A Single Goal is All You Need:
Skills and Exploration Emerge from Contrastive RL without Rewards, Demonstrations, or Subgoals

Grace Liu[*] **Michael Tang** **Benjamin Eysenbach**
Princeton University

Abstract

In this paper, we present empirical evidence of skills and directed exploration emerging from a simple RL algorithm long before any successful trials are observed. For example, in a manipulation task, the agent is given a single observation of the goal state (see Fig. 1) and learns skills, first moving its end-effector, then pushing the block, and finally lifting and placing the block. These skills emerge before the agent has ever successfully placed the block at the goal location and without the aid of any reward functions, demonstrations, or manually-specified distance metrics. Implementing our method involves a simple modification of prior work and does not require density estimates, ensembles, or any additional hyperparameters. We lack a clear theoretical understanding of why the method works so effectively, though our experiments provide some hints.

Videos and code: `https://graliuce.github.io/sgcrl`

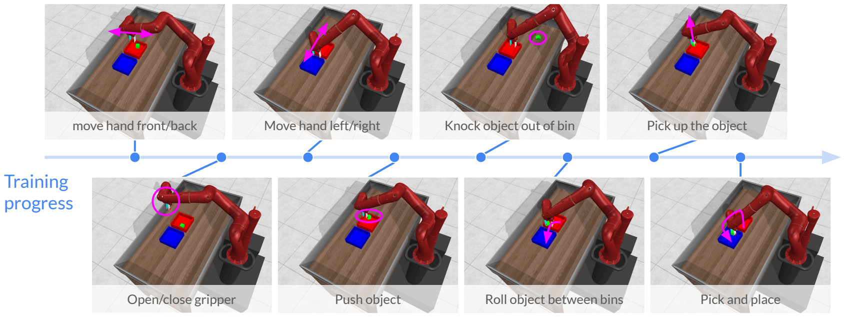


Figure 1: **Skills and Directed Exploration Emerge.** In this bin picking task, we provide the agent with a single goal observation where the green block is in the left bin. The agent never receives any rewards. Throughout the course of training, the agent learns skills that increase in complexity. Easier skills seem to enable the agent to unlock more complex skills: moving the hand is a prerequisite for pushing the object; closing the gripper is a prerequisite for picking up the object, which is a prerequisite for moving the object. Figures 7 and 8 in Appendix D show similar progress milestones for other tasks.

1 Introduction

Exploration is one of the grand challenges in reinforcement learning (RL) [69]. While there is a long history of exploration methods, even today's best methods fail to explore in settings with sufficiently sparse rewards, and the complexity of sophisticated exploration techniques means that most methods employ limited exploration techniques (e.g., adding random noise to actions [23, 29, 35, 42]). In this paper, we focus on a specific type of RL problem [6, 32, 37]: the agent is given an observation of the

[*] Correspondence to: Grace Liu <gliu2@andrew.cmu.edu>

Accepted at the Intrinsically Motivated Open-ended Learning workshop at NeurIPS 2024.

single desired goal state, which it tries to reach. This problem setting is exceedingly challenging for standard RL methods, as the agent does not receive any reward feedback about *how* it should solve the task. Because of the difficulty of this exploration problem, prior work typically assumes that a human user can provide a dense reward function [27, 77] (or distance metric/threshold [6, 54, 74]) or a set of easier training goals[2] [18]. A more comprehensive discussion of prior work can be found in Appendix section A. Our paper lifts the assumptions of prior work by considering a setting that is easier for human users but significantly more challenging for RL agents: a single goal state is provided and is used for both training and evaluation.

We present a simple RL algorithm where skills and directed exploration emerge long before any successful trials are ever observed. Implementing this method involves a simple modification of prior work and does not require density estimates, ensembles, or any additional hyperparameters. Our method works by learning a goal-conditioned value function via contrastive RL (CRL) [18] and using that value function to train a goal-conditioned policy. The key ingredient is embarrassingly simple: when doing exploration, always condition the goal-conditioned policy on the single target goal.

Empirically, we evaluate our approach on tasks ranging from bin picking to peg insertion to maze navigation, finding that it significantly outperforms prior methods that use a manually-designed curricula of subgoals [18], methods that automatically propose subgoals for training [6], and even methods that use dense rewards [26]. While we still lack a theoretical understanding of *why* this approach is so effective, experiments highlight that *(1)* the contrastive representations used to express the value function are important, and that *(2)* the gains are not caused by "overfitting" the policy or the value function to the single target goal.

2 Single-Goal Exploration with Contrastive RL

2.1 Preliminaries

Notation. We consider a controlled Markov process (i.e., an MDP without a reward function) defined by time-indexed states s_t and actions a_t. Our experiments will use continuous states and actions. The initial state is sampled $s_0 \sim p_0(s_0)$ and subsequent states are sampled from the Markovian dynamics $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$. Without loss of generality, we assume that episodes have an infinite horizon. We assume that the algorithm is given as input the single target goal state s^* and aims to learn a policy $\pi(a_t \mid s_t)$ by interacting with the environment. *Unlike prior work, we do not assume that a distribution of goals for exploration is given; we do not assume that either a dense or sparse reward function is given.*

Following prior work [16, 61], we define the objective as maximizing the probability of reaching the goal. Formally, define the γ-discounted state occupancy measure [11, 30, 67] as $\rho^\pi(s_f) \triangleq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t^\pi(s_t = s_f)$, where $p_t^\pi(s_t = s_f)$ is the probability of being at state s_f at time step t. In continuous settings, $p_t^\pi(s_f)$ is a probability *density*. The objective is to find a policy that maximizes the likelihood of the single target goal under this occupancy measure: $\max_\pi \rho^\pi(s_f = s^*)$. In discrete settings, this objective is equivalent to the standard discounted reward objective with a reward function $r(s_t, a_t) = \mathbb{1}(s_t = s^*)$; in continuous settings, it is equivalent to using a reward function $r(s_t, a_t) = p(s' = s^+ \mid s_t, a_t)$.

Contrastive RL. Our method builds on *contrastive RL* [18], prior work that uses temporal contrastive learning to solve goal-conditioned RL problems. Contrastive RL is an actor-critic method. The critic $C(s, a, s_f)$ is learned so that it outputs the (relative) likelihood that an agent starting at state s and taking action a will visit state s_f. Following prior work, we parameterize the critic as the dot product between two learned representations, $\phi(s,a)^T\psi(s_f)$ and learn these representations using the infoNCE contrastive objective [45] together with a LogSumExp regularization. In practice, this loss is implemented by sampling a random (s_t, a_t) pair from the replay buffer and then sampling a future state $s_f = s_{t+\Delta}$ by looking $\Delta \sim \textsc{Geom}(1-\gamma)$ steps ahead. The negative examples are obtained by shuffling the future states. Once learned, the representations encode a Q-value [18]: $\phi(s,a)^T\psi(s_f) = \log Q(s, a, s_f) - \log \rho(s_f)$. The policy is learned to maximize this (log) Q-value: $\max_\pi \mathbb{E}_{p(s)p(g)\pi(a|s,g)}\left[\phi(s,a)^T\psi(s_f) + \alpha\mathcal{H}(\pi(\cdot|s,s_f))\right]$, where α is an adaptive entropy coefficient.

[2] Some prior methods automatically propose training goals [20, 21, 46, 65], yet these methods require additional machinery and are primarily evaluated on settings where subgoals lie on a 2-dimensional manifold.

2.2 Our Approach

Our approach is a simple modification of contrastive RL: rather than manually providing training goals for exploration, we always command the policy to collect data with the single hard goal s^*. No other modifications are made. We will call this method "single [hard] goal CRL." Note that this is a multi-task algorithm because multiple goals are used for training the actor, even though it only ever collects data conditioned on a single goal and success is evaluated on a single goal. Ablation experiments (Fig. 4) show that only training the actor on the single goal decreases performance.

3 Experiments

The main aim of our experiments is to evaluate the performance of single-goal contrastive RL compared to its multi-goal counterpart as well as prior baselines. We do so on four exploration-heavy, goal-reaching tasks, involving robotic manipulation and maze navigation. All experiments were run with five random seeds, and error bars in the plots depict the standard error. Hyperparameters can be found in Appendix C and code to reproduce our results is available online: `https://graliuce.github.io/sgcrl`

Tasks. We measure the efficacy of our method on four goal-reaching tasks taken from prior work [15], which are chosen to measure long horizon exploration. The tasks include three robotic manipulation environments [77] and one maze navigation environment [17]. The robotic manipulation tasks require controlling a sawyer robot to grasp an object and accurately place it in a predetermined location. The point spiral task is a 2D maze navigation task. We quantify success in each episode by whether the agent reached sufficiently close to the goal in at least one state in an episode.

3.1 Single-goal Exploration is Exceedingly Effective.

A single goal works well. *We find that Contrastive RL effectively solves these four tasks*: equipped only with a single target goal, the agent automatically explores the environments and learns complex manipulation skills (see Fig. 2). We compare this method to an "oracle" variant that is trained on human-designed goals that vary in difficulty, ranging from easy goals to the single hard goal. Surprisingly, our proposed method significantly outperforms this "range of difficulties" method.

Directed exploration emerges. Not only does single-goal CRL consistently achieve high success rates on manipulation tasks, but it also demonstrates complex and directed exploration techniques during early training. To observe the agent's behavior throughout training, we saved learning checkpoints at fixed intervals and visualized the agent's behavior at each checkpoint (see Figure 1 and Figures 7, 8 in Appendix D). We found that the agent learns simple skills before complex ones. For example, in all three environments, the agent *(1)* first learns how to move its end-effector to varying locations, *(2)* then learns how to nudge and slide the object, and *(3)* finally learns to pick up and direct the object. We also observe a wide array of exploratory behavior in early training that is not directly connected to the goal. Refer to Appendix section D for a more detailed account of skill development.

3.2 The RL Algorithm is Key

Baselines. We compare single-goal CRL against prior methods that aim to address the sparse reward problem by making additional assumptions or employing additional machinery for exploration. Reinforcement learning with imagined subgoals (RIS) [6] maintains a high-level policy that predicts subgoals halfway to the end goal. We also employ a few variants of Soft Actor-Critic (SAC) with additional assumptions: SAC with sparse rewards, SAC+HER with sparse rewards, and SAC with dense rewards. In the dense reward setting, the agent receives a continuous reward tailored to the environment, using the distance to the goal for the spiral task and the Metaworld [77] reward function for sawyer tasks. Refer to Table 2 in appendix C for a summary of the assumptions for each method.

Results. As shown in Fig. 3 single-goal contrastive RL significantly outperforms these alternative methods, showing that the underlying RL algorithm is important for single goal exploration. Prior methods rarely reach the goal at all, with the exception of RIS on the simplest task (`point spiral`).

3.3 Why does Single-Goal Exploration Work?

The effectiveness of single-goal exploration is not explained by overfitting. One possible explanation for the method's success is that the algorithm overfits its policy parameters on the single-

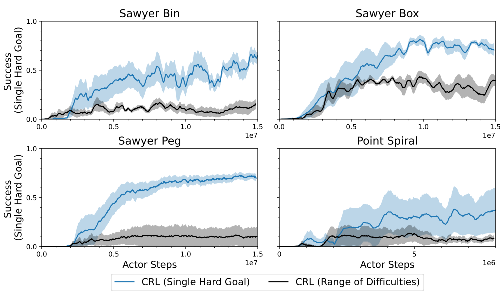


Figure 2: **Single goal Exploration is Highly Effective.** We compare single hard goal exploration (command the single hard goal in every trial) to "range of difficulties" exploration (sampling uniformly from a human-provided set of easy/medium/hard goals). In each of the four environments, single-goal exploration yields considerably higher success rates, all while being easier for the human user

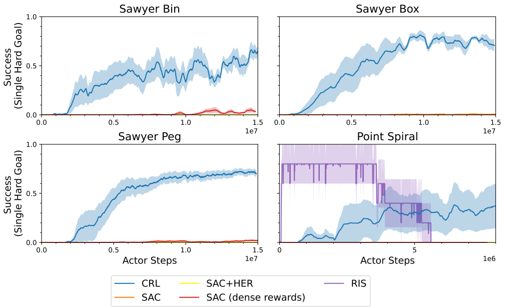


Figure 3: **The RL Algorithm Matters.** We compare several underlying RL algorithms all using single-goal exploration. CRL outperforms these prior methods, showing that *(i)* single-goal exploration is only effective with the right underlying RL algorithm and that *(ii)* with this algorithm, we can achieve considerably higher performance while making fewer assumptions (no rewards).

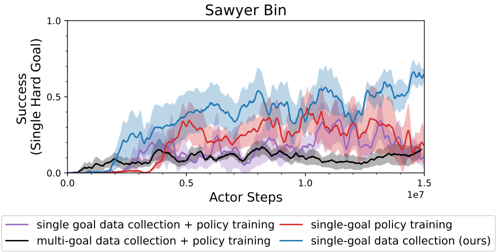


Figure 4: **Ablation experiments.** While our method uses an actor loss that uses many goals, alternatives that train the policy with a single goal perform no better.

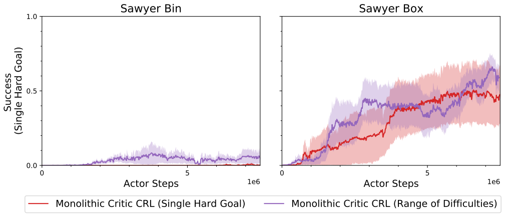


Figure 5: **The importance of representations.** Single goal exploration is less effective with using a monolithic critic architecture (as opposed to an inner product architecture), suggesting that the contrastive representations may drive exploration.

goal task. To test this hypothesis, we compared our method (which randomly samples different goals) with a variant that always uses the single hard goal in the actor loss. Note that this single hard goal is the one that is used for evaluation.The results in Fig. 4 show that when data are collected with a single goal, using a single goal in the actor loss degrades performance (purple vs. blue (ours)). When data are collected with multiple goals, using a single goal in the actor loss gives only a slight boost in performance (red vs. black). These results imply that the effectiveness of single-goal exploration is not explained by policy overfitting.

Representations are important. We also study how the representations might drive exploration. We replaced the inner product critic function ($\phi(s,a)^T\psi(s_f)$) with a monolithic critic function ($Q(s,a,s_f)$), which takes as input a concatenated array of the state, action, and goal.[3] The results, shown in Fig. 5 indicate that single-goal exploration is not effective with using a monolithic critic network. This experiment suggests that the contrastive representations $\phi(s,a)^T$ and $\psi(s_f)$ drive exploration, though the precise mechanism for *how* they drive exploration remains unclear.

4 Conclusion

In this paper, we showed that skills and directed exploration emerges from a straightforward RL algorithm: contrastive RL where every trajectory is collected by trying to reach a single fixed goal. There remain two outstanding questions raised by our experiments. *First*, we lack a clear understanding of why skills and directed exploration emerge. Our experiments provide some hints, yet much theoretical work remains to be done to understand what is driving the exploration. *Second*, how can we leverage the success of the proposed method to address exploration in other problem

[3] In this experiment only, we decreased the batch size from 256 to 32 for both methods, as otherwise we encountered out-of-memory errors for the monolithic method.

settings (e.g., if a reward function were given or if not even a single fixed goal were available). In this vein, we encourage future work to find ways of exploiting the emergent exploration properties that we have observed in the single goal setting.

Limitations. The primary limitation of our work is a lack of theoretical analysis explaining *why* skills and directed exploration emerge. Empirically, our experiments are focused primarily on manipulation tasks; we encourage future work to study applications to other settings.

Acknowledgments. Thanks to the members of the Princeton RL lab for feedback on preliminary versions of the work. We are also grateful to Elliot Chane-Sane for corresponding with us on the strengths and limitations of the RIS algorithm. The experiments in this paper were substantially performed using the Princeton Research Computing resources at Princeton University, which is a consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE) and Office of Information Technology's Research Computing.

References

[1] Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. (2017). Hindsight experience replay. *Advances in Neural Information Processing Systems*, 30.

[2] Asmuth, J., Li, L., Littman, M. L., Nouri, A., and Wingate, D. (2009). A bayesian sampling approach to exploration in reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, pages 19–26.

[3] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *International Conference on Machine Learning*, pages 41–48.

[4] Bougie, N. and Ichise, R. (2020). Skill-based curiosity for intrinsically motivated reinforcement learning. *Machine Learning*, 109:493–512.

[5] Campero, A., Raileanu, R., Kuttler, H., Tenenbaum, J. B., Rocktäschel, T., and Grefenstette, E. (2021). Learning with AMIGo: Adversarially motivated intrinsic goals.

[6] Chane-Sane, E., Schmid, C., and Laptev, I. (2021). Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning*, pages 1430–1440. PMLR.

[7] Chen, G., Peng, Y., and Zhang, M. (2018). Effective exploration for deep reinforcement learning via bootstrapped q-ensembles under tsallis entropy regularization. *arXiv preprint arXiv:1809.00403*.

[8] Chen, R. Y., Sidor, S., Abbeel, P., and Schulman, J. (2017). UCB exploration via q-ensembles. *arXiv preprint arXiv:1706.01502*.

[9] Conti, E., Madhavan, V., Petroski Such, F., Lehman, J., Stanley, K., and Clune, J. (2018). Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in Neural Information Processing Systems*, 31.

[10] Dann, C., Mohri, M., Zhang, T., and Zimmert, J. (2021). A provably efficient model-free posterior sampling method for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12040–12051.

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

[12] Ding, Y., Florensa, C., Abbeel, P., and Phielipp, M. (2019). Goal-conditioned imitation learning. *Advances in Neural Information Processing Systems*, 32.

[13] Dosovitskiy, A. and Koltun, V. (2016). Learning to act by predicting the future. In *International Conference on Learning Representations*.

[14] Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*.

[15] Eysenbach, B., Myers, V., Salakhutdinov, R., and Levine, S. (2024). Inference via interpolation: Contrastive representations provably enable planning and inference. *arXiv preprint arXiv:2403.04082*.

[16] Eysenbach, B., Salakhutdinov, R., and Levine, S. (2021). C-learning: Learning to achieve goals via recursive classification. In *International Conference on Learning Representations*.

[17] Eysenbach, B., Salakhutdinov, R. R., and Levine, S. (2019). Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in Neural Information Processing Systems*, 32.

[18] Eysenbach, B., Zhang, T., Levine, S., and Salakhutdinov, R. R. (2022). Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620.

[19] Fan, Y. and Ming, Y. (2021). Model-based reinforcement learning for continuous control with posterior sampling. In *International Conference on Machine Learning*, pages 3078–3087. PMLR.

[20] Florensa, C., Held, D., Geng, X., and Abbeel, P. (2018). Automatic goal generation for reinforcement learning agents. In *International Conference on Machine Learning*, pages 1515–1528. PMLR.

[21] Florensa, C., Held, D., Wulfmeier, M., Zhang, M., and Abbeel, P. (2017). Reverse curriculum generation for reinforcement learning. In *Conference on Robot Learning*, pages 482–495. PMLR.

[22] Fortunato, M., Azar, M. G., Piot, B., Menick, J., Hessel, M., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. (2018). Noisy networks for exploration. In *International Conference on Learning Representations*.

[23] Fujimoto, S., Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR.

[24] Ghosh, D., Gupta, A., Reddy, A., Fu, J., Devin, C. M., Eysenbach, B., and Levine, S. (2020). Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*.

[25] Guo, Z. D., Pires, B. A., Piot, B., Grill, J.-B., Altché, F., Munos, R., and Azar, M. G. (2020). Bootstrap latent-predictive representations for multitask reinforcement learning. In *International Conference on Machine Learning*, pages 3875–3886. PMLR.

[26] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR.

[27] Hansen-Estruch, P., Zhang, A., Nair, A., Yin, P., and Levine, S. (2022). Bisimulation makes analogies in goal-conditioned reinforcement learning. In *International Conference on Machine Learning*, pages 8407–8426. PMLR.

[28] Hartikainen, K., Geng, X., Haarnoja, T., and Levine, S. (2020). Dynamical distance learning for semi-supervised and unsupervised skill discovery. In *International Conference on Learning Representations*.

[29] Heess, N., Wayne, G., Silver, D., Lillicrap, T., Erez, T., and Tassa, Y. (2015). Learning continuous control policies by stochastic value gradients. *Advances in Neural Information Processing Systems*, 28.

[30] Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, 29.

[31] Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. (2020). Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR.

[32] Kaelbling, L. P. (1993). Learning to achieve goals. In *IJCAI*, volume 2, pages 1094–8. Citeseer.

[33] Kearns, M. and Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49:209–232.

[34] Li, J., Shi, X., Li, J., Zhang, X., and Wang, J. (2020). Random curiosity-driven exploration in deep reinforcement learning. *Neurocomputing*, 418:139–147.

[35] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

[36] Lin, X., Baweja, H. S., and Held, D. (2019). Reinforcement learning without ground-truth state. *arXiv preprint arXiv:1905.07866*.

[37] Liu, M., Zhu, M., and Zhang, W. (2022). Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*.

[38] Lynch, C., Khansari, M., Xiao, T., Kumar, V., Tompson, J., Levine, S., and Sermanet, P. (2020). Learning latent plans from play. In *Conference on Robot Learning*, pages 1113–1132. PMLR.

[39] Machado, M. C., Bellemare, M. G., and Bowling, M. (2020). Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5125–5133.

[40] Matiisen, T., Oliver, A., Cohen, T., and Schulman, J. (2019). Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740.

[41] McGovern, A. and Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density. In *International Conference on Machine Learning*, pages 361–368.

[42] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[43] Nachum, O., Gu, S. S., Lee, H., and Levine, S. (2018). Data-efficient hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 31.

[44] Nasiriany, S., Pong, V., Lin, S., and Levine, S. (2019). Planning with goal-conditioned policies. *Advances in Neural Information Processing Systems*, 32.

[45] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[46] OpenAI, O., Plappert, M., Sampedro, R., Xu, T., Akkaya, I., Kosaraju, V., Welinder, P., D'Sa, R., Petron, A., Pinto, H. P. d. O., et al. (2021). Asymmetric self-play for automatic goal discovery in robotic manipulation. *arXiv preprint arXiv:2101.04882*.

[47] Osband, I., Aslanides, J., and Cassirer, A. (2018). Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 31.

[48] Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016a). Deep exploration via bootstrapped DQN. *Advances in Neural Information Processing Systems*, 29.

[49] Osband, I., Van Roy, B., and Wen, Z. (2016b). Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pages 2377–2386. PMLR.

[50] O'Donoghue, B., Osband, I., Munos, R., and Mnih, V. (2018). The uncertainty bellman equation and exploration. In *International Conference on Machine Learning*, pages 3836–3845.

[51] Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pages 2778–2787. PMLR.

[52] Paul, S., Vanbaar, J., and Roy-Chowdhury, A. (2019). Learning from trajectories via subgoal discovery. *Advances in Neural Information Processing Systems*, 32.

[53] Pearce, T., Anastassacos, N., Zaki, M., and Neely, A. (2018). Bayesian inference with anchored ensembles of neural networks, and application to exploration in reinforcement learning. *arXiv preprint arXiv:1805.11324*.

[54] Plappert, M., Andrychowicz, M., Ray, A., McGrew, B., Baker, B., Powell, G., Schneider, J., Tobin, J., Chociej, M., Welinder, P., et al. (2018a). Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*.

[55] Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., and Andrychowicz, M. (2018b). Parameter space noise for exploration. In *International Conference on Learning Representations*.

[56] Pong, V., Dalal, M., Lin, S., Nair, A., Bahl, S., and Levine, S. (2020). Skew-fit: State-covering self-supervised reinforcement learning. In *International Conference on Machine Learning*, pages 7783–7792. PMLR.

[57] Rudner, T. G., Pong, V., McAllister, R., Gal, Y., and Levine, S. (2021). Outcome-driven reinforcement learning via variational inference. *Advances in Neural Information Processing Systems*, 34.

[58] Savinov, N., Dosovitskiy, A., and Koltun, V. (2018). Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*.

[59] Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). Universal value function approximators. In *International Conference on Machine Learning*, pages 1312–1320. PMLR.

[60] Schmeckpeper, K., Xie, A., Rybkin, O., Tian, S., Daniilidis, K., Levine, S., and Finn, C. (2020). Learning predictive models from observation and interaction. In *European Conference on Computer Vision*, pages 708–725. Springer.

[61] Schroecker, Y. and Isbell, C. (2020). Universal value density estimation for imitation learning and goal-conditioned reinforcement learning. *arXiv preprint arXiv:2002.06473*.

[62] Shah, D., Eysenbach, B., Rhinehart, N., and Levine, S. (2022). Rapid exploration for open-world navigation with latent goal models. In *Conference on Robot Learning*, pages 674–684. PMLR.

[63] Srinivas, A., Jabri, A., Abbeel, P., Levine, S., and Finn, C. (2018). Universal planning networks: Learning generalizable representations for visuomotor control. In *International Conference on Machine Learning*, pages 4732–4741. PMLR.

[64] Stadie, B. C., Levine, S., and Abbeel, P. (2015). Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*.

[65] Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R. (2018). Intrinsic motivation and automatic curricula via asymmetric self-play. In *International Conference on Learning Representations*.

[66] Sun, H., Li, Z., Liu, X., Zhou, B., and Lin, D. (2019). Policy continuation with hindsight inverse dynamics. *Advances in Neural Information Processing Systems*, 32:10265–10275.

[67] Syed, U., Bowling, M., and Schapire, R. E. (2008). Apprenticeship learning using linear programming. In *Proceedings of the International Conference on Machine Learning*, pages 1032–1039.

[68] Tang, H., Houthooft, R., Foote, D., Stooke, A., Xi Chen, O., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. (2017). # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 30.

[69] Thrun, S. B. (1992). *Efficient exploration in reinforcement learning*. Carnegie Mellon University.

[70] Tian, S., Nair, S., Ebert, F., Dasari, S., Eysenbach, B., Finn, C., and Levine, S. (2021). Model-based visual planning with self-supervised functional distances. In *International Conference on Learning Representations*.

[71] Tokic, M. (2010). Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *Annual Conference on Artificial Intelligence*, pages 203–210. Springer.

[72] Tosatto, S., D'Eramo, C., Pajarinen, J., Restelli, M., and Peters, J. (2019). Exploration driven by an optimistic bellman equation. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[73] Trott, A., Zheng, S., Xiong, C., and Socher, R. (2019). Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. *Advances in Neural Information Processing Systems*, 32.

[74] Venkattaramanujam, S., Crawford, E., Doan, T., and Precup, D. (2019). Self-supervised learning of distance functions for goal-conditioned reinforcement learning. *arXiv preprint arXiv:1907.02998*.

[75] Wu, Y., Tucker, G., and Nachum, O. (2019). The laplacian in RL: Learning representations with efficient approximations. In *International Conference on Learning Representations*.

[76] Yao, Y., Xiao, L., An, Z., Zhang, W., and Luo, D. (2021). Sample efficient reinforcement learning via model-ensemble exploration and exploitation. In *International Conference on Robotics and Automation (ICRA)*, pages 4202–4208.

[77] Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. (2020). Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR.

[78] Zhang, T., Eysenbach, B., Salakhutdinov, R., Levine, S., and Gonzalez, J. E. (2022). C-planning: An automatic curriculum for learning goal-reaching tasks. In *International Conference on Learning Representations*.

[79] Zhang, Z., Li, Y., Bastani, O., Gupta, A., Jayaraman, D., Ma, Y. J., and Weihs, L. (2024). Universal visual decomposer: Long-horizon manipulation made easy. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6973–6980. IEEE.

[80] Zheng, C., Salakhutdinov, R., and Eysenbach, B. (2024). Contrastive difference predictive coding. In *International Conference on Learning Representations*.

A Related Work

Our work builds on a long line of prior work in exploration methods for reinforcement learning [2, 31, 33, 41, 64, 68, 69, 71], and will study this problem in the specific setting of goal-conditioned RL (GCRL) [1, 12, 13, 16, 17, 24, 32, 36, 38, 43, 44, 57, 58, 58–60, 63, 66]. This section reviews three types of strategies for exploration. Our proposed method aims to lift the limitations associated with these prior methods.

Rewards and demonstrations. One of the key challenges with GCRL is the sparsity of the learning problem, so many prior GCRL methods assume access to a dense, hand-crafted reward function [54, 59] or a distance metric [28, 70, 73, 75]. Other methods attempt to make GCRL more tractable by using expert demonstrations [12, 52] to guide learning and planning. Although well-designed reward functions and expert demonstrations are useful for training, these components add complexity, and collecting demonstrations can be challenging. Our method builds upon a growing collection of GCRL algorithms that require neither a reward function nor demonstrations [16, 18, 24, 36, 66, 80] – specifically, we consider a variant of GCRL where only a single goal is provided for training and evaluation. As such, we could treat it as a single-task problem, but we find that treating it as a multi-task problem is crucial to achieving good performance.

Exploration and subgoal sampling. Without a dense reward function or expert demonstrations, the primary challenge of GCRL is effective exploration. One class of exploration strategies adds noise to the actions [23, 26, 29, 35] or policies [22, 55]. While these methods are simple to implement, they typically fail to perform directed exploration [48]. A second class of methods formulate an intrinsic exploration reward [4, 9, 14, 34, 39, 51], which the agent aims to optimize in addition to whatever rewards (dense or sparse) that are provided by the environment. While these methods can work effectively, they can be challenging to scale to high-dimensional and long-horizon tasks. A third class of methods use probabilistic techniques, including ensembles [7, 8, 48, 53, 76], posterior sampling [10, 19, 47, 49], and uncertainty propagation [50, 72] – these methods can excel at directed exploration, though challenges include tuning the prior and dealing with large ensembles. A fourth class of methods modifies the goals that are used in training. For example, some methods automatically propose subgoals, breaking down a hard task into a sequence of easier tasks [6, 58, 62, 78, 79]. Other methods automatically adjust the goal distribution [20, 56, 74] or initial state distribution [21], so that the difficulty of learning increases throughout training. Despite excellent results in certain settings, scaling these methods beyond 2D navigation remains challenging, and the algorithms remain complex. We compare against one prototypical subgoal sampling method (RIS [6]).

Multi-task learning for single-task problems The last strategy for exploration is so ubiquitous it is easy to forget: training on multiple related tasks, even when we only care about performance on a single difficult task. For example, many prior GCRL methods command a range of goals during exploration. Intuitively, the easy tasks can be learned with little exploration, and learning those tasks should enable the agent to solve more challenging tasks (similar to curriculum learning [3, 5, 40]). However, actually constructing these multiple training tasks or goals requires additional human supervision: the human often lays out a "trail of breadcrumbs", and the agent learns how to navigate to each [18]. Our paper studies the setting where only a single goal is provided for training, yet our experiments will compare against baselines that have access to training goals with a range of difficulties.

B Environment dynamics are visible in the norms of contrastive representations

Approach. To further investigate our predictions about targeted exploration and robustness arising from the building of rich representations early in training, we visualized the norms of the contrastive goal encoder $||\psi(s_g)||_2^2$ at an early checkpoint, shown in Figure 6. We target goal encoder norms since we observe that mean goal encoder norms over the training distribution rollout positions closely correlate with training loss, and hypothesize that these norms reflect environment-learning.

Specifically, we fix the end-effector distance to the target distance for gripping, uniformly randomly sample many states x_i corresponding to both the agent end-effector and object being at x_i, and plot the corresponding values $||\psi(x_i)||_2^2$ from an early (pre-first-success) encoder checkpoint on the `sawyer bin` task.

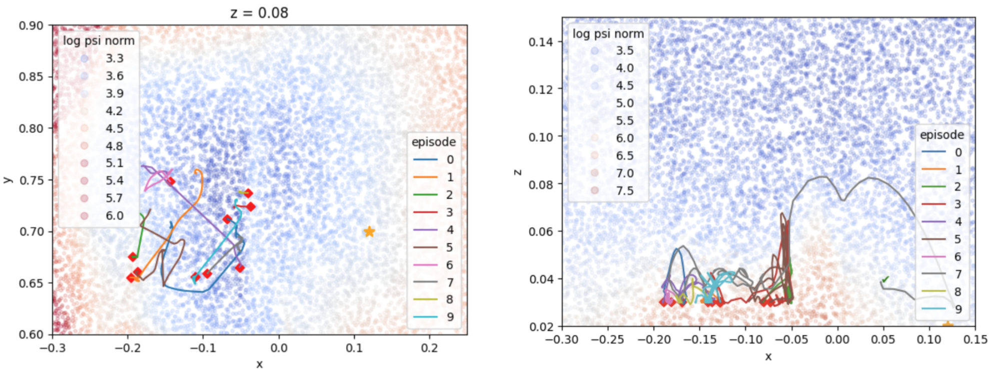


Figure 6: **Contrastive representations capture environment dynamics (impassable bin walls and floor) in an interpretable way**. We plot the log of the goal encoder norms, $\log(||\psi(s_g)||_2^2)$, for various observations where the end effector and block are at the same location. We show a top-down view at box-height (left) and side view (right). We find that the impassable bin wall is visible both from the top view (thin strip of lighter blue) and side view (vertical spike of red), represented by relatively higher norms. Example policy rollouts (colored lines) from checkpoints at the same stage in training are overlaid for reference, with their starting positions marked as red diamonds and the goal marked as a gold star.

Results. As shown in Figure 6, we find interpretable patterns corresponding to a map of environment dynamics, such as high goal encoder norms at the location of the impassable bin walls and bin bottom. We hypothesize that these strongly-represented environment features contribute to the development of higher-level skills and ultimately behaviors such as perturbation robustness.

C Experimental Details

Table 1: Hyperparameters for our method and the baselines.

hyperparameter	value
Contrastive RL (CRL) [18]	
batch size	256
learning rate	3e-4
discount	0.99
actor target entropy	0
hidden layers sizes (policy, critic, representations)	(256, 256)
initial random data collection	10,000 transitions
replay buffer size	1e6
samples per insert[1]	256
representation dimension (dim($\phi(s, a)$), dim($\psi(s_g)$))	64
actor minimum std dev	1e-6
SAC [26]	
batch size	256
learning rate	3e-4
discount	0.99
hidden layers sizes (policy, critic)	(256, 256)
target EMA term	5e-3
initial random data collection	10,000 transitions
replay buffer size	1e6
samples per insert[1]	256
actor minimum std dev	1e-6
RIS [6]	
batch size	256
learning rate	1e-3 (critic), 1e-4 (policy)
high-level policy learning rate	1e-4
discount	0.99
hidden layers sizes (policy, high-level policy, critic)	(256, 256)
initial random data collection	10,000 transitions
replay buffer size	1e5
Polyak coefficient for target networks	5e-3
valid state KL constraint (ϵ)	1e-4
subgoal KL penalty (α)	0.1
high-level policy weight regularization (λ)	0.1

[1] How many times is each transition used for training before being discarded.

[2] We collect N transitions, add them to the buffer, and then do N gradient steps using the experience sampled randomly from the buffer.

Table 2: **Baselines:** Assumptions for methods used in the experiments below.

		Requirements		
Algorithm	Exploration	Exploration Goals	Dense Rewards	Distance Threshold
Contrastive RL [18]	single goal (ours)	✗	✗	✗
	multiple goals	✓	✗	✗
SAC [26] (sparse rewards)	single goal	✗	✗	✓
SAC [26] (dense rewards)	single goal	✗	✓	✓
SAC [26] (sparse rewards) + HER [1]	single goal	✗	✗	✓
RIS [6]	single goal	✗	✗	✓

D Investigating Exploration in Single-Goal Contrastive RL

In this section we describe the stages of skill development exhibited by the Single-Goal Contrastive RL agent.

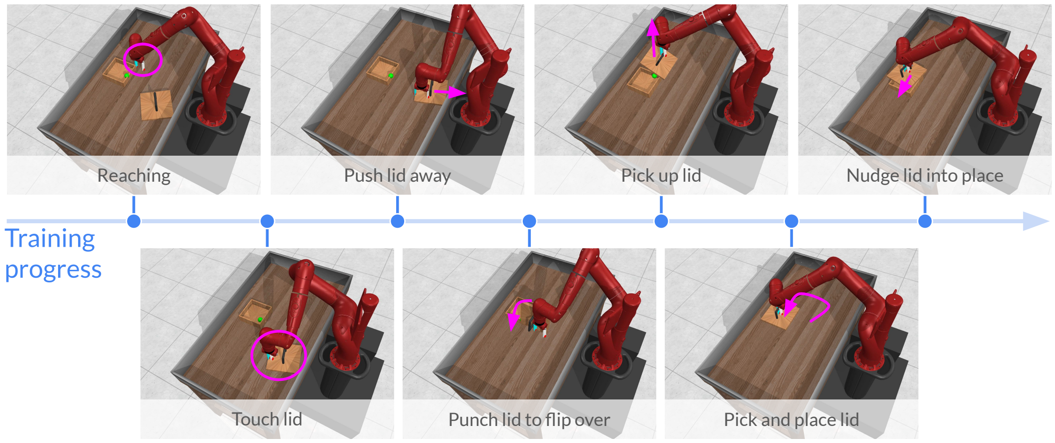


Figure 7: **Skills and Directed Exploration for Putting a Lid on a Box:** This manipulation task contains an open box and a lid. The single fixed goal has the lid placed neatly on top of the center of the box. The images above show skills acquired throughout the course of learning. Note that some skills unlock subsequent skills (e.g., reaching is a prerequisite for picking, which is a prerequisite for placing) while others look like open-ended "play" (flipping the lid over, pushing the lid away from the box).

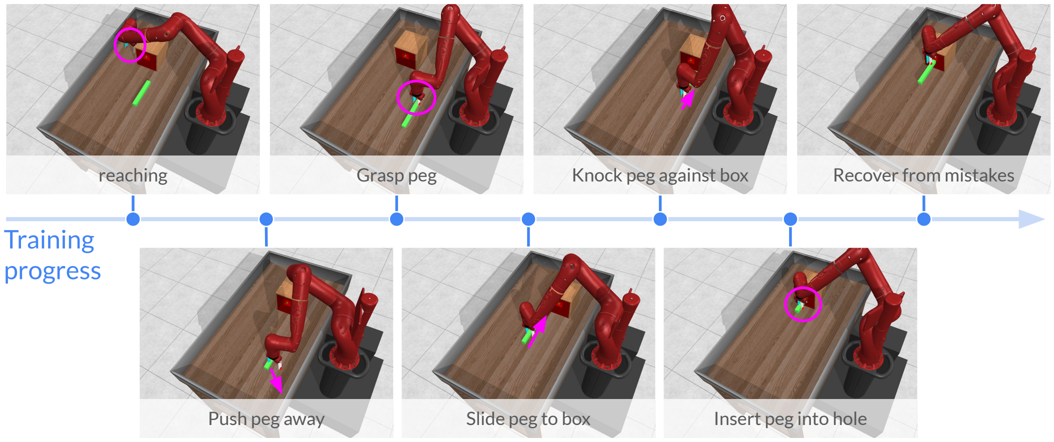


Figure 8: **Skills and Directed Exploration for Peg Insertion:** This manipulation task contains a peg and box with a narrow hole; the single fixed goal is a state where the peg is inside the hole. The agent acquires a sequence of increasingly complex skills throughout training, some of which are important for solving the task (e.g., reaching, grasping) while others are more "playful" (e.g., knocking the peg against the box). The agent also learns to recover from mistakes (see Fig. 11).

Early training: agent develops an emergent curriculum of skills. Not only does single-goal CRL consistently achieve high success rates on manipulation tasks, but it also demonstrates complex and directed exploration techniques during early training. To observe the agent's behavior throughout training, we saved learning checkpoints at fixed intervals and visualized the agent's behavior at each checkpoint (see Figures 1, 7 and 8). We found that the agent learns simple skills before complex ones. For example, in all three environments, the agent *(1)* first learns how to move its end-effector to varying locations, *(2)* then learns how to nudge and slide the object, and *(3)* finally learns to pick up and direct the object. We observe a wide array of exploratory behavior in early training that is not directly connected to the goal: from punching the box lid to flip it over (Fig. 7) to pushing the block far away in a random direction (Fig. 1).

First successes: agent trades off exploration for exploitation. As the agent learns to reach the goal more consistently, the agent's behavior becomes less exploratory, qualitatively similar to UCB exploration [10, 19, 25, 47, 49]. To quantify exploration, we discretized the state space of the robotic manipulation environments and recorded the cumulative number of unique positions visited throughout training. Fig. 9 shows that the growth rate of this exploration metric decreases as the success rate increases (compare with Fig. 2). For the `sawyer box` and `sawyer peg` environments,

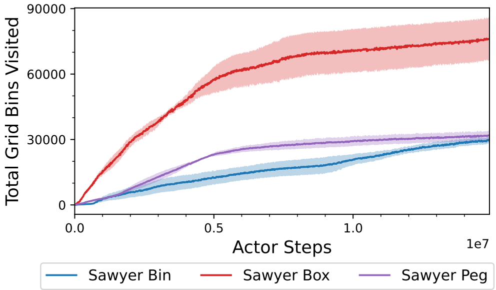
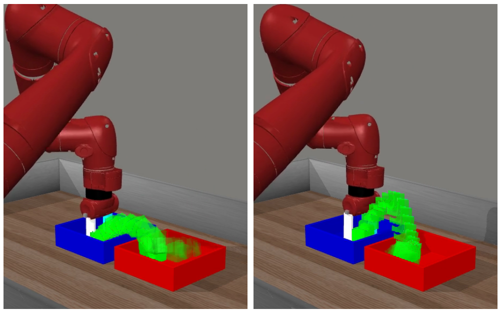


Figure 9: **Quantifying Exploration.** We analyze "single hard goal" exploration by discretizing the object's XY position and counting the cumulative number of unique positions visited throughout training. Exploration starts to plateau after the agent can successfully reach the goal (compare with Fig. 2).



(a) "Push and flick" (seed = 3)

(b) "gently pick" (seed = 4)

Figure 10: **Different random seeds learn different strategies.** Other seeds learn a policy whose strategy depends on the initial position of the block.

the agent achieves a high success rate earlier in training, which corresponds to the earlier plateau of the unique grid cells curve. For the `sawyer bin` environment, the agent takes longer to reach high success, and the exploration metric does not start leveling out until the end of training. This trend highlights how single-goal CRL develops a self-directed exploration strategy that automatically trades off between exploration and exploitation.

Consistent successes: agent finds diverse paths to the goal. Not only is the performance of single-goal CRL reproducible (all random seeds solve the manipulation tasks), but policies trained with different random seeds learn qualitatively different goal-reaching strategies. For example, Fig. 10 shows the strategies learned by different random seeds on the `sawyer bin` task. One seed consistently moves the block flush against the wall of the red bin and flicks it into the blue bin. Another seed tends to grasp the block, lift it, and gently drop it in the blue bin. A third seed chooses between these strategies depending on whether the block starts near the wall or away from the wall. Without explicit human guidance in the form of rewards, demonstrations, or subgoals, we see multiple creative and divergent strategies emerge for solving the same problem.

Further training: agent develops robustness and self-recovery. During later stages of training, we observe that the agent demonstrates robustness and learns to recover from mistakes. For example, in the `sawyer peg` environment, when the agent drops the peg, it is able to recover by bending down and grasping the peg again. To quantify robustness, we ran perturbation experiments in the `sawyer peg` and `sawyer box` environments, in which we randomly perturbed the target object's location between 0 and 0.05 meters along each of the three axes. We tested two settings: *(1)* perturbation at the start of the episode ("static perturbations") and *(2)* in the middle of an episode ("dynamic perturbations": $t = 20$ for `sawyer box` and $t = 50$ for `sawyer peg`). As shown in Fig. 11, single goal exploration is robust to static perturbations and somewhat robust to dynamic perturbations, notably outperforming multi-goal CRL in three out of four scenarios. We hypothesize that this is also the result of better exploration, which leads to learned representations that generalize better across unusual or unseen states.

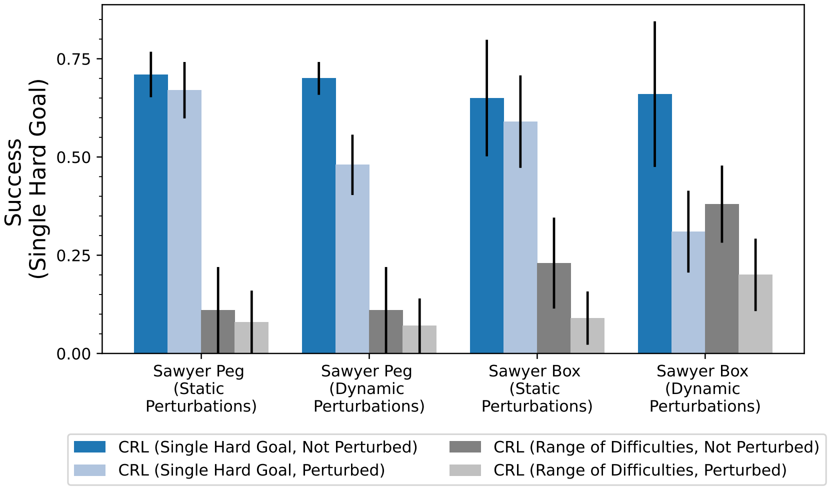


Figure 11: **Robustness to perturbations:** Single (Hard) Goal exploration results in policies that are more robust to environment perturbations, as compared to policies trained with goals ranging in difficulty. The success rate remains high even when the object is perturbed at the start ("static") or in the middle ("dynamic") of an episode, likely because its effective exploration means that it has seen a wide range of states during training.