaluation Park et al. (2025), for evaluating our methods.

4.1.1 Task Type

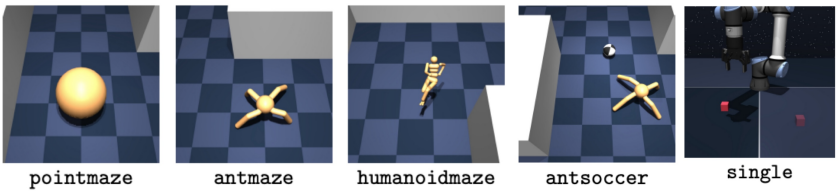

Figure 2: Dataset Task Type Demonstration. From left to right is: Pointmaze, Antmaze, Humaniodmaze, Antsoccer, and Cube

The tasks we evaluated are the following tasks, which also shown in Figure 2:

- Pointmaze: This task is to control a 2D point mass agent to reach a goal location in a given maze.
- Antmaze: This task is to control a quadrupedal Ant agent with 8 degrees of freedom to reach a goal location in a given maze.

- Humanoidmaze: This task is to contraol a complex 21 degrees of freedom humanoid agent to reach a goal location in a given maze.
- Antsoccer: This task build upon the simple maze navigation tasks, involves controlling an Ant agent to dribble a scocer ball. It requires the agent not only to navigate to the correct position, but also control the ball with it. The data provided with this task contains two part of trajectories: one is maze navigation without the ball and another one is dribbling with the ball near the agents.
- Cube: This task is to control a robot arm to pick and place the cube blocks into designed configurations.

4.1.2 Data Type

All maze related tasks have various data type for training and evaluation process, as also shown in Figure 3:

Figure 3: Data Type Illustrations. From left to right is Navigate, Stitch, and Explore data type.

- Navigate: The standard dataset in the benchmark, collected by a noisy expert policy that navigates the maze.
- Stitch: Different than navigate type, stitch data consists of short goal-reaching trajectories. The agents need to stitch multiple trajectories together to complete the tasks.
- Explore: It consists of random exploratory trajectories, aiming to test whether the agent can learn navigation skills from low quality data.

4.1.3 Task Size

There are multiple size and difficulty choices for tasks, the larger the task, the more difficult the task:

- For maze task, the maze size and difficulty we evaluated can be medium, large, and giant, example mazes with different size are illustrated by Figure 4.
- For cube task, the number of cube we evaluated are single and double
- For antscoccer task, the scene difficulty we evaluated are arena and medium, which are without obstacles and with obstacles, illustrated by Figure 5.

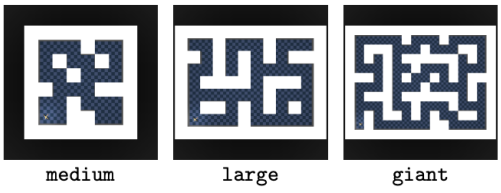

Figure 4: Maze Task Data Size Illustration

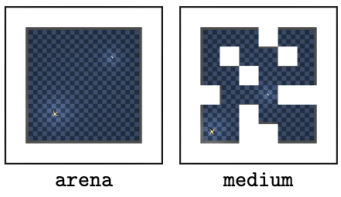

Figure 5: Antsoccer Task Data Size Illustration

4.2 Baseline

We evaluate our method against two primary baselines: GCBC (Goal-Conditioned Behavior Cloning) and HIQL (Hierarchical Implicit Q-Learning), both of which are included in the OGBench benchmark. GCBC serves as a simple behavior cloning baseline that directly maps state-goal pairs to actions without leveraging any value function, while HIQL represents a more advanced hierarchical approach that uses goal-conditioned value learning and subgoal generation. For consistency, we use the reported performance of these baselines from the OGBench leaderboard where available. Additionally, we re-implemented and ran HIQL on selected tasks to ensure the reproducibility of results and confirm they fall within the reported performance range. This provides a reliable basis for assessing the effectiveness of our proposed enhancements.

4.3 Evaluation Metric

In line with the OGBench evaluation protocol, we use success rate as our primary metric. Success rate is defined as the percentage of evaluation episodes in which the agent successfully reaches the specified goal state within the allowed time horizon. Each task is evaluated with varying initial states and goals, and success is determined based on whether the final state lies within a predefined threshold steps of the goal.

5 Results

Our experiments results are shown in Table 1. All the experiments are evaluated by succuss rate, and we provide the results of GCBC and HIQL from the OGBench with average rate and standard deviations. We also provide sample succuss trajectories in Appendix A.

5.1 Quantitative Evaluation

For **maze-related** tasks, the results demonstrate that our method achieves performance that is equivalent to or better than HIQL on both medium and large-scale navigation environments, while significantly outperforming GCBC across the board. In **PointMaze**, our method achieves a success rate of **98%** on the `medium-navigate` task, compared to **79%** for HIQL and **9%** for GCBC. On the `large-stitch` variant, our method reaches **23%**, surpassing HIQL (**13%**) and GCBC (**7%**). Similarly, in **AntMaze**, our method achieves **47%** on the `medium-explore` task, exceeding HIQL (**37%**) and GCBC (**2%**), and scores **22%** on the `large-explore` task, again significantly higher than HIQL (**4%**) and GCBC (**0%**). These results highlight the effectiveness of our contrastive pretraining approach in producing robust state and goal embeddings that generalize well across spatially complex environments.

In the more challenging **HumanoidMaze** task, our method achieves a score of **2%**, which is comparable to HIQL's **3%**, while GCBC fails entirely (**0%**). This suggests our method retains competence even in high-dimensional control scenarios. A detailed breakdown and discussion of this result will follow in the next section.

For **Cube tasks**, our method clearly outperforms GCBC in both settings. On `cube-single-play`, we achieve **23%**, compared to **15%** from HIQL and **6%** from GCBC. On the more complex `cube-double-play`, our method matches HIQL with a success rate of **6%**, while GCBC lags behind at **1%**. These results suggest that our method not only enhances goal representation quality but also generalizes effectively to tasks requiring precise object manipulation.

In contrast, for **AntSoccer** tasks, our method underperforms relative to both HIQL and GCBC. On the `medium-navigate` variant, our method scores **0%**, compared to **13%** from HIQL and **2%** from GCBC. In the `arena-stitch` task, we achieve only **1%**, whereas GCBC achieves **24%** and HIQL scores **15%**. This outcome reveals a limitation of our current approach in multi-skill environments that require coordination between navigation and dynamic object interaction like dribbling a ball. We hypothesize that richer temporal dynamics or contact-aware representations may be necessary, and explore this further in the following section.

Overall, these results indicate a strong potential of our contrastive goal pretraining method in structured navigation and manipulation tasks, with particularly strong gains in maze and cube-based environments.

Table 1: Experiments Results of Succuss Rate (in %) on Various Dataset. The GCBC and HIQL have multiple runs and reported the average succuss rate with standard deviation. The highest performances are bold.

Environment	Dataset	GCBC	HIQL	Our Method
PointMaze	pointmaze-medium-navigate-v0	9 ± 6	79 ± 5	**98**
	pointmaze-large-stitch-v0	7 ± 5	13 ± 6	**23**
AntMaze	antmaze-medium-explore-v0	2 ± 1	$\mathbf{37 \pm 10}$	**47**
	antmaze-large-explore-v0	0 ± 0	4 ± 5	**22**
HumanoidMaze	humanoidmaze-giant-stitch-v0	0 ± 0	$\mathbf{3 \pm 2}$	2
Cube	cube-single-play-v0	6 ± 2	15 ± 3	**23**
	cube-double-play-v0	1 ± 1	$\mathbf{6 \pm 2}$	**6**
AntSoccer	antsoccer-arena-stitch-v0	$\mathbf{24 \pm 8}$	15 ± 1	1
	antsoccer-medium-navigate-v0	2 ± 0	$\mathbf{13 \pm 2}$	0

5.2 Qualitative Analysis

1. **PointMaze**
   (a) *pointmaze-medium-navigate-v0* **(navigate).**
   In this experiment, the offline data trajectories are long, expert-like trajectories which densely populate the maze. This is ideal for contrastive learning because it can cover most neighboring states, so the frozen encoder learns an almost Euclidean metric of reachability. Decoupled from the value function noise, its additional positional signal helps the value network and policies plan better. Hence we observe a gain ($\sim 9\%$) over the original HIQL method.
   (b) *pointmaze-large-stitch-v0* **(stitch).**
   In this experiment the maze is larger, and the dataset contains only short, goal-directed trajectories. During contrastive pre-training we sample state pairs solely within each trajectory, so the encoder receives no supervision on "stitching" separate paths together. Despite this limitation we still observe a $\sim 10\%$ boost over the baseline, likely because the encoder captures the maze's local geometry while the subsequent value- and policy-learning stages learn to generalize across segments.
2. **AntMaze**
   (a) *antmaze-medium/large-explore-v0* **(explore).**
   In this environment, data are random exploratory data covering the whole maze. Even though the data contains noise, the global positional knowledge dominates the embedding learning thus helps the value and policy learning.
3. **HumanoidMaze**
   (a) *humanoidmaze-giant-stitch-v0* **(stitch).**
   In this task the contrastive encoder provides no benefit over original HIQL. The agent struggles to maintain locomotion for more than a few steps because the training set is composed of very short trajectories, while the 21-DoF humanoid dynamics and the giant maze create an exceptionally complex landscape. Lacking long-range positives, the encoder ends up modeling mainly local pose variations—not genuine spatial progress—so it cannot help downstream value or policy learning.
4. **Cube**
   (a) *cube-single/double-play-v0*
   In the cube environment the agent controls a robotic arm that must set cubes at a target pose, but the demonstrations are highly non-Markovian—many contain arbitrary pick-and-place detours unrelated to task progress. With a single cube these distractions are infrequent, so the contrastive encoder still extracts a useful task-space metric and boosts learning. With two cubes, more and more different states are still reachable within H steps, so the model ends up mixing genuine moves with random cube-shuffling and the

representation gets confused. Consequently the learned policy often oscillates between picking up and dropping cubes instead of executing a coherent placement plan.

5. **AntSoccer**
   (a) *antsoccer-medium/arena-navigate/stitch-v0.*
   In the AntSoccer task the agent must operate a robot ant while pushing a ball to the goal position, but the dataset splits into two disjoint behaviors—"ant locomotion without ball" and "ball control with little locomotion." When contrastive pre-training treats states from both modes as interchangeable positives, the resulting embedding cannot reconcile these conflicting skills. As a result the learned policy either stands still or walks off without the ball, never integrating locomotion and ball control into a coherent strategy.

6 Discussion

A core limitation of our contrastive pre-training is that positive and negative pairs are sampled only within a single trajectory; we lack a reliable way to judge whether a state from a different trajectory is H-step–reachable. This isolates each roll-out in its own place of representation space and makes "stitching" trajectories together difficult for long-horizon or compositional tasks. The issue is compounded when demonstrations are non-Markovian: exploratory or redundant actions hide the true notion of progress and can mislead the encoder. Future work could explore (i) principled mechanisms to create cross-trajectory pretraining pairs, and (ii) objectives that remain robust in the presence of meaningless/non-Markovian transitions. While our results confirm that contrastive pre-training can give helpful signals for value- and policy-learning, systematically comparing alternative self-supervised objectives may discover more data-efficient embeddings.

The main challenges we faced during this project were time and resource constraints. We encountered repeated issues with AWS, including difficulties in launching or maintaining instances, which significantly slowed down our progress. Additionally, each task required both contrastive representation training and policy training, making the overall process time-consuming. Combined with the AWS issues, this made it particularly challenging to run as many experiments as we initially planned.

7 Conclusion

In conclusion, we proposed a contrastive pretraining approach for goal representation learning in offline hierarchical reinforcement learning. By decoupling representation learning from value estimation, our method addresses a key limitation of HIQL, which is its reliance on noisy value-based embeddings, and instead produces stable, horizon-aware subgoal encodings from unsupervised data.

Our method demonstrates significant improvements in complex navigation and manipulation environments, outperforming existing baselines on PointMaze, AntMaze, and Cube tasks. In particular, it achieves up to around 20 to 40% higher success rates than GCBC and offers consistent gains over HIQL on several tasks. However, we also identified limitations in multi-skill and stitching-heavy environments such as AntSoccer and HumanoidMaze, where contrastive pretraining struggles to reconcile diverse behaviors or sparse long-horizon dependencies.

These findings highlight both the potential and the current limitations of using contrastive learning for goal representations. Future work could explore ways to incorporate temporal structure, task-specific knowledge, or cross-trajectory information to improve generalization. It may also be valuable to design hybrid objectives that capture not just spatial reachability but also meaningful task progress. Overall, our approach points toward promising directions for making offline hierarchical reinforcement learning more robust and scalable through better representation learning.

8 Team Contributions

Our team work on method and experiments designs together, and also discuss and write report together in person or over the zoom, with more details of some work split:

- **Yongce Li:** Work on method implementation and contrastive representation learning, as well as help on experiment running.

- **Xiaoyue Wang:** Work on contrastive representation learning and experiment running and evaluations.

All other related works are done together.

Changes from Proposal Our original Proposal is about to enhance the capability of current model-based reinforcement learning approaches to effectively handle long-horizon planning tasks in the Minecraft game environments. We found that Minecraft environments are too complex and time-consuming to complete our experiments within the project timeline. To better align with our constraints, we identified a paper with a similar original idea (Park et al., 2023). However, the paper shows that the state-based methods underperform compared to pixel-based ones. We hypothesize that this is due to poor sub-goal representation caused by inadequate encoding. Therefore, we aim to improve sub-goal representation within the algorithm.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

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

Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, et al. 2021. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749* (2021).

Ishan Durugkar, Mauricio Tec, Scott Niekum, and Peter Stone. 2021. Adversarial intrinsic motivation for reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 8622–8636.

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

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

Meng Fang, Cheng Zhou, Bei Shi, Boqing Gong, Jia Xu, and Tong Zhang. [n. d.]. DHER: Hindsight experience replay for dynamic goals. In *International Conference on Learning Representations*.

Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-Policy Deep Reinforcement Learning without Exploration. In *International Conference on Machine Learning*. 2052–2062.

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

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

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

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline Reinforcement Learning with Implicit Q-Learning. (2021).

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems* 33 (2020), 1179–1191.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. 2012. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*. Springer, 45–73.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. 2017. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948* (2017).

Alexander Li, Lerrel Pinto, and Pieter Abbeel. 2020. Generalized hindsight for reinforcement learning. *Advances in neural information processing systems* 33 (2020), 7754–7767.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. 2020. Learning latent plans from play. In *Conference on robot learning*. Pmlr, 1113–1132.

Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. 2022a. How Far I'll Go: Offline Goal-Conditioned Reinforcement Learning via f-Advantage Regression. *arXiv preprint arXiv:2206.03023* (2022).

Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. 2022b. How Far I'll Go: Offline Goal-Conditioned Reinforcement Learning via f-Advantage Regression. *arXiv preprint arXiv:2206.03023* (2022).

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. 2024. Learning Temporal Distances: Contrastive Successor Features Can Provide a Metric Structure for Decision-Making. *arXiv preprint arXiv:2406.17098* (2024).

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. 2025. OGBench: Benchmarking Offline Goal-Conditioned RL. In *International Conference on Learning Representations (ICLR)*.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. 2023. Hiql: Offline goal-conditioned rl with latent states as actions. *Advances in Neural Information Processing Systems* 36 (2023), 34866–34891.

Vitchyr Pong, Shixiang Gu, Murtaza Dalal, and Sergey Levine. 2018. Temporal difference models: Model-free deep rl for model-based control. *arXiv preprint arXiv:1802.09081* (2018).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PmLR, 8748–8763.

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

Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. 2022. Rethinking goal-conditioned supervised learning and its connection to offline rl. *arXiv preprint arXiv:2202.04478* (2022).

Rui Yang, Lin Yong, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. 2023. What is essential for unseen goal generalization of offline goal-conditioned rl?. In *International Conference on Machine Learning*. PMLR, 39543–39571.

Tianjun Zhang, Benjamin Eysenbach, Ruslan Salakhutdinov, Sergey Levine, and Joseph E Gonzalez. 2021. C-planning: An automatic curriculum for learning goal-reaching tasks. *arXiv preprint arXiv:2110.12080* (2021).

A Experiment Trajectory Samples

We provided some sample trajectories for succuss cases in our experiments.

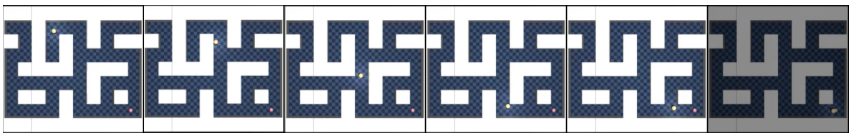

Figure 6: Succuss Trajectory Sample of pointmaze-large-stitch dataset

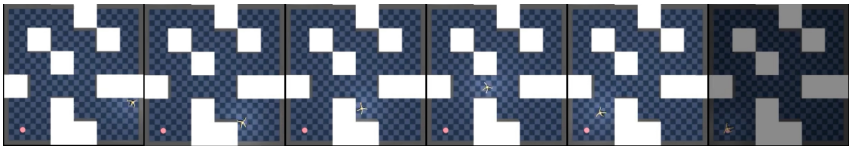

Figure 7: Succuss Trajectory Sample of antmaze-medium-explore Dataset

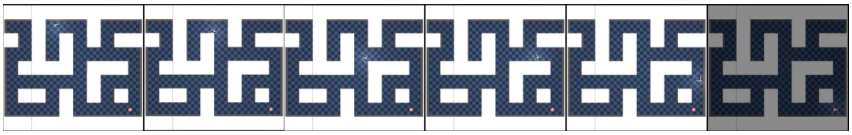

Figure 8: Succuss Trajectory Sample of antmaze-large-explore Dataset

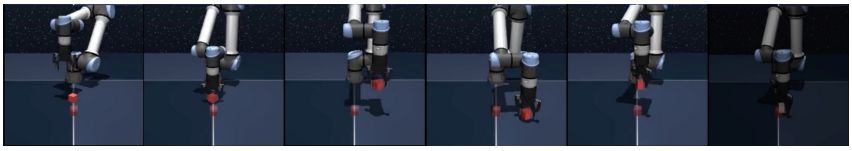

Figure 9: Succuss Trajectory Sample of cube-single-play Dataset