Stabilizing Contrastive RL: Techniques for Robotic Goal Reaching from Offline Data

Chongyi Zheng[1] **Benjamin Eysenbach**[1] **Homer Walke**[2] **Patrick Yin**[2] **Kuan Fang**[2]
Ruslan Salakhutdinov[1] **Sergey Levine**[2]

[1]Carnegie Mellon University [2]UC Berkeley
chongyiz@andrew.cmu.edu

Abstract

Robotic systems that rely primarily on self-supervised learning have the potential to decrease the amount of human annotation and engineering effort required to learn control strategies. In the same way that prior robotic systems have leveraged self-supervised techniques from computer vision (CV) and natural language processing (NLP), our work builds on prior work showing that the reinforcement learning (RL) itself can be cast as a self-supervised problem: learning to reach any goal without human-specified rewards or labels. Despite the seeming appeal, little (if any) prior work has demonstrated how self-supervised RL methods can be practically deployed on robotic systems. By first studying a challenging simulated version of this task, we discover design decisions about architectures and hyperparameters that increase the success rate by $2\times$. These findings lay the groundwork for our main result: we demonstrate that a self-supervised RL algorithm based on contrastive learning can solve real-world, image-based robotic manipulation tasks, with tasks being specified by a single goal image provided after training.

1 Introduction

Self-supervised learning serves as the bedrock for many NLP and computer vision applications, leveraging unlabeled data to acquire good representations for downstream tasks. How might we enable similar capabilities for robot learning algorithms? In NLP and computer vision, self-supervised learning is typically done via one objective (often denoising), while the downstream tasks use a different objective (e.g., linear regression). In the RL setting, prior work has shown how a (self-supervised) contrastive learning objective can simultaneously be used to learn (1) compact representations, (2) a goal-conditioned policy, and (3) a corresponding value function [17]. From a robotics perspective, this framing is appealing because users need not manually specify these components (e.g., no tuning hyperparameters in a reward function). However, to the best of our knowledge these contrastive RL methods have not been applied on real-world robotic systems.

The aim of this paper is to use self-supervised RL methods that solve real-world goal-reaching tasks. We will focus on a class of prior RL methods based on *contrastive RL* [64, 17, 15, 23]. With an eye towards enabling real-world deployment, we start by studying how various design decisions can stabilize these contrastive RL methods, focusing both on their *learned representations* and their *learned policies*. Through careful experiments, we find a set of design decisions regarding model capacity and regularization that boost performance by $+45\%$ over prior implementations of contrastive RL, and by $2\times$ relative to alternative goal-conditioned RL methods. We call our implementation **stable contrastive RL**. The key contribution is our real-world experiments, where we demonstrate that these design decisions enable image-based robotic manipulation. Additional experiments reveal an intriguing property of the learned representations: linear interpolation seems to correspond to planning (see Fig. 2).

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Contrastive RL and Design Decisions

In this section we introduce notations and the goal-conditioned RL objective, and then revise a prior algorithm that we will use in our experiments, mentioning important design decisions.

2.1 Preliminaries

We assume a goal-conditioned controlled Markov process (an MDP without a reward function) defined by states $s_t \in \mathcal{S}$, goals $g \in \mathcal{S}$, actions a_t, initial state distribution $p_0(s_0)$ and dynamics $p(s_{t+1} \mid s_t, a_t)$. We will learn a goal-conditioned policy $\pi(a \mid s, g)$, where the state s and the goal g are both RGB images. Given a policy $\pi(a \mid s, g)$, we define $\mathbb{P}^{\pi(\cdot|\cdot,g)}(s_t = s \mid s_0, a_0)$ as the probability density of reaching state s after exactly t steps, starting at state s_0 and action a_0. The discounted state occupancy measure is a geometrically-weighted average of these densities:

$$p^{\pi(\cdot|\cdot,g)}(s_{t+} = s \mid s_0, a_0) \triangleq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}^{\pi(\cdot|\cdot,g)}(s_t = s \mid s_0, a_0). \tag{1}$$

The policy is learned by maximizing the likelihood of reaching the desired goal state under this discounted state occupancy measure [15, 64, 55, 17]:

$$\mathbb{E}_{p_g(g)}[p^{\pi(\cdot|\cdot,g)}(s_{t+} = g)] = \mathbb{E}_{p_g(g)p_0(s_0)\pi(a_0|s_0,g)}\left[p^{\pi(\cdot|\cdot,g)}(s_{t+} \mid s_0, a_0)\right].$$

Following prior work [17], we estimate this objective via contrastive representation learning [24]; we will learn a critic function that takes a state-action pair (s, a) and a future state s_{t+} as input, and outputs a real number $f(s, a, s_{t+})$ estimating the (ratio of) likelihood of reaching the future state, given the current state and action. We will parameterize the critic function as an inner product between the state-action representation $\phi(s, a)$ and the future state representation $\psi(s_{t+})$, $f(s, a, s_{t+}) = \phi(s, a)^\top \psi(s_{t+})$, interpreting the critic value as the similarity between those representations.

Contrastive RL distinguishes a future state sampled from the average discounted state occupancy measure, $s_f^+ \sim p^{\pi(\cdot|\cdot)}(s_{t+} \mid s, a) = \int p^{\pi(\cdot|\cdot,g)}(s_{t+} \mid s, a)p^\pi(g \mid s, a)dg$, from a future state sampled from a arbitrary state-action pair, $s_f^- \sim p(s_{t+}) = \int p^{\pi(\cdot|\cdot)}(s_{t+} \mid s, a)p(s, a)dsda$, using the NCE-Binary [24, 41, 31] objective:

$$\mathbb{E}_{s_f^+ \sim p^{\pi(\cdot|\cdot)}(s_{t+}|s,a)}[\underbrace{\log \sigma(\phi(s, a)^\top \psi(s_f^+))}_{\mathcal{L}_1(\phi(s,a),\psi(s_f^+))}] + \mathbb{E}_{s_f^- \sim p(s_{t+})}[\underbrace{\log(1 - \sigma(\phi(s, a)^\top \psi(s_f^-)))}_{\mathcal{L}_2(\phi(s,a),\psi(s_f^-))}]. \tag{2}$$

This objective can also be formulated in an off-policy manner (see Eysenbach et al. [15] and Appendix A); we will use this off-policy version in our experiments. Appendix B provides some intuition about a connection between contrastive RL and hard negative mining.

2.2 Design Decisions for Stabilizing Contrastive RL

In this section, we describe the most important design factors to stabilize contrastive RL: *(i)* using appropriate encoder architecture and batch size, *(ii)* stabilizing and speeding up training with layer normalization and cold initialization, and *(iii)* combating overfitting with data augmentation.

Neural network architecture for image inputs. In vision and language domains, scaling the architecture size has enabled large neural networks to achieve ever higher performance on ever larger datasets [61, 6, 11, 53, 4]. While large vision models (e.g., ResNets [30] and Transformers [66]) have been adopted to the RL setting [14, 58, 35, 7, 32], shallow CNNs remain pervasive in RL [57, 37, 36, 3, 26, 25], suggesting that simple architectures might be sufficient. In our experiments, we will study two aspects of the architecture: the visual feature extractor, and the contrastive representations learned on top of these visual features. For the visual feature extractor, we will compare a simple *CNN* versus a much larger *ResNet*. For the contrastive representations, we will study the effect of scaling the width and depth of these MLPs. While prior methods often train the visual feature extractor separately from the subsequent MLP [53], our experiments will also study end-to-end training approaches. Appendix C includes an overview of our network architecture.

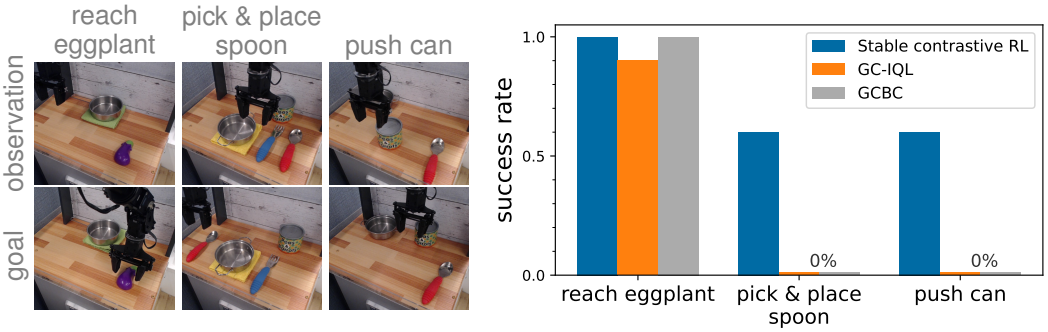


Figure 1: **Evaluation on real manipulation tasks.** Stable contrastive RL matches or outperforms GC-IQL and GCBC on all manipulation tasks. Each success rate is calculated from taking 10 rollouts.

Batch size. Prior works in computer vision [8, 29, 22, 9] have found that contrastive representation learning benefits from large batch sizes, which can stabilize and accelerate learning. In the context of contrastive RL, using larger batches increases not only the number of positive examples (linear in batch size) but also the number of negative examples (quadratic in batch size). This means that algorithm would be able to see more random goals s_f^- given a future goal s_f^+. We will study how these growing positive and negative examples as a result of growing batch sizes will affect contrastive RL.

Layer normalization. We conjecture that learning from a *diverse* offline dataset, containing examples of various manipulation behaviors, may result in features and gradients that are different for different subsets of the dataset [72]. We will study whether adding layer normalization [2] to the visual feature extractor and the subsequent MLP can boost performance, following prior empirical studies on RL [5, 35]. We will experiment with applying layer normalization to every layer of both CNN and MLP, before the non-linear activation function.

Cold initialization. Prior work has proposed that the alignment between positive examples is crucial to contrastive representation learning [67]. To encourage the alignment of representations during the initial training phase, we will weight initialization of the final feed-forward layer. Precisely, we will initialize the weights in the final layers of $\phi(s, a)$ and $\psi(g)$ using $\textsc{Unif}[-10^{-12}, 10^{-12}]$, resulting in representations that remain close to one another during the initial stages of learning. We will compare this "cold initialization" approach to a more standard initialization strategy, $\textsc{Unif}[-10^{-4}, 10^{-4}]$.

Data augmentation. Following prior work [17, 20, 4, 51, 59, 44, 68], we will augment the actor objective to include an additional behavioral cloning regularizer, which penalizes the actor for sampling out-of-distribution actions. While most of our design decisions increase the model capacity, adding this behavioral cloning acts as a sort of regularization often important in the offline RL setting to avoid overfitting [20]. Initial experiments found that this regularizer itself was prone to overfitting, motivating us to investigate data augmentation (random cropping), similar to prior work in offline RL [70, 36, 27].

3 Experiments

We study design decisions that drive stable contrastive RL and use simulated and real-world benchmarks to compare contrastive RL to other offline goal-conditioned policy learning methods. We also analyzes properties of the representations learned by stable contrastive RL. Our findings include:

Ablation study. Our experiments suggest the following design decisions stabilize contrastive RL: *(i)* Using a simple 3-layer CNN visual feature extractor followed by a wide MLP. *(ii)* Add layer normalization. *(iii)* Initialize last-layer weights of the MLP with small values, $\textsc{Unif}[-10^{-12}, 10^{-12}]$. *(iv)* Apply random cropping to both the state and goal images. *(v)* Use a large batch size (2048). We use **stable contrastive RL** to denote this combination of design decisions. See Appendix D.2 for details.

Comparing to prior methods. Stable contrastive RL matches or surpasses all baselines on four of the five simulated manipulation tasks (Fig. 5). Stable contrastive RL also outperforms all three baselines on locomotion benchmarks, achieving a 33% higher (relative) average success rate than the strongest baseline. Additionally, we study whether our method can effectively solve real world, image-based manipulation tasks, learning entirely from offline data. Stable contrastive RL performs

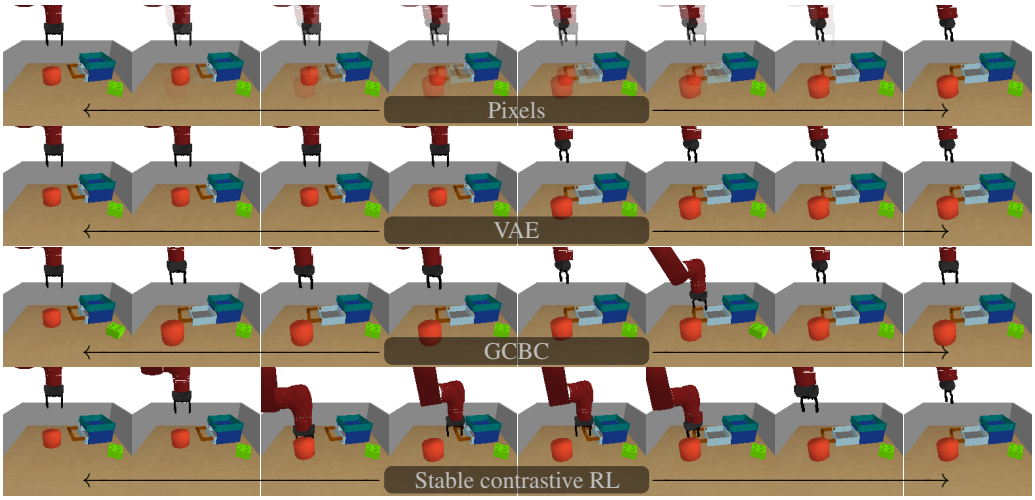


Figure 2: **Visualizing the representations.** *(Row 1)* Directly interpolating between two images in pixel space results in unrealistic images. *(Row 2)* Using a VAE, we interpolate between the representations of the left-most and right-most images, visualizing the nearest-neighbor retrievals from a validation set. The VAE captures the contents of the images but not the causal relationships – the object moves without being touched by the robot arm. *(Row 3)* GCBC also produces realistic images while ignoring temporal causality. *(Row 4)* Stable contrastive RL learns representations that capture not only the content of the images, but also the causal relationships – the arm first moves *away* from its position in the goal state so that it can move the object into place.

similarly to baselines on the simple (`reach eggplant`) task, while achieving a 60% success rate on the two challenging tasks (`pick & place spoon`, `push can`), where all baselines fail to make progress (Fig. 1). See Appendix D.3 for details.

Latent interpolation. When interpolating the representations of stable contrastive RL, the intermediate representations correspond to sequence of observations that the policy should visit when transiting between the initial observation and the final goal (Fig. 2). See Appendix D.4 for details.

The arm matching problem. Comparing the Q values for an observation where the object is not in the correct location, but the gripper is in the correct position, stable contrastive RL predicts a small value for this observation, highlighting the accuracy of its Q function, while GC-IQL mistakenly assigns high Q values to this observation. See Appendix D.5 for details.

Generalizing to unseen camera angles and object colors Because stable contrastive RL learns features solely with the critic objective, we expect that the features will not retain task-irrelevant information. We run an experiment comparing the generalization of stable contrastive RL against a baseline that uses features pre-trained via reconstruction on unseen environments with unseen camera angles and object colors. See Appendix E.10 for details.

Additional experiments. Appendix D.6 studies the influence of dataset size on the performance of stable contrastive RL. In Appendix E.11 & E.12 we study the effect of augmenting stable contrastive RL with auxiliary objectives and sub-goal planning. These components are critical for PTP but decrease performance of stable contrastive RL.

4 Conclusion

In this paper, we have studied design decisions that enabled a self-supervised RL method to solve real-world robotic manipulation tasks that stymie prior methods. We have found that decisions regarding the architecture, batch size, normalization, initialization, and augmentation are all important.

References

[1] Alakuijala, M., Dulac-Arnold, G., Mairal, J., Ponce, J., and Schmid, C. (2022). Learning reward functions for robotic manipulation by observing humans. *arXiv preprint arXiv:2211.09019*.

[2] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

[3] Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., and Blundell, C. (2020). Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning*, pages 507–517. PMLR.

[4] Baker, B., Akkaya, I., Zhokov, P., Huizinga, J., Tang, J., Ecoffet, A., Houghton, B., Sampedro, R., and Clune, J. (2022). Video pretraining (vpt): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems*.

[5] Bjorck, N., Gomes, C. P., and Weinberger, K. Q. (2021). Towards deeper deep reinforcement learning with spectral normalization. *Advances in Neural Information Processing Systems*, 34:8242–8255.

[6] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[7] Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. (2021a). Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

[8] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

[9] Chen, X., Xie, S., and He, K. (2021b). An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649.

[10] Ding, Y., Florensa, C., Abbeel, P., and Phielipp, M. (2019). Goal-conditioned imitation learning. *Advances in neural information processing systems*, 32.

[11] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[12] Ebert, F., Yang, Y., Schmeckpeper, K., Bucher, B., Georgakis, G., Daniilidis, K., Finn, C., and Levine, S. (2021). Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*.

[13] Emmons, S., Eysenbach, B., Kostrikov, I., and Levine, S. (2021). Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*.

[14] Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. (2018). Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pages 1407–1416. PMLR.

[15] Eysenbach, B., Salakhutdinov, R., and Levine, S. (2020). C-learning: Learning to achieve goals via recursive classification. In *International Conference on Learning Representations*.

[16] 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.

[17] Eysenbach, B., Zhang, T., Levine, S., and Salakhutdinov, R. (2022). Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*.

[18] Fang, K., Yin, P., Nair, A., and Levine, S. (2022a). Planning to practice: Efficient online fine-tuning by composing goals in latent space. *arXiv preprint arXiv:2205.08129*.

[19] Fang, K., Yin, P., Nair, A., Walke, H., Yan, G., and Levine, S. (2022b). Generalization with lossy affordances: Leveraging broad offline data for learning visuomotor tasks. *arXiv preprint arXiv:2210.06601*.

[20] Fujimoto, S. and Gu, S. S. (2021). A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145.

[21] 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*.

[22] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

[23] Guo, Z. D., Azar, M. G., Piot, B., Pires, B. A., and Munos, R. (2018). Neural predictive belief representations. *arXiv preprint arXiv:1811.06407*.

[24] Gutmann, M. U. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2).

[25] Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. (2019). Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*.

[26] Hafner, D., Lillicrap, T. P., Norouzi, M., and Ba, J. (2020). Mastering atari with discrete world models. In *International Conference on Learning Representations*.

[27] Hansen, N., Su, H., and Wang, X. (2021). Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in Neural Information Processing Systems*, 34:3680–3693.

[28] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

[29] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

[30] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[31] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

[32] Janner, M., Li, Q., and Levine, S. (2021). Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286.

[33] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR (Poster)*.

[34] Kostrikov, I., Nair, A., and Levine, S. (2021). Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.

[35] Kumar, A., Agarwal, R., Geng, X., Tucker, G., and Levine, S. (2022). Offline q-learning on diverse multi-task data both scales and generalizes. In *Deep Reinforcement Learning Workshop NeurIPS 2022*.

[36] Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. (2020a). Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895.

[37] Laskin, M., Srinivas, A., and Abbeel, P. (2020b). Curl: contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5639–5650.

[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] Ma, J. Y., Yan, J., Jayaraman, D., and Bastani, O. (2022a). Offline goal-conditioned reinforcement learning via f-advantage regression. *Advances in Neural Information Processing Systems*, 35:310–323.

[40] Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. (2022b). Vip: Towards universal visual reward and representation via value-implicit pre-training. In *Deep Reinforcement Learning Workshop NeurIPS 2022*.

[41] Ma, Z. and Collins, M. (2018). Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *EMNLP*.

[42] Mendonca, R., Rybkin, O., Daniilidis, K., Hafner, D., and Pathak, D. (2021). Discovering and achieving goals via world models. *Advances in Neural Information Processing Systems*, 34:24379–24391.

[43] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

[44] Nair, A., Dalal, M., Gupta, A., and Levine, S. (2020). Accelerating online reinforcement learning with offline datasets.

[45] Nair, A. V., Pong, V., Dalal, M., Bahl, S., Lin, S., and Levine, S. (2018). Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31.

[46] Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. (2022). R3m: A universal visual representation for robot manipulation. In *6th Annual Conference on Robot Learning*.

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

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

[49] Paster, K., McIlraith, S. A., and Ba, J. (2020). Planning from pixels using inverse dynamics models. *arXiv preprint arXiv:2012.02419*.

[50] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

[51] Peng, X. B., Kumar, A., Zhang, G., and Levine, S. (2019). Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*.

[52] 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.

[53] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

[54] Rosete-Beas, E., Mees, O., Kalweit, G., Boedecker, J., and Burgard, W. (2022). Latent plans for task-agnostic offline reinforcement learning. *arXiv preprint arXiv:2209.08959*.

[55] 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*, 34:13045–13058.

[56] Savinov, N., Dosovitskiy, A., and Koltun, V. (2018). Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*.

[57] Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609.

[58] Shah, R. M. and Kumar, V. (2021). Rrl: Resnet as representation for reinforcement learning. In *International Conference on Machine Learning*, pages 9465–9476. PMLR.

[59] Siegel, N., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., Hafner, R., Heess, N., and Riedmiller, M. (2019). Keep doing what worked: Behavior modelling priors for offline reinforcement learning. In *International Conference on Learning Representations*.

[60] Srivastava, R. K., Shyam, P., Mutz, F., Jaśkowski, W., and Schmidhuber, J. (2019). Training agents using upside-down reinforcement learning. *arXiv preprint arXiv:1912.02877*.

[61] Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

[62] Tian, S., Nair, S., Ebert, F., Dasari, S., Eysenbach, B., Finn, C., and Levine, S. (2020a). Model-based visual planning with self-supervised functional distances. *arXiv preprint arXiv:2012.15373*.

[63] Tian, Y., Krishnan, D., and Isola, P. (2020b). Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer.

[64] Touati, A. and Ollivier, Y. (2021). Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23.

[65] Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, 30.

[66] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[67] Wang, T. and Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

[68] Wang, Z., Novikov, A., Żołna, K., Springenberg, J. T., Reed, S., Shahriari, B., Siegel, N., Merel, J., Gulcehre, C., Heess, N., et al. (2020). Critic regularized regression. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 7768–7778.

[69] Yang, R., Lu, Y., Li, W., Sun, H., Fang, M., Du, Y., Li, X., Han, L., and Zhang, C. (2022). Rethinking goal-conditioned supervised learning and its connection to offline rl. *arXiv preprint arXiv:2202.04478*.

[70] Yarats, D., Kostrikov, I., and Fergus, R. (2020). Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*.

[71] Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., and Fergus, R. (2019). Improving sample efficiency in model-free reinforcement learning from images. *arXiv preprint arXiv:1910.01741*.

[72] Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. (2020). Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836.

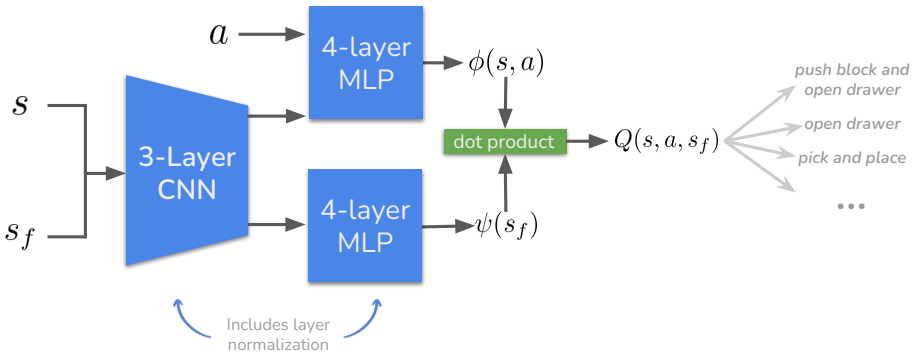


Figure 3: **Overview of network architecture.** Our experiments suggest that this architecture can stabilize and accelerate learning for contrastive RL. Layer normalization is applied to each layer of the visual feature extractor (CNN) and contrastive representations (MLP). We initialize weights in final layers of MLP close to zero to encourage alignment of contrastive representations, and use data augmentation to prevent overfitting.

A Review of TD Contrastive RL

The recursive definition of the discounted state occupancy measure (Eq. 1) allows us to rewrite Eq. 2 into a *temporal different* (TD) variant [15], with $w(s,a,s_f) \triangleq \frac{\sigma(f(s,a,s_f))}{1-\sigma(f(s,a,s_f))}$,

$$\max_f \mathbb{E}_{\substack{(s,a)\sim p(s,a), s'\sim p(s'|s,a) \\ s_f\sim p(s_{t+}), a'\sim\pi(a'|s,g)}} \Big[(1-\gamma)\log\sigma(f(s,a,s')) + \log(1-\sigma(f(s,a,s_f))) + \gamma \lfloor w(s',a',s_f)\rfloor_{\text{sg}} \log\sigma(f(s,a,s_f))\Big]. \tag{3}$$

A prototypical example of this TD variant is C-Learning [15] that lies in the family of contrastive RL methods [17]. We will focus on the TD learning critic objective in our discussion and adapt it to the offline setting with a goal-conditioned behavioral cloning regularized policy objective:

$$\max_{\pi(\cdot|\cdot,\cdot)} \mathbb{E}_{p_g(g)p(s,a_{\text{orig}})\pi(a|s,g)} \left[(1-\lambda)\cdot f(s,a,g) + \lambda \log\pi(a_{\text{orig}} \mid s,a)\right]. \tag{4}$$

B Intuition: A Connection between Contrastive RL and Hard Negative Mining

One practical challenges with learning value functions or distance functions for goal-reaching tasks is that visually similar images may actually require a large number of steps to transit between. Prior work has addressed this challenging by manually mining hard negative examples [62, 54]. Our motivation for studying contrastive methods was a hypothesis that, with appropriate design decisions, contrastive methods would automatically induce a form of hard-negative mining. In particular, we wrote the gradient of the objective (Eq. 2). with respect to the representation of random goals $\psi(s_f^-)$,

$$\frac{\partial}{\partial\psi(s_f^-)}\mathcal{L}_2(\phi(s,a),\psi(s_f^-)) = -\sigma(\phi(s,a)^\top\psi(s_f^-))\phi(s,a).$$

This gradient "pushes" the representation of a random goal $\psi(s_f^-)$ away from the anchor $\phi(s,a)$ with a larger weight when it is misclassified. Intuitively, if we think of these weights as importance weights, then this gradient corresponds to finding the hardest negative examples within the batch. We conjecture that realizing this effect will require a large batch size, as well as a few other design decisions, which we discuss next.

C Architecture Diagram

An overview of our network architecture is shown in Fig. 3.

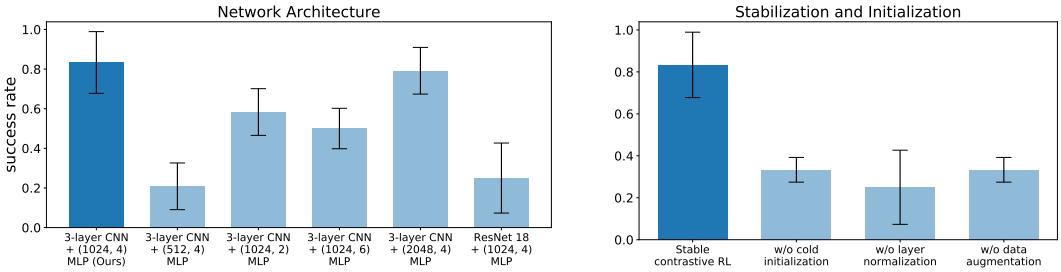


Figure 4: **Importance of architecture initialization, and normalization.** *(Left)* Our experiments suggest that using a deep CNN *decreases* performance, but using a wide MLP is important. *(Right)* The good performance of our implementation depends on "cold initialization", layer normalization, and data augmentation. Prior work [17] uses an even smaller network architecture (256, 2) and none of the stabilization/initialization decisions.

D Experimental Details

Our experiments start with studying design decisions that drive stable contrastive RL and use simulated and real-world benchmarks to compare contrastive RL to other offline goal-conditioned policy learning methods, including those that use conditional imitation and employ representation pre-trained by auxiliary objectives. We then analyze unique properties of the representations learned by stable contrastive RL, providing an empirical verification of the connection drawn in Appendix B. Finally, we conduct various ablation studies to test the generalizing and scalability of our algorithm.

D.1 Experimental Setup

Tasks. Our experiments use a suite of simulated and real-world goal-conditioned control tasks based on prior work [18, 19, 12, 42]. *First*, we use a benchmark of five manipulation tasks proposed in [18, 19] (Fig. 5). The observations and goals are $48 \times 48 \times 3$ RGB images of the current scene. These tasks are challenging benchmarks because solving most of them requires multi-stage reasoning. *Second*, we use the goal-conditioned locomotion benchmark proposed in [42]. The observations and goal poses are $64 \times 64 \times 3$ RGB images. This benchmark helps study those design decisions for offline goal-reaching beyond manipulation. *Third*, we study the performance of our algorithm in real-world using a 6-DOF WidowX250 robot arm. We use Bridge dataset [12] where observation and the goal are $128 \times 128 \times 3$ RGB images with much noisier backgrounds than the simulated tasks. For evaluation, we use three tasks from `Toy Kitchen 2` scene in the dataset where the initial and desired object positions are unseen during training (see Fig. 1). These tasks are challenging because they require multi-stage reasoning (e.g., the drawer can only be opened after the orange object has been moved).

Offline dataset. The offline manipulation dataset we used in simulation consists of near-optimal demonstrations of primitive behaviors, such as opening the drawer, pushing blocks, and picking up objects. The scripted data collection policy can access the underlying object states. The lengths of demonstration vary from 50 to 100 time steps and the offline dataset approximately contains 250K transitions in total. We note that none of the offline trajectories complete the demonstration from the initial state to the goal. For evaluation, we create a dataset of 50 goals by randomly sampling the positions of objects and the robot arm, and evalaute the success rate of reaching these goals. The locomotion benchmark includes a mixture of optimal and suboptimal transitions. We randomly collect 100K transitions from the replay buffer of a policy trained with LEXA [42], which achieves success rates of 75% on `Walker` and 58% on `Quadruped` at convergence. Our real-world offline dataset uses the existing Bridge Data [12], including around 20K demonstrations of various manipulation skills. Note that some trajectories in this dataset are suboptimal since they were collected by a weakly scripted data collection policy that sometimes misses grasping the target objects. This dataset provides broad and diverse demonstrations to evaluate whether an agent is able to stitch skills and generalize to different scenes.

D.2 Ablation Study

The good performance of contrastive RL depends on a few key design decisions (e.g., architectures, initialization), which we will study in this section. We will use the `drawer` task and report mean and standard deviations over three random seeds. The appendix contains additional experiments

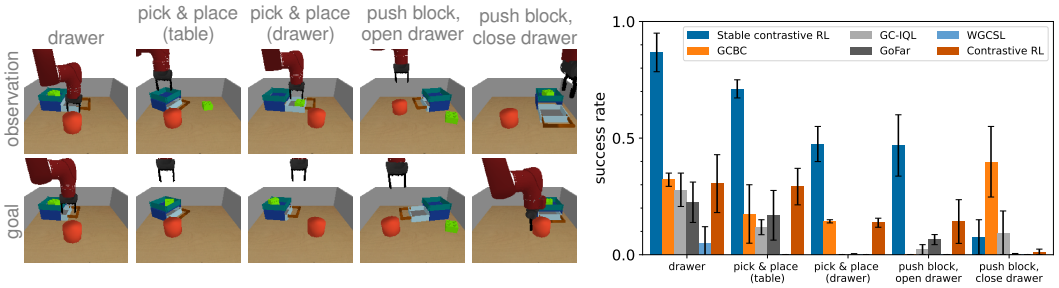


Figure 5: **Evaluation on simulated manipulation tasks.** *(Left)* The five simulated evaluation tasks, with examples of the initial observation (top) and goal observation (bottom). *(Right)* Stable contrastive RL outperforms all baselines on $\frac{4}{5}$ tasks, often by a large margin. The comparison with "contrastive RL" underscores the importance of our design decisions.

underscoring the importance of large batches (Appendix E.13) and showing that lower-dimensional representations achieve higher success rates (Appendix E.15).

Network architecture. We study the architectures for the image encoder and the subsequent MLP. For the image encoder, we compare a shallow *3-layer CNN* (similar to DQN [43]) to a *ResNet 18* [30]. Fig 4 *(Left)* shows that 3-layer CNN encoder outperforms ResNet 18 encoder by $2.89\times$ (81% vs. 28%), when incorporting same design decisions other than architectures. Additional experiments in Appendix E.1 suggest that this counterintuitive finding can likely be explained by overfitting. We next study the architecture of the subsequent MLP, denoted as $(\text{width}, \text{depth})$. Our experiments show that a $(1024, 4)$ MLP yields the best result, suggesting that wider MLPs perform better. In subsequent experiments, we use the 3-layer CNN as the visual feature extractor followed by a $(1024, 4)$ MLP. Appendix E.2 contains additional experiments on a "pretrain and finetune" setting using a pretrained visual encoder.

Stabilization and initialization. We hypothesize that layer normalization balances feature magnitude and prevents gradient interference, while data augmentation mitigates overfitting. Our experiments in Fig. 4 *(Right)* suggest that both layer normalization and data augmentation (via random cropping) are critical for good performance.

We next study weight initialization, using the "cold initialization" strategy outlined in Sec. 2.2. We find that this strategy, which uses very small initial weights $\textsc{Unif}[-10^{-12}, 10^{-12}]$, boosts performance by 2.49 times (82% vs. 33%) than the initialization strategy $\textsc{Unif}[-10^{-4}, 10^{-4}]$. Additional experiments in Appendix Fig. 27 show that $\textsc{Unif}[-10^{-8}, 10^{-8}]$ and $\textsc{Unif}[-10^{-16}, 10^{-16}]$ achieve only slightly worse results than $\textsc{Unif}[-10^{-12}, 10^{-12}]$. In Appendix E.14, we show a smaller learning rates and a learning rate "warmup" have a different (worse) effect than cold initialization. Results from these ablation experiments encourage us to apply layer normalization, data augmentation, and cold initialization in other experiments.

Summary. Our experiments suggest the following design decisions stabilizing contrastive RL: *(i)* Using a simple 3-layer CNN visual feature extractor followed by a wide MLP. *(ii)* Add layer normalization. *(iii)* Initialize last-layer weights of the MLP with small values, $\textsc{Unif}[-10^{-12}, 10^{-12}]$. *(iv)* Apply random cropping to both the state and goal images. *(v)* Use a large batch size (2048). We use **stable contrastive RL** to denote this combination of design decisions and provide a implementation in Appendix D. We call our method "stable" because we find that these design decisions yield more stable learning curves (see Appendix Fig. 13). We also ran an additional ablation experiments on `push block, open drawer` (see Appendix Fig. 14), drawing similar conclusions.

D.3 Comparing to Prior Methods

Next, we study the efficacy of stable contrastive RL comparing to prior methods, including those that use goal-conditioned supervised learning and those that employ auxiliary representation learning objectives. We run experiments on datasets collected from both simulation and the real-world without any interaction with the environment during training – that is, all experiments focus on offline, goal-conditioned RL.

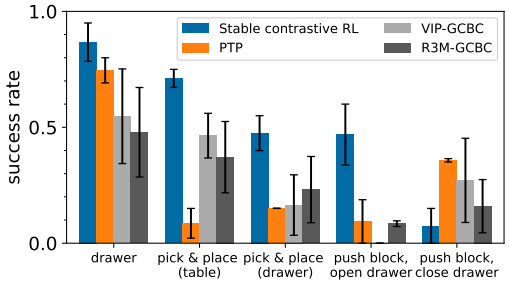


Figure 6: **Comparing to pretrained representations.** On $\frac{4}{5}$ tasks, stable contrastive RL outperforms baselines that use frozen representations from a VAE (PTP [18]), VIP [40], and R3M [46].

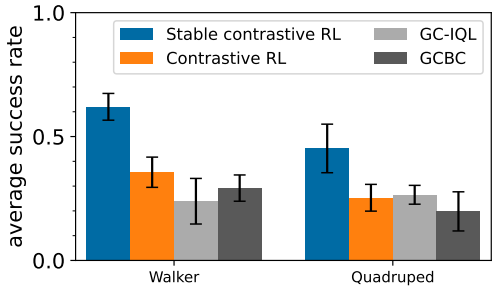


Figure 7: **Evaluation on simulated locomotion tasks.** Stable contrastive RL outperforms contrastive RL, GC-IQL and GCBC in two locomotion tasks. Each success rate is calculated by averaging over 12 diverse goal poses [42].

Simulation evaluation – manipulation. We first compare to five baselines that build on goal-conditioned behavioral cloning. The simplest goal-conditioned behavioral cloning ("GCBC") [7, 10, 13, 21, 38, 49, 60] trains a policy to conditionally imitate trajectories reaching goal g. Goal-conditioned IQL (GC-IQL) is a goal-conditioned version of IQL [34], a state-of-the-art offline RL algorithm. GoFar [39] is an improvement of GCBC that weights the log likelihood of actions using a learned critic function. The fourth baseline is WGCSL [69], which augments GCBC with discounted advantage weights for policy regression. We also include a comparison with contrastive RL [17], which helps us test whether those design decisions improve the performance. Like stable contrastive RL, these baselines are all trained directly on goal images in an end-to-end fashion, not using any explicit feature learning. As shown in Fig. 5, stable contrastive RL matches or surpasses all baselines on four of the five tasks. On those more challenging tasks (`push block, open drawer`; `pick & place (table)`; `pick & place (drawer)`), we see a marked difference between these baselines and stable contrastive RL. However, the baseline methods fail to solve these more challenging tasks. Stable contrastive RL performs worse than GC-IQL and GCBC on one of the tasks (`push block, close drawer`), perhaps because the block in that task occludes the drawer handle and introduces partial observability.

We next compare to three methods that employ learning policy on top of pre-trained representations. PTP [18] focuses on finding sub-goals in a pre-trained VQ-VAE [65] representation space and learns a policy to reach them sequentially. We also compare to a variant of GCBC learning on top of features from VIP [40]. We call this method VIP-GCBC. Similarly, our third baseline, R3M-GCBC, trains a policy via supervised learning on top of representations learned by R3M [46]. Results in Fig. 6 show that stable contrastive RL outperforms all baselines on $\frac{4}{5}$ tasks. While the baselines all require separate objectives for RL and representation learning, stable contrastive RL achieves excellent results without the need for an additional representation learning objective. The full curves can be found in Appendix E.8. Appendices E.11 and E.12 show comparisons against an end-to-end version of PTP and to a variant without sub-goal planning.

Simulation evaluation – locomotion. Our next set of experiments focus on locomotion tasks. We conduct these experiments to see whether the same design decisions are helpful on tasks beyond manipulation, and on tasks from which we have only suboptimal data. We compare stable contrastive RL against contrastive RL, GCBC, and GC-IQL on two goal-reaching tasks from prior work [42]: `Walker` and `Quadruped`. We evaluate all methods in the offline setting, recording the average success rates over reaching 12 unique goal poses. As shown in Fig. 7, stable contrastive RL outperforms all three baselines on both locomotion tasks, achieving a 33% higher (relative) average success rate than the strongest baseline.

Real-world evaluation. We next study whether these methods can effectively solve real world, goal-directed, image-based manipulation tasks, learning entirely from offline data. We compare stable constrastive RL against the strongest performing baseline from the simulated results, GCBC. We compare to GC-IQL as well because it performed well in the simulated experiments, and its simplicity makes it practically appealing. We evaluated on three goal-reaching tasks and report success rates in

Fig. 1. Stable contrastive RL performs similarly to baselines on the simple (`reach eggplant`) task, while achieving a 60% success rate on the two harder tasks (`pick & place spoon`, `push can`), where all baselines fail to make progress. We conjecture that this good performance might be explained by the hard-negative mining dynamics of stable contrastive RL, which are unlocked by our design decisions.

D.4 Latent Interpolation

Self-supervised learning has achieved remarkable success in learning representations for image and representation data [48, 63, 8]. While auxiliary losses or data augmentation improve reconstruction and robustness of the learned representation, we hypothesize that representation acquired by self-supervised RL approach contains task-specific information. Given an initial image (Fig. 2 *far left*) and the desired goal image (Fig. 2 *far right*), we interpolate between the representations of these two images, and retrieve the nearest neighbor in a held-out validation set. For comparison, we also include direct interpolation in pixel space. As expected, interpolating in pixel space generates unrealistic images. While linearly interpolating in the VAE and GCBC representation spaces produce realistic images, the retrieved images fails to capture causal relationships, motivating the necessity of adding other machinery, e.g., relabeling latent goal [45, 52, 54] and latent sub-goal planning with value constraints [18, 19]. When interpolating the representations of stable contrastive RL, the intermediate representations correspond to sequence of observations that the policy should visit when transiting between the initial observation and the final goal. Appendices E.5 and E.6 contain a quantitative experiment and additional visualizations.

D.5 The Arm Matching Problem

Previous works have found that some goals are easier to be distinguish from the others (easy negative examples) [54, 62, 1], while some goals might require temporal-extended reasoning to reach (hard negative examples). For example, on task `push block, open drawer`, prior goal-conditioned algorithms [18, 34] simply match the arm position with the one in the goal image and fail to move the green block to its target position. We call this failure *the arm matching problem*. Our theoretical analysis in Appendix B suggests that contrastive approaches should perform this sort of hard negative mining and avoid the arm matching problem. We test this hypothesis in Fig. 8 by comparing the Q values learned by stable contrastive RL and contrastive RL, aiming to assess whether our design decisions lead to more accurate estimates of the Q-values. Prior work has suggested that pre-trained VAE representations can also be effective, so we also compare to a version of GC-IQL applied on top of VAE features [18, 19]. We normalize these Q values by the minimum and maximum values in a rollout. Fig. 8 compares the Q values for an observation where the object is not in the correct location, but the gripper is in the correct position. GC-IQL (mistakenly) assigns high Q values to this observation. In contrast, stable contrastive RL predicts a small value for this observation, highlighting the accuracy of its Q function and showing how it can avoid arm matching behavior. Appendix E.7 includes a comparison of Q values in an optimal trajectory.

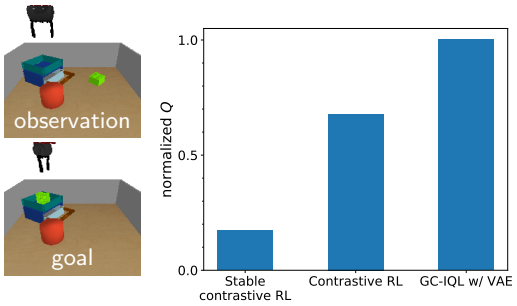


Figure 8: **Misclassifying success.** GC-IQL mistakenly predicts that this observation has succeeded in reaching this goal, even though the object is in the incorrect position. In contrast, stable contrastive RL recognizes that this observation is far from the goal and assigns it a low Q value.

D.6 The Influence of Dataset Size on Performance

Scaling model performances with the amount of data have been successfully demonstrated in CV and NLP [6, 28], motivating us to study whether contrastive RL offers similar scaling capabilities in the offline RL setting. To answer this question, we run experiments with dataset sizes increasing from 100K to 1M, comparing stable contrastive RL against a baseline that uses pre-trained features to improve policy learning. The fact that this baselines doesn't improve suggests that most of the gains observed for stable contrastive RL may be coming from better representations, rather than from a better reactive policy. Results in Fig. 9 shows that stable contrastive RL effectively makes use of

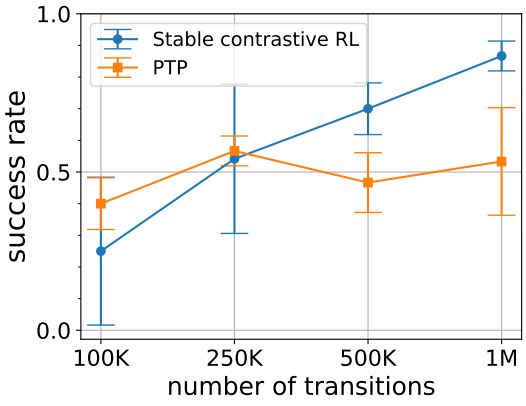


Figure 9: **The influence of dataset size.** Increasing the amount of offline data boosts the success rate of stable contrastive RL, while the performance of PTP saturates after 250,000 transitions.

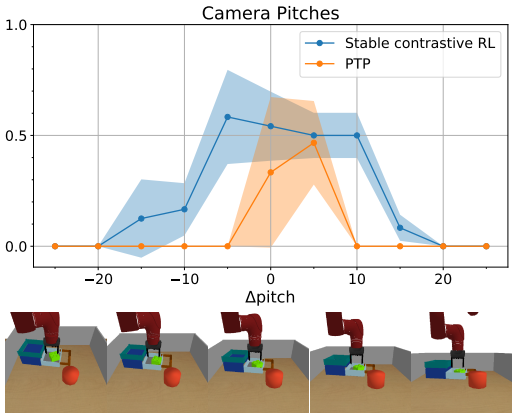


Figure 10: **Generalizing to unseen camera angles.** Stable contrastive RL retrains good performance on a broad range of unseen camera angles (pitch). Appendix Fig. 23 shows similar generalization capabilities for variations in yaw and object color.

additional data, with a success rate that increases by $3\times$ as we increase the dataset size. In contrast, the performance of PTP saturates around 250K transitions, suggesting that most of the gains observed for stable contrastive RL may be coming from better representations, rather than from a better reactive policy. In Appendix E.9, we also find that the binary accuracy of the stable contrastive RL (measured on predicting the future) increases with dataset size, suggesting that the method benefits from a more diverse pool of positive and negative examples.

D.7 Generalizing to Unseen Camera Angles and Object Colors

Our next set of experiments study the robustness of stable contrastive RL. We hypothesize that stable contrastive RL might generalize to unseen tasks reasonably well for two reasons: *(1)* stable contrastive RL resembles the contrastive auxiliary objectives used by prior work to improve robustness [46, 36, 37]; *(2)* because stable contrastive RL learns features solely with the critic objective, we expect that the features will not retain task-irrelevant information (unlike, say, representation based on auto-encoding). We run an experiment comparing the generalization of stable contrastive RL against a baseline that uses features pre-trained via reconstruction by varying the environment. We show results on varying pitch in Fig. 10, and include the other results and full details in Appendix E.10. These experiments provide preliminary evidence that contrastive representations might generalize reasonably well.

D.8 Implementations and Hyperparameters

Implementations. We implement stable contrastive RL using PyTorch [50][1]. We use the open-sourced implementation of PTP[2] as our baseline and adapt the underlaying IQL algorithm to GC-IQL and GCBC. Unless otherwise mentioned, we use same hyperparameters as this implementation. For baselines WGCSL[3], GoFar[4], VIP-GCBC[5], and R3M-GCBC[6], we adapted the implementation provided by the authors.

Hyperparameters and computation resources. We implement stable contrastive RL using PyTorch again on top of the public PTP codebase. Our algorithm is adapted from contrastive RL implementation[7] and incorporate those novel design decisions mentioned in Sec. D.2. We summarize

[1] https://anonymous.4open.science/r/stable_contrastive_rl-5A42.
[2] https://github.com/patrickhaoy/ptp
[3] https://github.com/YangRui2015/AWGCSL
[4] https://github.com/JasonMa2016/GoFAR
[5] https://github.com/facebookresearch/vip
[6] https://github.com/facebookresearch/r3m
[7] https://github.com/google-research/google-research/tree/master/contrastive_rl

Table 1: Hyperparameters for stable contrastive RL.

Hyperparameters	Values	Notes
batch size	2048	
number of training epochs	300	
number of training iterations per epoch	1000	equivalent to number of gradient steps per epoch
dataset size	250K	number of transitions (s, a, s', s_f)
image size	$48 \times 48 \times 3$; $128 \times 128 \times 3$	Size of RGB images in the simulated and real-world tasks
episode length	400	
image encoder architecture	3-layer CNN	kernel size = (8, 4, 3), number of channels = (32, 64, 64), strides = (4, 2, 1), paddings = (2, 1, 1)
policy network architecture	(1024, 4) MLP	
critic network architecture	(1024, 4) MLP	
weight initialization for final layers of critic and policy	$\textsc{Unif}[-10^{-12}, 10^{-12}]$	
policy stand deviation	0.15	
contrastive representation dimension	16	
data augmentation	random cropping	replicating edge pixel-padding size = 4
augmentation probability	0.5	applying random cropping to both s and g in the GCBC regularizer in Eq. 4 with this probability
discount factor	0.99	
learning rate of Adam [33] optimizer	0.0003	

hyperparameters in Table 1. For each experiment, we allocated 1 NVIDIA V100 GPU and 64 GB of memories to do computation.

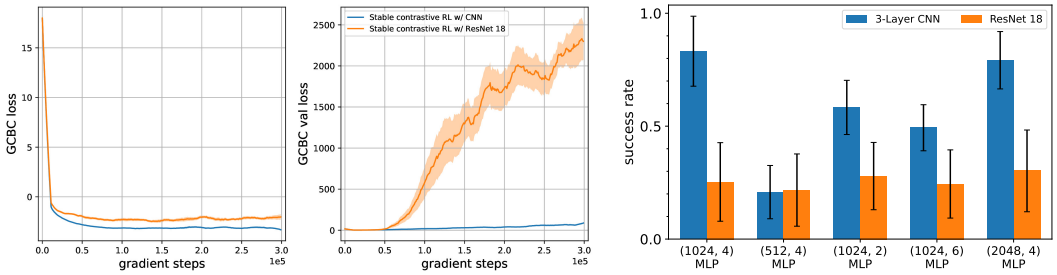


Figure 11: ResNet 18 is susceptible to overfitting for the policy on offline datasets.

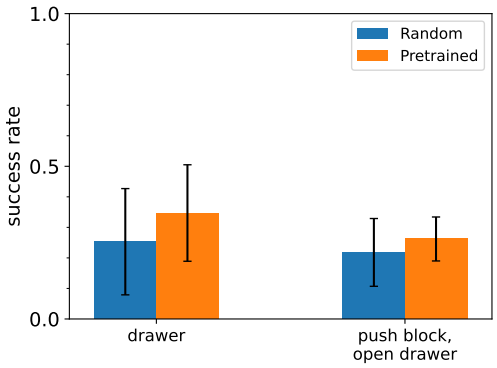


Figure 12: Using weights pretrained on supervised learning tasks might boosts performances of stable contrastive RL.

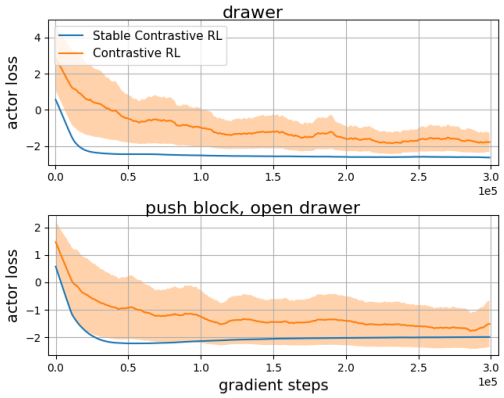


Figure 13: "Stable" contrastive RL is more stable than the original contrastive RL.

E Additional Experiments

E.1 Overfitting of the ResNet 18 Encoder

We hypothesize that the over-parameterized ResNet 18 visual encoder overfits to the dataset when comparing with the 3-layer CNN. To study this, we conduct experiments on task `push block, open drawer`, showing training and validation loss of the GCBC regularization for both visual backbones. Note that we apply the cold initialization, layer normalization, and random cropping data augmentation introduced in Appendix D.2 to both ResNet 18 and 3-layer CNN to make a fair comparison in Fig. 4 and these additional experiments. The observation that the training loss remains low while the validation loss starts to increase after 50K gradient steps (Fig. 11 *(Left)*) suggests that the larger ResNet 18 is susceptible to overfitting for the policy. Additionally, we include ablations between 3-layer CNN and ResNet 18 with different subsequent MLP architectures for completeness in Fig. 11 *(Right)*. We find that ResNet 18 with deeper and wider MLP architecture does not improve the performance significantly and the best combination still underperforms Stable contrastive RL by $2.75\times$. Together with the GCBC training and validation losses above, we suspect that the ResNet 18 overfits to the training data *despite* the data augmentation (random cropping) we applied in attempts to mitigate this.

E.2 Effect of Pretraining

We ran additional experiments to study the effect of pretrained models for contrastive RL. We applied our design decisions to stable contrastive RL variants that use a ResNet-18 visual encoder and compared two initialization strategies: initialization with random weights and initialization with weights pretrained on ImageNet [30]. In Fig. 12, we report results on `drawer` and `push block, open drawer`, a challenging multi-stage task that involves first pushing the block and then opening the drawer.

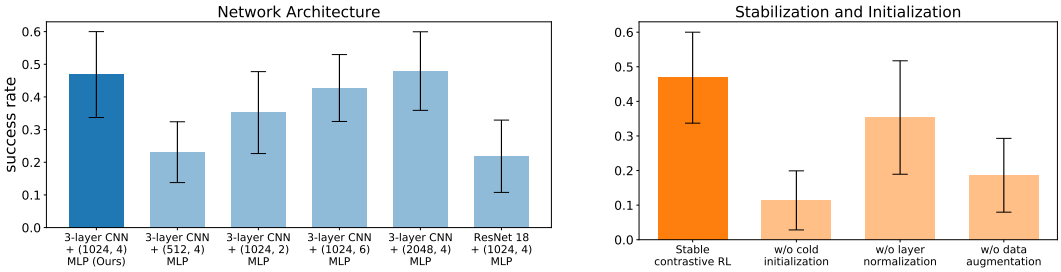


Figure 14: Additional ablations on `push block, open drawer`.

We find that using pretrained weights boosts performance of stable contrastive RL on the relatively simple `drawer` but does not significantly benefit the algorithm on `push block, open drawer`. These results suggest that those weights pretrained on supervised learning tasks might help the learning of contrastive RL in some cases. However, we note that we have only explored one pretraining method. Different pretraining datasets and methods (e.g., R3M [46], VIP [40]) may produce different results. Nonetheless, we leave the investigation of which pretraining methods best accelerate stable contrastive RL to future work.

E.3 Understanding "Stable" in Stable Contrastive RL

The reason for using "stable" contrastive RL to name our method is that the original contrastive RL [17] is unstable. We provide a comparison between the learning curves of stable contrastive RL and contrastive RL on tasks `drawer` and `push block, open drawer`. The observation that the actor loss of stable contrastive RL are less oscillatory than those of the original contrastive RL demonstrates that our design decisions improve stability.

E.4 Additional Ablations

We ran a set of additional ablation experiments studying whether the techniques used by stable contrastive RL still boost the performance. To do this, we compare stable contrastive RL with the same set of design decision variants as in Fig. 4 on tasks `push block, open drawer`. We report the results in Fig. 14.

Regarding network architectures, we find that an MLP with 2, 4, and 6 numbers of layers and 1024 units perform similarly on this task, suggesting that our choice, an (1024, 4) MLP, is still effective. For stabilization and initialization techniques, we find that cold initialization and data augmentation still improve the performance significantly, while removing layer normalization decreases the performance by 24.5%. Taken together, our design decisions continue to outperform or perform similarly to other variants on this new task.

E.5 More Examples of Latent Interpolation

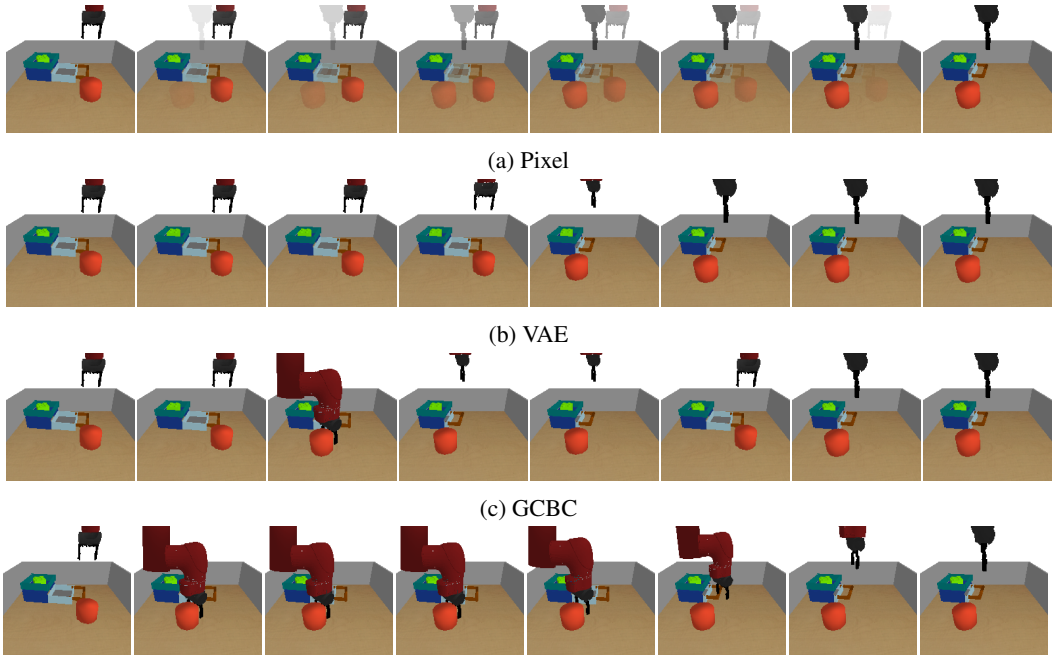
We show more examples of interpolating different representations on tasks `drawer`, `pick & place (table)`, and `pick & place (drawer)` in Fig. 15, Fig. 16, and Fig. 17 respectively.

E.6 Measuring Interpolation

To quantitatively study interpolation, we run another set of experiments comparing the similarity of interpolated representations (images) with the ground truth sequence. Specifically, we retrieve the nearest neighbor of each interpolated representation from a policy rollout and label the time step of interpolations accordingly (perm$[t]$). With these time steps, we define a permutation error with respect to the ground truth time steps:

$$\sum_{t=0}^{T} |\text{perm}[t] - t|$$

Results in Fig. 18 demonstrate that interpolation in the VAE representation space and the pixel space are not well-aligned with the ground truth time steps (better than random permutations), while contrastive representations achieves a lower error, suggesting that it might contain information that is uniquely well-suited for control and potentially leverage a goal-conditioned policy.



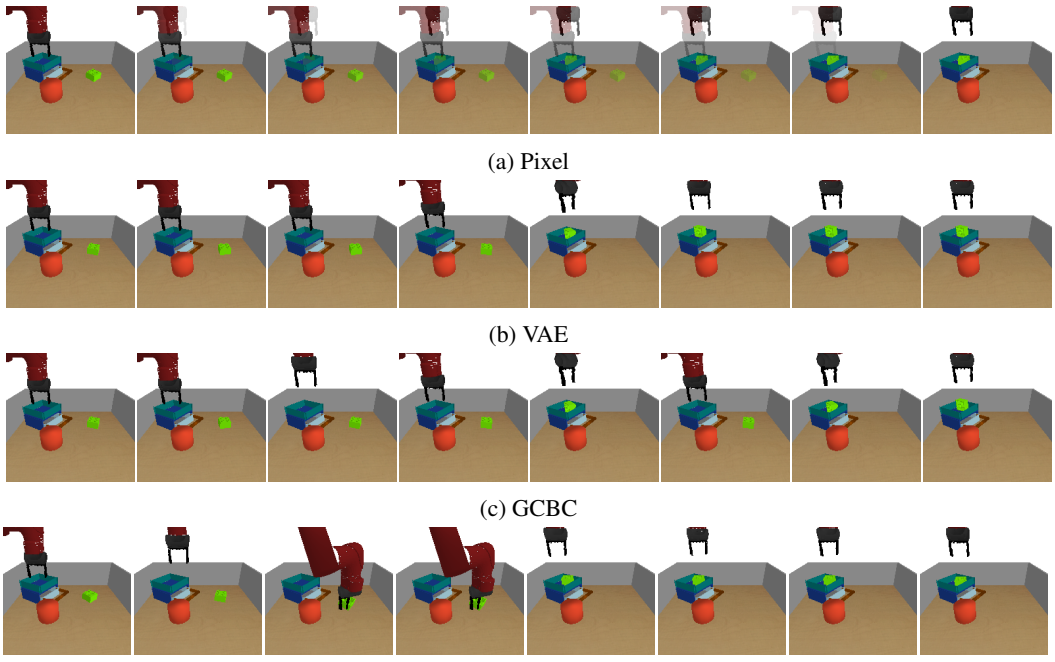
(a) Pixel

(b) VAE

(c) GCBC

(d) Stable contrastive RL

Figure 15: Interpolation of different representations on `drawer`.



(a) Pixel

(b) VAE

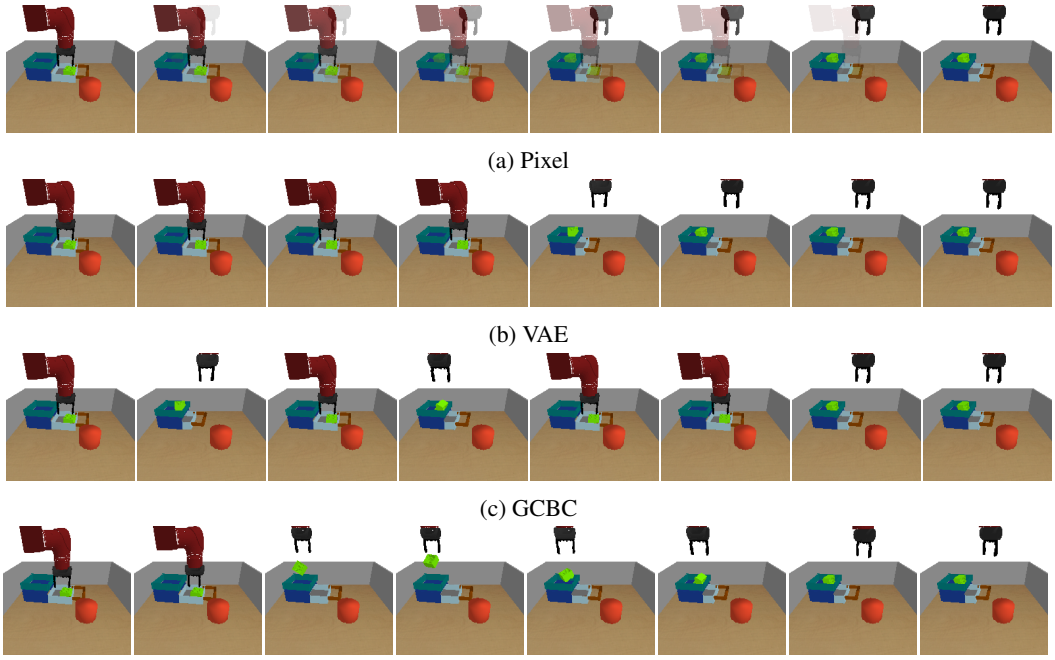
(c) GCBC

(d) Stable contrastive RL

Figure 16: Interpolation of different representations on `pick & place (table)`.

E.7 Arm Matching

To evaluate the performance of contrastive RL in tackling the arm matching problem, we collect a trajectory of contrastive RL trained on the offline dataset and comparing the Q value predicted by stable contrastive RL with a GC-IQL trained on top of VAE representations, assuming that pre-trained features might help mitigate the arm matching problem as well. We normalize both Qs by values of



(a) Pixel

(b) VAE

(c) GCBC

(d) Stable contrastive RL

Figure 17: Interpolation of different representations on `pick & place (drawer)`.

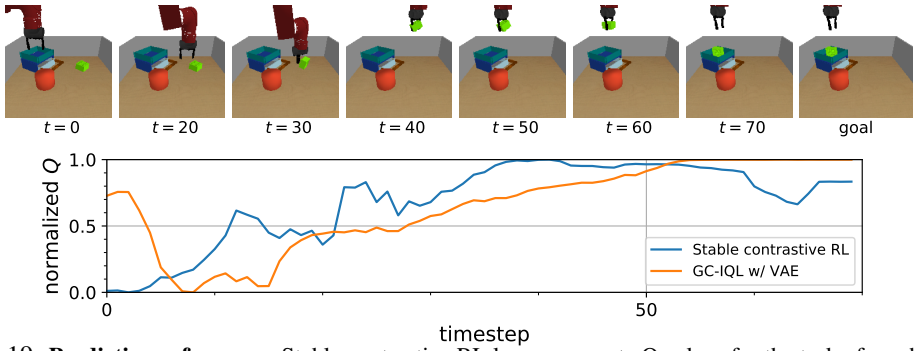


Figure 19: **Predictions of success.** Stable contrastive RL learns accurate Q values for the task of reaching the goal shown above: they increase throughout a successful rollout. On the contrary, the GC-IQL baseline mistakenly predicts that picking up the green object *decreases* the Q-value, likely because the GC-IQL representations ignore the position of the green block. This experiment suggests that representations derived from a VAE, like those learned by GC-IQL and many prior methods [71, 45, 47], may be less effective at predicting success than those learned by contrastive RL methods.

the minimum and the goal. As shown in Fig. 19, stable contrastive RL correctly learns a monotonic increasing Q throughout the rollout, while GC-IQL learns a V-shape Q relating higher values to a closer arm to the target position, ignoring the position of the green block ($t = 30$). This experiment demonstrates that representations derived from VAE may be less efficient at predicting success than contrastive representations in the scene involving temporal-extended reasoning.

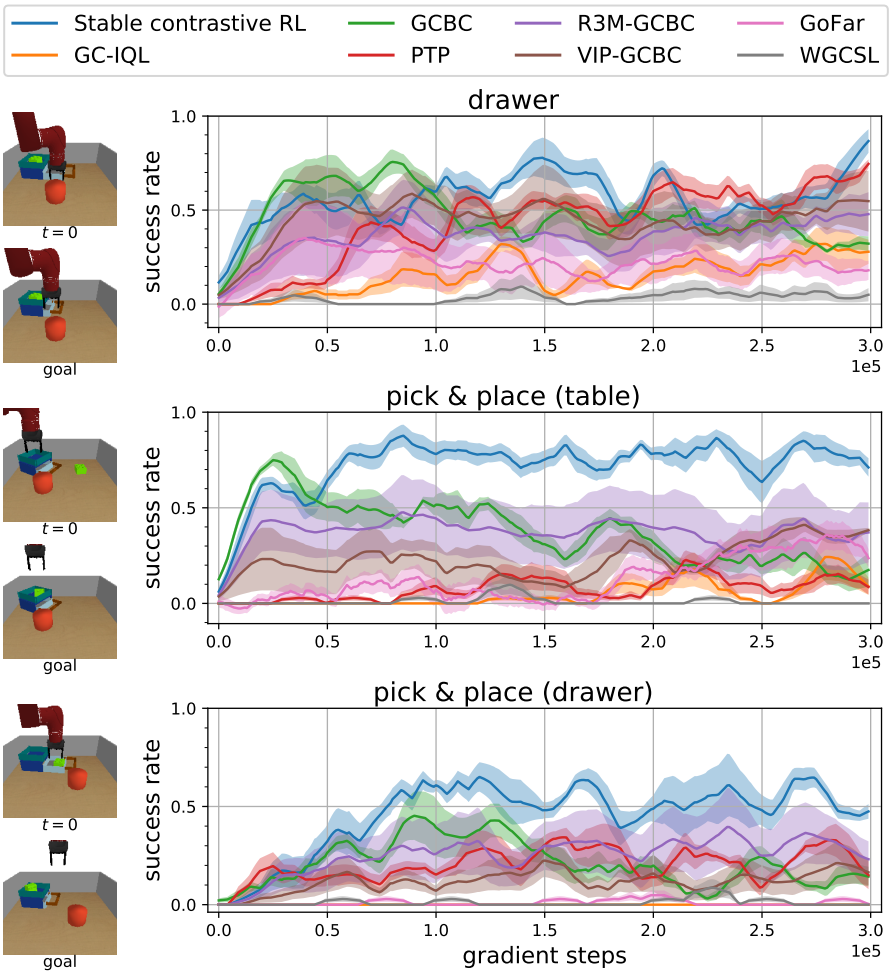


Figure 20: **Evaluation on simulated manipulation tasks**. Stable contrastive RL outperforms all baselines on `drawer`, `pick & place (table)`, and `pick & place (drawer)`.

E.8 Evaluation on simulated manipuation tasks

We report results in Fig. 20 and Fig. 21 with curves indicating mean success rate and shaded regions indicating standard deviation across 5 random seeds after 300K gradient steps' training. Stable contrastive RL outperforms or achieves similar performance on 4 out of 5 tasks comparing to other baselines. These results suggest that when accompanied with proper techniques, stable contrastive RL is able to leverage a diverse offline dataset and emerges a general-purpose goal-conditioned policy, thus serving as a competitive offline goal-conditioned RL algorithm.

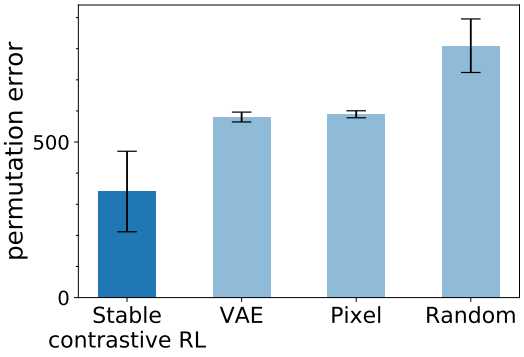


Figure 18: To quantitatively evaluate interpolation, we measure the similarity of the retrieved representations with the ground truth sequence. Stable contrastive RL achieves lower error than the alternative methods. See text for details.

E.9 Dataset Size

Scaling model performances with the amount of data have been successfully demonstrated in CV and NLP [6, 28], motivating us to study whether contrastive RL offers similar scaling capabilities in the offline RL setting. Sec. 2.2 have discussed the benefits of training with more positive and negative pairs to contrastive representation learning, suggesting that increasing amounts of data might also

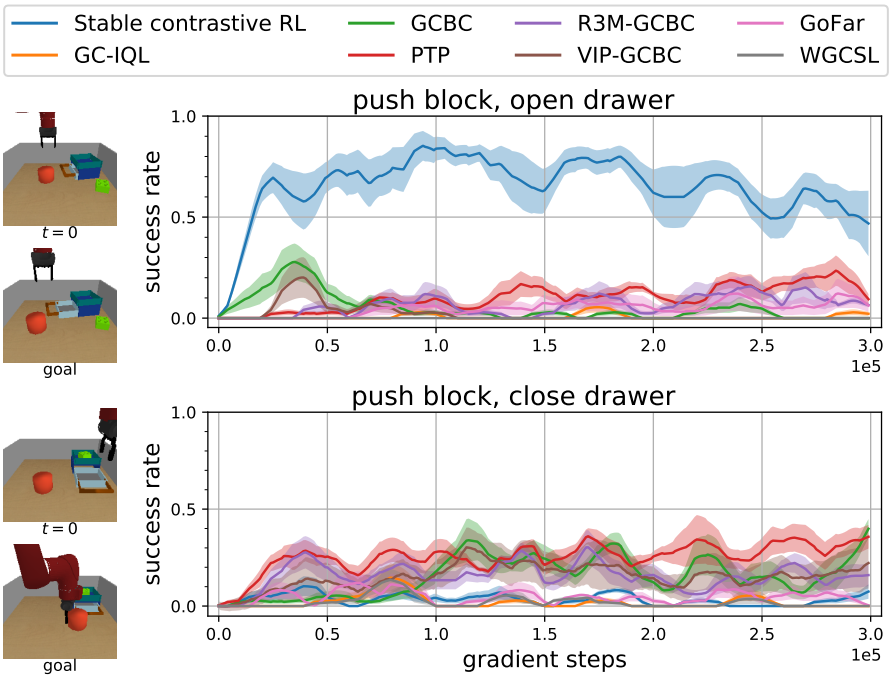


Figure 21: **Evaluation on simulated manipulation tasks**. Stable contrastive RL outperforms or performs similarly to all baselines on `push block, open drawer` and `push block, close drawer`.

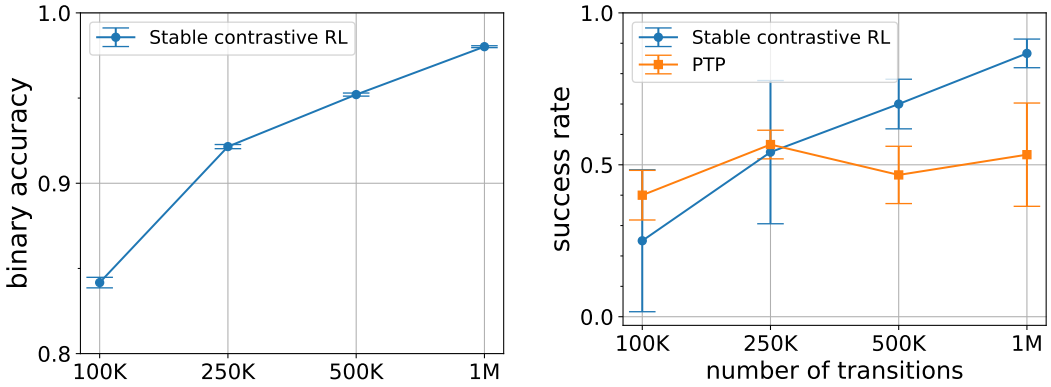


Figure 22: Increasing the amount of offline data improves the binary future prediction accuracy of stable contrastive RL and boosts the success rate.

boost the performance of contrastive RL. We empirically study the hypothesis that *stable contrastive RL learns a representation that achieves better performances with growing dataset size.*

To test this hypothesis, we create datasets of different sizes containing various manipulation skills, select task `drawer` for evaluation, and compare the accuracy of stable contrastive RL in predicting future states on each of the datasets. We compare stable contrastive RL against PTP that is based on a pre-trained VAE. We compare both algorithms by training three random seeds of them for 300K gradient steps on each dataset. Fig. 22 shows the mean and standard deviation of success rates over 10 rollouts and binary future prediction accuracies computed between critic outputs and ground truth labels. Observe that when the size of dataset *increased*, the binary accuracy of contrastive RL *increase* accordingly ($84\% \rightarrow 98\%$), suggesting that the algorithm indeed strengthens its representation in future prediction. These results make sense, as increasing the size of the dataset is especially important for contrastive approaches that directly use all of the data as *negative* examples. In addition, our experiments show that contrastive RL consistently improves its success rates when we use larger dataset: $31\% \rightarrow 84\%$, verifying our hypothesis. We suspect that the emerging of a denser distribution of positive and negative pairs given a larger dataset might facilitate policy learning. In contrast, a

steady or even lower success rate of PTP given the increasing amounts of data might due to the capacity limitation of pre-trained VAE representation.

E.10 Generalization

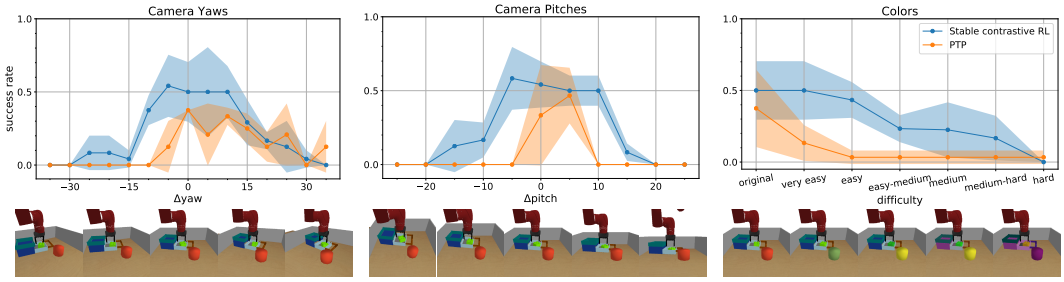


Figure 23: **Generalizing to unseen tasks.** Stable contrastive RL retrains good performance on a broad range of unseen camera angles (yaws / pitches) and object colors.

One of the drawback of driving RL by pre-trained visual representation learning is tying algorithm performance with the latent representation to reconstruct corresponding images. Figure 2 has shown the *support* of VAE representation distribution could be discontinuous and there could be noise in a out-of-distribution reconstructed image, indicating the degradation of RL performance on unseen tasks. Rather than predicating on a pre-trained visual representation, stable contrastive RL trains its self-supervised representation with the actor-critic framework [17], expecting to acquire a more robust and generalizable algorithm. We hypothesize that *stable contrastive RL generalizes better to unseen tasks than prior method*.

To test this hypothesis, we ran experiments with one of our goal-conditioned tasks in three visually different scenarios (Fig. 23): varying the camera yaw angle, varying the camera pitch angle, and varying the color of objects in the scene. We evaluate the policy of stable contrastive RL and PTP after 300K gradient steps' offline training in these three scenerios, plotting the mean and standard deviation of success rate across three random seeds. Since the color of objects cannot be quantified precisely, we change colors of one to several objects and define the difficulty of each scene accordingly. As shown in Fig. 23, PTP is highly sensitive to both camera viewpoints and object colors: large values of camera Δyaw and Δpitch and significant different object colors result in uninformative VAE representation, PTP consequently performs poorly. Stable contrastive RL, which does not depend on VAE representation, outperforms PTP and is robust to a wide range of camera angles and object colors. These experiments provide preliminary evidence that contrastive representations might generalize reasonably well.

E.11 Additional Perceptual Losses

Prior work has found that auxiliary visual loss is necessary and important to drive an RL algorithm [71]. However, visualization and analysis in Fig. 2 suggest that stable contrastive RL is already acquiring useful representations, leading us to hypothesize that it may not require additional perception-specific losses. To test this hypothesis, we compare contrastive RL (with representations learned end-to-end) with a variant that uses a pre-trained VAE representation as the input for both actor and critic. We compare against PTP (which uses the same VAE representation) and an end-to-end variant of PTP. We use the task `push block, open drawer` for these experiments, as it demands the most temporally-extended reasoning to solve.

We compare all methods by training three random seeds of each of the four methods for 300K gradient steps on the offline setting. We measure performance by evaluating each seed in 10 rollouts. Fig. 24 *(Left)* shows the mean and standard deviation of these results. First, we find that stable contrastive RL achieves 53% higher success rates than the VAE-based version of contrastive RL, showing that contrastive RL does not require a perception specific loss, and instead suggesting that a good representation emerges purely from optimizing the contrastive RL objective. Intuitively, this result makes sense, as the contrastive RL critic objective already resembles existing representation learning objectives. The observation that adding the VAE objective *decreases* performance might

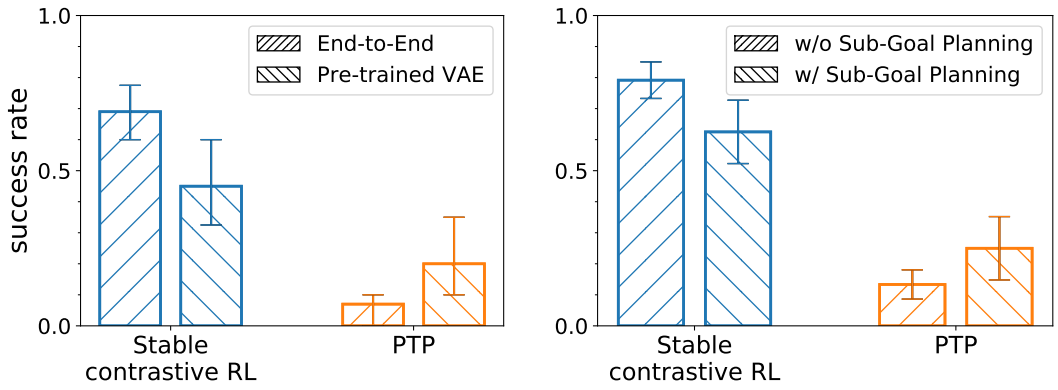


Figure 24: *(Left)* Additional perceptual losses are not needed for stable contrastive RL to outperform PTP. *(Right)* Without sub-goal planning, stable contrastive RL can solve temporal-extended tasks.

be explained by the misalignment between the VAE objective and the contrastive RL objective – directly optimizing the representations for the target task (learning a critic) is better than optimizing them for reconstruction (as done by the VAE). We conjecture that VAE representation could contain task-irrelevant noise that impairs policy learning. Second, we find that the VAE objective is important for the prior method, PTP; removing the VAE objective and learning the representations end-to-end decreases performance by 55%. This result supports the findings of the original PTP paper [18], while also illustrating that good representations can be learned in an end-to-end manner if the RL algorithm is chosen appropriately.

E.12 Sub-goal planning

Goal-conditioned RL often requires the agent to reason over long horizons. Prior works has proposed to combine critic learning with an explicit planner [56, 16, 18], using the learned value network as distance function to generate sub-goals. These *semi-parametric* methods can work really well in practice, but it remains unclear why *fully-parametric* model-free GCRL algorithms failed to perform similarly given the same training data. We have shown that stable contrastive RL learns a representation preserving causal information (Fig. 2) and a policy rearranging representations in temporal order (Fig. 19). So *can fully-parametric methods achieve comparable or better performance to semi-parametric methods with proper design decisions?*

To answer this question, we compare stable contrastive RL (a fully-parametric algorithm) with a variant that use explicit sub-goal planning. We compare against PTP (a semi-parametric method) and a variant of PTP without sub-goal planning. To make a fair comparison, we apply the same planning algorithm of PTP to the variant of stable contrastive RL. We choose the task `pick & place (table)` which requires the agent to pick and place a block to conduct our experiments. We train three random seeds of all the four methods for 300K gradient steps with the same offline dataset and evaluate the success rate over 10 rollouts, reporting the mean and standard deviation across seeds.

As shown in Fig. 24 *(Right)*, stable contrastive RL achieves a 30% absolute improvement over its semi-parametric variant, suggesting that stable contrastive RL does not require planner and, surprisingly, sub-goal planning could *hurt* the performance. We suspect that the policy can find a path in representation space that corresponds to an optimal trajectory in state space without additional supervision. The reason for adding planning module *decrease* performance might be explained by the fact that self-supervised critic objective does not imply a distance measure between the current state and the goal, and the growing computational complexity of planning. We observe that sub-goal planning is critical to improve the performance of PTP by 50%, which is consistent with the claim in the original PTP paper [18]. Nonetheless, neither version of PTP achieves comparable results to stable contrastive RL, showing that semi-parametric methods might not do well in a task with causal reasoning. Taken together, these experiments suggest that, with proper design decisions, a fully-parametric method like stable contrastive RL can at match (if not surpass) the performance of more complex semi-parametric methods that employ planning.

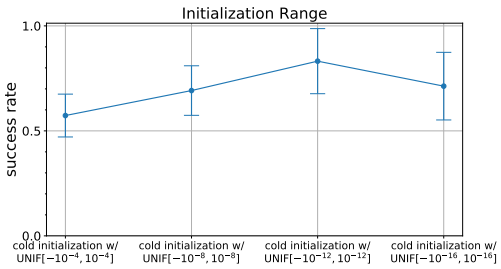


Figure 27: Smaller cold initialization (in a certain range) improves the performance of stable contrastive RL.

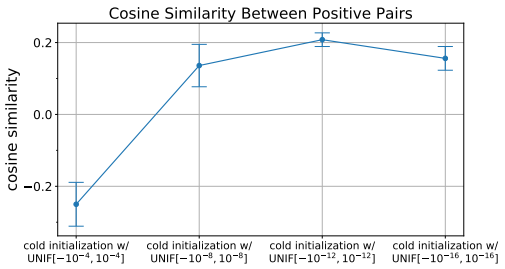


Figure 28: A fairly small cold initialization value encourages alignment.

E.13 Ablation Experiments: Batch Size

These ablation experiments test whether the batch size can affect the performance of stable contrastive RL and whether an ever larger batch size can induce an ever improvement in performance. To answer these questions, we vary the batch size from 512 to 4096 during training of stable contrastive RL and find the performance follows the batch size increasing paradigm as expected. However, the improvement stops after a certain threshold, indicating there might be other factors, e.g. network capacity and training steps [8], that dominate the performance after a batch size limit.

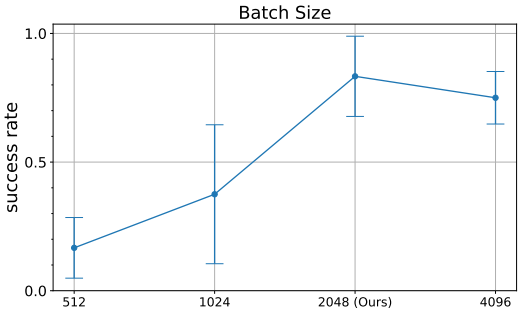


Figure 25: Larger batch size (in a certain range) improves the performance of stable contrastive RL.

E.14 Ablation Experiments: Cold Initialization

We run additional experiments to study whether cold initialization provides similar performance gains as learning rate schemes. We compare three methods: *(a)* cold initialization as mentioned in Sec. 2.2. *(b)* learning rate warmup, following the same warmup paradigm in [66] – linearly increasing the learning rate to 3×10^{-4} for the first 100K gradient steps and then decreasing it proportionally to the inverse square root of remaining gradient steps. *(c)* using a $10\times$ smaller learning rate for the last layers of $\phi(s, a)$ and $\psi(g)$. Our new experiments (Fig. 26) find that both *(b)* and *(c)* consistently perform worse than cold initialization.

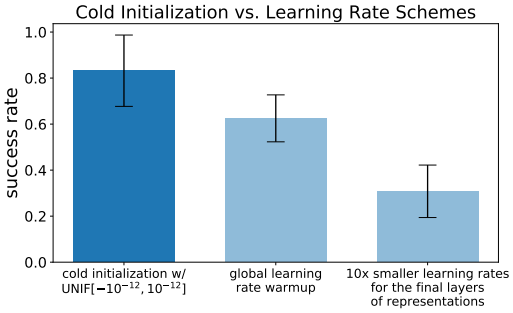


Figure 26: Cold initialization is more effective than learning rate schemes for stable contrastive RL.

To get a better understanding of the mechanism of cold initialization, we hypothesize that the range of initialization will affect the performance of stable contrastive RL. We conducted ablation experiments with initialization strategies $\textsc{Unif}[-10^{-4}, 10^{-4}]$, $\textsc{Unif}[-10^{-8}, 10^{-8}]$, $\textsc{Unif}[-10^{-12}, 10^{-12}]$, and $\textsc{Unif}[-10^{-16}, 10^{-16}]$ for the final layers of $\phi(s, a)$ and $\psi(g)$, plotting the success rates over 10 episodes and 3 random seeds in Fig. 27. The observation that small initialization ranges tend to perform better, with a value of 10^{-12} achieving 45% higher success rate than a (more standard) range of 10^{-4}, is consistent with our expectation that a fairly small initialization range incurs better performance. We note that the performance of the range 10^{-16} is $0.857\times$ (71.3% vs. 83.2%) than that of the range 10^{-12}, suggesting that an ever decreasing initialization range start to hinder representation learning after a certain value.

One effect of the cold initialization is that it changes the average distance between representations at random initialization. We next measure the average pairwise (cosine) distance between representa-

tions, for different initialization ranges. We randomly sample 10 validation batches and compute the average cosine similarities of positive pair at initialization as a function of different cold initialization ranges, showing results in Fig. 28. The finding that cosine similarities between positive examples decrease when we vary the initialization ranges from 10^{-4} to 10^{-12} explains the intuition that alignment between positive examples is crucial to contrastive representation learning. Of particular note is that using a very small initialization range 10^{-16} could force all representations to collapse, resulting in a worse performance. We observe that initialization strategies for which the representations are closer (higher cosine similarity) also result in higher success rates after training. This suggests that the average pairwise distance at initialization may be an effective way of selecting the initialization range *that does not require performing any training*.

E.15 Ablation Experiments: Contrastive Representation Dimension

While we used small representations (16 dimensional) in the main experiments, we now study how increasing the representation dimension affects the success rates. We ablate the dimension of contrastive representation in the set $\{16, 128, 512\}$, averaging success rates over 10 episodes of 3 random seeds. As shown in Fig. 29, representations of sizes 128 and 512 achieve considerably lower success rates, echoing prior work in finding that smaller representations yield better performance in the offline setting [17]. These results might also be explained by the increase of noise with a larger representation size, suggesting that the smaller representations effectively act as a sort of regularization and mitigate overfitting.

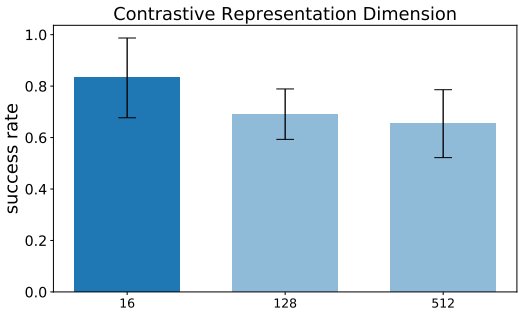


Figure 29: Smaller representation dimension benefits the learning of stable contrastive RL.