Contrastive Learning as Goal-Conditioned Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

In reinforcement learning (RL), it is easier to solve a task if given a good representation. While *deep* RL should automatically acquire such good representations, prior work often finds that learning representations in an end-to-end fashion is unstable and instead equip RL algorithms with additional representation learning parts (e.g., auxiliary losses, data augmentation). How can we design RL algorithms that directly acquire good representations? In this paper, instead of adding representation learning parts to an existing RL algorithm, we show (contrastive) representation learning methods can be cast as RL algorithms in their own right. To do this, we build upon prior work and apply contrastive representation learning to action-labeled trajectories, in such a way that the (inner product of) learned representations exactly corresponds to a goal-conditioned value function. We use this idea to reinterpret a prior RL method as performing contrastive learning, and then use the idea to propose a much simpler method that achieves similar performance. Across a range of goal-conditioned RL tasks, we demonstrate that contrastive RL methods achieve higher success rates than prior non-contrastive methods. We also show that contrastive RL outperforms prior methods on image-based tasks, without using data augmentation or auxiliary objectives.[1]

1 Introduction

Representation learning is an integral part of reinforcement learning (RL[2]) algorithms. While such representations might emerge from end-to-end training [7, 74, 113, 121], prior work has found it necessary to equip RL algorithms with perception-specific loss functions [29, 40, 66, 84, 86, 95, 110, 135] or data augmentations [64, 68, 110, 112], effectively decoupling the representation learning problem from the reinforcement learning problem. Given what prior work has shown about RL in the presence of function approximation and state aliasing [2, 130, 133], it is not surprising that end-to-end learning of representations is fragile [64, 68]: an algorithm needs good representations to drive the learning of the RL algorithm, but the RL algorithm needs to drive the learning of good representations. So, *can we design RL algorithms that do learn good representations, without the need for auxiliary perception losses?*

Rather than using a reinforcement learning algorithm to also solve a representation learning problem, we will use a representation learning algorithm to also solve certain types of reinforcement learning problems, namely goal-conditioned RL. Goal-conditioned RL is widely studied [6, 13, 20, 58, 75, 114], and intriguing from a representation learning perspective because it can be done in an entirely self-supervised manner, without manually-specified reward functions. We will focus on

[1] Project website with videos and code: `https://contrastive-rl.github.io/`

[2] RL = reinforcement learning, not representation learning.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

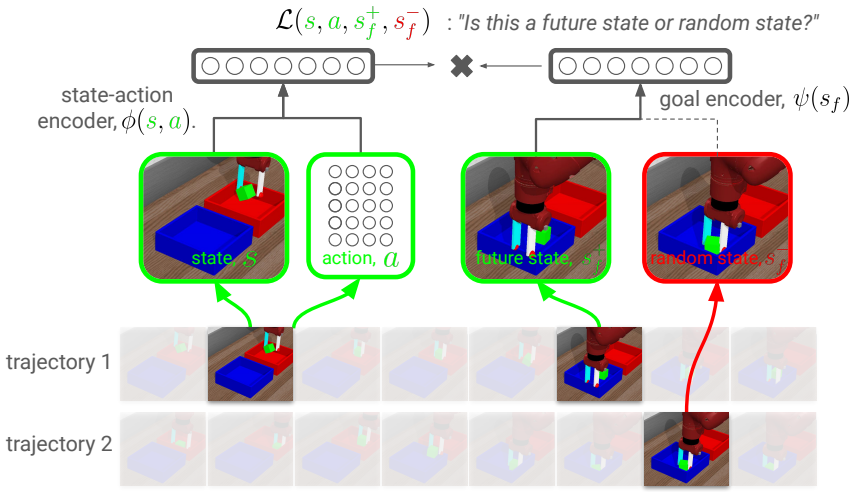


Figure 1: **Reinforcement learning via contrastive learning.** Our method uses contrastive learning to acquire representations of state-action pairs ($\phi(s, a)$) and future states ($\psi(s_f)$), so that the representations of future states are closer than the representations of random states. We prove that learned representations corresponds to a value function for a certain reward function. To select actions for reaching goal s_g, the policy chooses the action where $\phi(s, a)$ is closest to $\psi(s_g)$.

contrastive (representation) learning methods, using observations from the same trajectory (as done in prior work [90, 103]) while also including actions as an additional input (See Fig. 1). Intuitively, contrastive learning then resembles a goal-conditioned value function: nearby states have similar representations and unreachable states have dissimilar representations. We make this connection precise, showing that sampling positive pairs using the discounted state occupancy measure results in learning representations whose inner product exactly corresponds to a value function.

In this paper, we show how contrastive representation learning can be used to perform goal-conditioned RL. We formally relate the learned representations to reward maximization, showing that the inner product between representations corresponds to a value function. This framework of contrastive RL generalizes prior methods, such as C-learning [27], and suggests new goal-conditioned RL algorithms. One new method achieves performance similar to prior methods but is simpler; another method consistently outperforms the prior methods. On goal-conditioned RL tasks with image observations, contrastive RL methods outperform prior methods that employ data augmentation and auxiliary objectives, and do so without data augmentation or auxiliary objectives.

2 Related Work

This paper will draw a connection between RL and contrastive representation learning, building upon a long line of contrastive learning methods in NLP and computer vision, and deep metric learning [15, 49, 50, 50, 52, 72, 79, 81, 82, 89, 90, 102, 103, 107, 117, 124, 127]. Contrastive learning methods learn representations such that similar ("positive") examples have similar representations and dissimilar ("negative") examples have dissimilar representations.[3] While most methods generate the "positive" examples via data augmentation, some methods generate similar examples using different camera viewpoints of the same scene [103, 117], or by sampling examples that occur close in time within time series data [4, 90, 103, 112]. Our analysis will focus on this latter strategy, as the dependence on time will allow us to draw a precise relationship with the time dependence in RL.

Deep RL algorithms promise to automatically learn good representations, in an end-to-end fashion. However, prior work has found it challenging to uphold this promise [7, 74, 113, 121], prompting many prior methods to employ separate objectives for representation learning and RL [29, 40, 66, 84, 86, 95, 110, 112, 135]. Many prior methods choose a representation learning objectives that reconstruct the input state [29, 43, 45, 46, 66, 86, 88, 136] while others use contrastive representation learning methods [84, 90, 105, 110, 112]. Unlike these prior methods, we will not use a separate representation learning objective, but instead use the same objective for both representation learning and reinforcement learning. Some prior RL methods have also used contrastive learning to acquire reward functions [12, 18, 30, 35, 59, 63, 87, 128, 129, 140], often in imitation learning settings [34,

[3] Our focus will not be on recent methods that learn representations without negative samples [16, 39].

51]. In contrast, we will use contrastive learning to directly acquire a value function, which (unlike a reward function) can be used directly to take actions, without any additional RL.

This paper will focus on goal-conditioned RL problems, a problem prior work has approached using temporal difference learning [6, 27, 58, 75, 97, 100], conditional imitation learning [20, 37, 78, 99, 114], model-based methods [21, 101], hierarchical RL [85], and planning-based methods [28, 88, 99, 109]. The problems of automatically sampling goals and exploration [22, 32, 80, 93, 138] are orthogonal to this work. Like prior work, we will parametrize the value function as an inner product between learned representations [31, 54, 100]. Unlike these prior methods, we will learn a value function directly via contrastive learning, without using reward functions or TD learning.

Our analysis will be most similar to prior methods [9, 13, 27, 97] that view goal-conditioned RL as a data-driven problem, rather than as a reward-maximization problem. Many of these methods employ hindsight relabeling [6, 24, 58, 73], wherein experience is *relabeled* with an outcome that occurred in the future. Whereas hindsight relabeling is typically viewed as a trick to add on top of an RL algorithm, this paper can roughly be interpreted as showing that the hindsight relabeling is a standalone RL algorithm. Many goal-conditioned methods learn a value function that captures the similarity between two states [27, 58, 86, 120]. Such distance functions are structurally similar to the critic function learned for contrastive learning, a connection we make precise in Sec. 4. In fact, our analysis shows that C-learning [27] is already performing contrastive learning, and our experiments show that alternative contrastive RL methods can be much simpler and achieve higher performance.

Prior work has studied how representations related to reward functions using the framework of universal value functions [10, 100] and successor features [8, 48, 76]. While these methods typically require additional supervision to drive representation learning (manually-specified reward functions or features), our method is more similar to prior work that estimates the discounted state occupancy measure as an inner product between learned representations [9, 126]. While these methods use temporal difference learning, ours is akin to Monte Carlo learning. While Monte Carlo learning is often (but not always [21]) perceived as less sampling efficient, our experiments find that our approach can be as sample efficient as TD methods. Other prior work has focused on learning representations that can be used for planning [55, 77, 98, 99, 123]. Our method will learn representations using an objective similar to prior work [99, 103], but makes the key observation that the representation already encodes a value function: no additional planning or RL is necessary to choose actions.

Please see Appendix A for a discussion of how our work relates to unsupervised skill learning.

3 Preliminaries

Goal-conditioned reinforcement learning. The goal-conditioned RL problem is defined by states $s_t \in \mathcal{S}$, actions a_t, an initial state distribution $p_0(s)$, the dynamics $p(s_{t+1} \mid s_t, a_t)$, a distribution over goals $p_g(s_g)$, and a reward function $r_g(s, a)$ for each goal. This problem is equivalent to a multi-task RL [5, 41, 116, 125, 134], where tasks correspond to reaching goals states. Following prior work [9, 13, 27, 97], we define the reward as the probability (density) of reaching the goal at the next time step:[4]

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

This reward function is appealing because it avoids the need for a human user to specify a distance metric (unlike, e.g., [6]). Even though our method will not estimate the reward function, we will still use the reward function for analysis. For a goal-conditioned policy $\pi(a \mid s, s_g)$, we use $\pi(\tau \mid s_t)$ to denote the probability of sampling an infinite-length trajectory $\tau = (s_0, a_0, s_1, a_1, \cdots)$. We defined the expected reward objective and Q-function as

$$\max_\pi \mathbb{E}_{p_g(s_g), \pi(\tau \mid s_g)} \left[\sum_{t=0}^{\infty} \gamma^t r_g(s_t, a_t) \right], \quad Q^\pi_{s_g}(s, a) \triangleq \mathbb{E}_{\pi(\tau \mid s_g)} \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_g(s_{t'}, a_{t'}) \mid {}^{s_t = s,}_{a_t = a} \right]. \tag{2}$$

Intuitively, this objective corresponds to sampling a goal s_g and then optimizing the policy to go to that goal and stay there. Finally, we define the discounted state occupancy measure as [51, 137]

$$p^{\pi(\cdot \mid \cdot, s_g)}(s_{t+} = s) \triangleq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p_t^{\pi(\cdot \mid \cdot, s_g)}(s_t = s), \tag{3}$$

[4] At the initial state, this reward also includes the probability that the agent started at the goal: $r_g(s_0, a_0) = (1-\gamma)(p(s_1 = s_g \mid s_0, a_0) + p_0(s_0 = s_g))$

where $p_t^\pi(s)$ is the probability density over states that policy π visits after t steps. Sampling from the discounted state occupancy measure is easy: first sample a time offset from a geometric distribution ($t \sim \textsc{Geom}(1-\gamma)$), and then look at what state the policy visits after exactly t steps. We will use s_{t+} to denote states sampled from the discounted state occupancy measure. Because our method will combine experience collected from multiple policies, we also define the average stationary distribution as $p^{\pi(\cdot|\cdot)}(s_{t+} = s \mid s, a) \triangleq \int p^{\pi(\cdot|\cdot,s_g)}(s_{t+} = s \mid s, a) p^\pi(s_g \mid s, a) ds_g$, where $p^\pi(s_g \mid s, a)$ is the probability of the *commanded* goal given the current state-action pair. This stationary distribution is equivalent to that of the policy $\pi(a \mid s) \triangleq \int \pi(a \mid s, s_g) p^\pi(s_g \mid s) ds_g$ [139].

Contrastive representation learning. Contrastive representation learning methods [15, 42, 49, 50, 57, 72, 79, 81, 82, 117, 119, 124] take as input pairs of positive and negative examples, and learn representations so that positive pairs have similar representations and negative pairs have dissimilar representations. We use (u, v) to denote an input pair (e.g., u is an image and v is an augmented version of that image). Positive examples are sampled from a joint distribution $p(u, v)$ while negative examples are sampled from the product of marginal distributions, $p(u)p(v)$. We will use an objective based on binary classification [72, 81, 82, 89]. Let $f(u, v) = \phi(u)^T \psi(v)$ be the similarity between the representations of u and v. We will call f the *critic function*[5] and note that its range is $(-\infty, \infty)$. We will use NCE-binary [79] objective (also known as InfoMAX [50]):

$$\max_{f(u,v)} \mathbb{E}_{\substack{(u,v^+)\sim p(u,v) \\ v^- \sim p(u)}} \Big[\log \sigma(\underbrace{f(u, v^+)}_{\phi(u)^T\psi(v^+)}) + \log(1 - \sigma(\underbrace{f(u, v^-)}_{\phi(u)^T\psi(v^-)})) \Big]. \tag{4}$$

4 Contrastive Learning as an RL Algorithm

This section shows how to use contrastive representation to *directly* perform goal-conditioned RL. The key idea (Lemma 4.1) is that contrastive learning estimates the Q-function for a certain policy and reward function. To prove this result, we relate the Q-function to the state occupancy measure (Sec. 4.1) and then relate the optimal critic function to the state occupancy measure (Sec. 4.2).

This result allows us to propose a new algorithm for goal-conditioned RL based on contrastive learning. Unlike prior work, this algorithm is not adding contrastive learning on top of an existing RL algorithm. This framework generalizes C-learning [27], offering a cogent explanation for its good performance while also suggesting new methods that are simpler and can achieve higher performance.

4.1 Relating the Q-function to probabilities

This section sets the stage for the main results of this section by providing a probabilistic perspective goal-conditioned RL. The objective expected reward objective and associated Q-function in (Eq. 5) can equivalently be expressed as the probability (density) of reaching a goal in the future:

Proposition 1 (rewards $\rightarrow$ probabilities)**.** *The Q-function for the goal-conditioned reward function r_g (Eq. 1) is equivalent to the probability of state s_g under the discounted state occupancy measure:*

$$Q^\pi_{s_g}(s, a) = p^{\pi(\cdot|\cdot,s_g)}(s_{t+} = s_g \mid s, a). \tag{5}$$

The proof is in Appendix B. Translating rewards into probabilities not only makes it easier to analyze the goal-conditioned problem, but also means that any method for estimating probabilities (e.g., contrastive learning) can be turned into a method for estimating this Q-function.

4.2 Contrastive Learning Estimates a Q-Function

We will use contrastive learning to learn a value function by carefully choosing the inputs u and v. The first input, u, will correspond to a state-action pair, $u = (s_t, a_t) \sim p(s, a)$. In practice, these pairs are sampled from the replay buffer. Including the actions in the input is important because it will allow us to determine which actions to take to reach a desired future state. The second variable, v, is a *future* state, $v = s_f$. For the "positive" training pairs, the future state is sampled from the discounted state occupancy measure, $s_f \sim p^{\pi(\cdot|\cdot)}(s_{t+} \mid s_t, a_t)$. The "negative" training pairs, we sample a future

[5] In contrastive learning, the critic function indicates the similarity between a pair of inputs [94]; in RL, the critic function indicates the future expected returns [62]. Our method combines contrastive learning and RL in a way that these meanings become one and the same.

state from a random state-action pair: $s_f \sim p(s_{t+}) \triangleq \int p^{\pi(\cdot|\cdot)}(s_{t+} \mid s, a) p(s, a) ds da$. With these inputs, the contrastive learning objective (Eq. 4) can be written as

$$\max_f \mathbb{E}_{\substack{(s,a)\sim p(s,a), s_f^- \sim p(s_f) \\ s_f^+ \sim p^{\pi(\cdot|\cdot)}(s_{t+}|s_t,a_t)}} \left[\mathcal{L}(s, a, s_f^+, s_f^-)\right],$$
$$\text{where} \quad \mathcal{L}(s, a, s_f^+, s_f^-) \triangleq \log \sigma(\underbrace{f(s, a, s_f^+)}_{\phi(s,a)^T \psi(s_f^+)}) + \log(1 - \sigma(\underbrace{f(s, a, s_f^-)}_{\phi(s,a)^T \psi(s_f^-)})). \tag{6}$$

Intuitively, the critic function $f(u = (s_t, a_t), v = s_f)$ now tells us the correlation between the current state-action pair and future outcomes, analogous to a Q-function. We therefore can use the critic function in the same way as actor-critic RL algorithms [62], figuring which actions lead to the desired outcome. Because the Bayes-optimal critic function is a function of the state occupancy measure [79], $f^*(s, a, s_g) = \log\left(\frac{p^{\pi(\cdot|\cdot)}(s_{t+}=s_g|s,a)}{p(s_g)}\right)$, it can be used to express the Q-function:

Lemma 4.1. *The critic function that optimizes Eq. 6 is a Q-function for the goal-conditioned reward function (Eq. 1), up to a multiplicative constant* $\frac{1}{p(s_f)}$*:* $\exp(f^*(s, a, s_f)) = \frac{1}{p(s_f)} \cdot Q_{s_f}^{\pi(\cdot|\cdot)}(s, a)$.

The critic function can be viewed as an unnormalized density model, where $p(s_g)$ is the partition function. Much of the appeal of contrastive learning is it avoids estimating the partition function [42], which can be challenging; in the RL setting, it will turn out that this constant can be ignored when selecting actions. Our experiments show that learning a normalized density model works well when s_g is low-dimensional, but struggles to solve higher-dimensional tasks.

This lemma relates the critic function to $Q_{s_f}^{\pi(\cdot|\cdot)}(s, a)$, not $Q_{s_f}^{\pi(\cdot|\cdot, s_f)}(s, a)$. The underlying reason is that the critic function combines together experience collected when commanding different goals. Prior goal-conditioned behavioral cloning methods [20, 37, 78, 114] perform similar sharing, but do not analyze the relationship between the learned policies and Q functions. Sec. 4.5 shows that this critic can be used as the basis for a convergent RL algorithm, under some assumptions.

4.3 Learning the Goal-Conditioned Policy

The learned critic function not only tells us the likelihood of future states, but also tells us how different actions change the likelihood of a state occurring in the future. Thus, to learn a policy for reaching a goal state, we choose the actions that make that state most likely to occur in the future:

$$\max_{\pi(a|s,s_g)} \mathbb{E}_{\pi(a|s,s_g)p(s,s_g)}\left[f(s, a, s_f = s_g)\right] \approx \mathbb{E}_{\pi(a|s,s_g)p(s,s_g)}\left[\log Q_{s_g}^{\pi(\cdot|\cdot)}(s, a) - \log p(s_g)\right]. \tag{7}$$

The approximation above reflects errors in learning the optimal critic, and will allow us to prove that this policy loss corresponds to policy improvement in Sec. 4.5, under some assumptions.

In practice, we parametrize the goal-conditioned policy as a neural network that takes as input the state and goal and outputs a distribution over actions. The actor loss (Eq. 7) is computed by sampling states and *random* goals from the replay buffer, sampling actions from the policy, and then taking gradients on the policy using a reparametrization gradient. On tasks with image observations, we add an action entropy term to the policy objective.

4.4 A Complete Goal-Conditioned RL Algorithm

The complete algorithm alternates between fitting the critic function using contrastive learning, updating the policy using Eq. 7, and collecting more data. Alg. 1 provides a JAX [11] implementation of the actor and critic losses. Note that the critic is parameterized as an inner product between a representation of the state-action pair, and a representation of the goal state: $f(s, a, s_g) = \phi(s, a)^T \psi(s_g)$. This parametrization allows for efficiency computation, as we can compute the goal representations just once, and use them both in the positive pairs and the negative pairs. While this is common practice in representation learning, it is not exploited by most goal-conditioned RL algorithms. We refer to this method as `contrastive RL (NCE)`. In Appendix C, we derive a variant of this method (`contrastive RL (CPC)`) that uses the infoNCE bound on mutual information.

Contrastive RL (NCE) is an on-policy algorithm because it only estimates the Q-function for the policy that collected the data. However, in practice we take as many gradient steps on each transition

Algorithm 1 Contrastive RL (NCE): the actor and critic losses for our method.

```
from jax.numpy import einsum, eye
from optax import sigmoid_binary_cross_entropy
def critic_loss(states, actions, future_states):
  sa_repr = sa_encoder(states, actions)  # (batch_dim, repr_dim)
  g_repr = g_encoder(future_states)      # (batch_dim, repr_dim)
  logits = einsum('ik,jk->ij', sa_repr, g_repr)  # outer product: <sa_repr[i], g_repr[j]>
  return sigmoid_binary_cross_entropy(logits=logits, labels=eye(batch_size))

def actor_loss(states, future_states):
  actions = policy.sample(states, goal=future_states)  # (batch_size, action_dim)
  sa_repr = sa_encoder(states, actions)                # (batch_dim, repr_dim)
  g_repr = g_encoder(goals)                            # (batch_dim, repr_dim)
  logits = einsum('ik,ik->i', sa_repr, g_repr)  # inner product: <sa_repr[i], g_repr[i]>
  return -1.0 * logits
```

as standard off-policy RL algorithms [36, 44]. Please see Appendix E for full implementation details. We will also release an efficient implementation based on ACME [53] and JAX [11]. On a single TPUv2, training proceeds at $1,100\frac{\text{batches}}{\text{sec}}$ for state-based tasks and $105\frac{\text{batches}}{\text{sec}}$ for image-based tasks; for comparison, our implementation of DrQ on the same hardware setup runs at $28\frac{\text{batches}}{\text{sec}}$ ($3.9\times$ slower).[6] Architectures and hyperparameters are described in Appendix E.[7]

4.5 Convergence Guarantees

In general, providing convergence guarantees for methods that perform relabeling is challenging. Most prior work offers no such guarantees [6, 20, 21] or guarantees under only restrictive assumptions [37, 114]. Our convergence analysis will also make assumptions, though our experiments show that the method continues to perform well when those assumptions are violated.

Assumption 1. *Assume that each goal conditioned policy does not visit different goal states:*

$$p^{\pi(\cdot|\cdot,s_g)}(s_{t+} = s'_g) = 0 \quad \textit{for all} \quad (s_g \neq s'_g) \in \{s_g \mid p_g(s_g) > 0\}.$$

This assumption will only have to hold for the commanded goals, and is weaker than assuming that the policies visit disjoint states.

Assumption 2. *Further, assume that these goal-conditioned policies induce an average policy $\pi(a \mid s)$ that has a uniform distribution over subset of actions: $\pi(a \mid s) = \textsc{Unif}(\mathcal{A}(s) \subseteq \mathcal{A})$.*

Lemma 4.2 (Policy improvement)**.** *Let $p_g(s_g)$ be a distribution over a discrete set of goals and let $\pi(a \mid s, s_g)$ be a goal-conditioned policy. Assume that this goal distribution and goal-conditioned policy satisfy Assumptions 1 and 2, and that the critic is Bayes-optimal. Then the goal-conditioned policy obtained after one interation of contrastive RL, $\pi'(a \mid s, s_g)$, achieves higher rewards than the initial goal-conditioned policy:*

$$\mathbb{E}_{\pi'(\tau|s_g)}\left[\sum_{t=0}^{\infty}\gamma^t r_{s_g}(s_t, a_t)\right] \geq \mathbb{E}_{\pi(\tau|s_g)}\left[\sum_{t=0}^{\infty}\gamma^t r_{s_g}(s_t, a_t)\right] \qquad \textit{for all goals } s_g \in \{s_g \mid p_g(s_g) > 0\}.$$

In summary, we have shown that applying contrastive learning to a particular choice of inputs results in an RL algorithm, one that learns a Q-function and (under some assumptions) converges to the reward-maximizing policy. Contrastive RL (NCE) is simple: it does not require multiple Q-values [36], target Q networks [83], data augmentation [64, 68], or auxiliary objectives [110, 132].

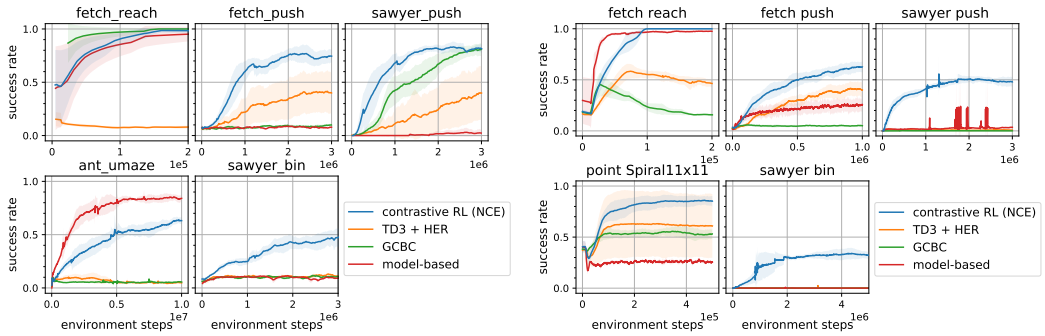
4.6 C-learning as Contrastive Learning

C-learning [27] is a special case of contrastive RL: it learns a critic function to distinguish future goals from random goals. Compared with contrastive RL (NCE), C-learning learns the classifier using temporal difference learning.[8] Viewing C-learning as a special case of contrastive RL suggests

[6] The more recent DrQ-v2 [131] uses on 1 NVIDIA V100 GPU to achieve a training speed of $96/4 = 24\frac{\text{batches}}{\text{sec}}$. The factor of 4 comes from an action repeat of 2 and an update interval of 2.

[7] Code and more results are available: `https://contrastive-rl.github.io/`

[8] The objectives are subtly different: C-learning estimates the probability that policy $\pi(\cdot \mid \cdot, s_g)$ visits state $s_f = s_g$, whereas contrastive RL (NCE) estimates the probability that any of the goal conditioned policies visit state s_f.



(a) State-based tasks

(b) Image-based tasks

Figure 2: **Goal-conditioned RL.** Contrastive RL (NCE) outperforms prior methods on most tasks. **Baselines**: `HER` [75] is a prototypical actor-critic method that uses hindsight relabeling [6]; `Goal-conditioned behavioral cloning (GCBC)` [20, 37, 78, 111] performs behavior cloning on relabeled experience; `model-based` fits a density model to the discounted state occupancy measure, similar on [19, 21, 56].

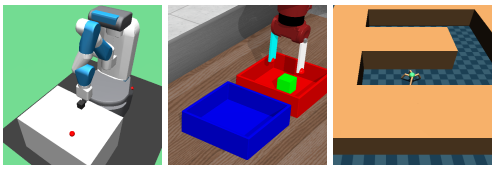
that contrastive RL algorithms might be implemented in a variety of different ways, each with relative merits. For example, `contrastive RL (NCE)` is much simpler than C-learning and tends to perform a bit better. Appendix D introduces another member of the contrastive RL family (`contrastive RL (NCE + C-learning)`) that tends to yield the best performance .

5 Experiments

Our experiments use goal-conditioned RL problems to compare contrastive RL algorithms to prior non-contrastive methods, including those that use data augmentation and auxiliary objectives. Appendices E, G, and H contain additional experiments, visualizations, and failed experiments.

5.1 Comparing to prior goal-conditioned RL methods

Baselines. We compare to three baselines. "HER" [75] is a goal-conditioned RL method that uses hindsight relabeling [6] with a high-performance actor-critic algorithm (TD3). This baseline is representative of a large class of prior work that uses hindsight relabeling [6, 71, 96, 100]. Like contrastive RL, this baseline does not assume access to a reward function. The second baseline is goal-conditioned behavioral cloning ("GCBC") [14, 20, 23, 37, 78, 91, 111, 114], which trains a policy to reach goal s_g by performing behavioral cloning on trajectories that reach state s_g. GCBC is a simple method that achieves excellent results [14, 23] and has the same inputs as our method ((s, a, s_f) triplets). The third baseline is a model-based approach that fits a density model to the future state distribution $p^{\pi(\cdot|\cdot)}(s_{t+} \mid s, a)$ and trains a goal-conditioned policy to maximize the probability of the commanded goal. This baseline is similar successor representations [19] and prior multi-step models [21, 56]. Both contrastive RL (Alg. 1) and this model-based approach encode the future state distribution, but the output dimension of this model-based method depends on the state dimension. We therefore expect this approach to excel in low-dimensional settings but struggle on image-based tasks. Where possible, we use the same architectures and hyperparameters for all methods. We will include additional representation learning baselines when studying representations in the subsequent section.



Tasks. We compare to a suite of goal-conditioned tasks, mostly taken from prior work. Four standard manipulation tasks include `fetch reach` and `fetch push` from Plappert et al. [92] and `sawyer push` and `sawyer bin` from Yu et al. [134]. We evaluate these tasks both with state-based observations and (unlike most prior work) image-based observations. The `sawyer bin` task poses an exploration challenge, as the agent must learn to pick up an object from one bin and place it at a goal location in another bin; the agent does not receive any reward shaping or demonstrations. We include two navigation

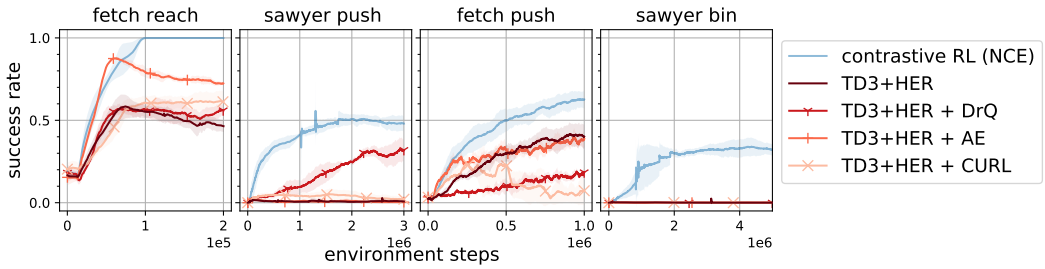


Figure 4: **Representation learning for image-based tasks.** While adding data augmentation and auxiliary representation objectives can boost the performance of the `TD3+HER` baseline, replacing the underlying goal-conditioned RL algorithm with one that resembles contrastive representation learning (i.e., ours) yields a larger increase in success rates. **Baselines**: `DrQ` [64] augments images and averages the Q-values across 4 augmentations; `auto encoder (AE)` adds an auxiliary reconstruction loss [29, 86, 88, 132]; `CURL` [110] applies RL on top of representations learned via augmentation-based contrastive learning.

tasks: `point Spiral11x11` is a 2D maze task with image observations and `ant umaze` [33] is a 111-dimensional locomotion task that presents a challenging low-level control problem. Where possible, we use the same initial state distribution, goal distribution, observations, and definition of success as prior work. Goals have the same dimension as the states, with one exception: on the `ant umaze` task, we used the global XY position as the goal. We illustrate three of the tasks to the right. The agent does not have access to any ground truth reward function in any of these tasks.

We report results in Fig. 2, using five random seeds for each experiment and plotting the mean and standard deviation across those random seeds. One the state-based tasks (Fig. 2a), most methods solve the easiest task (`fetch reach`) while only our method solves the most challenging task (`sawyer bin`). Our method also outperforms all prior methods on the two pushing tasks. The model-based baseline performs best on the `ant umaze` task, likely because learning a model is relatively easy when the goal is lower-dimensional goal (just the XY location). On the image-based tasks (Fig. 2b), most methods make progress on the two easiest tasks (`fetch reach` and `point Spiral11x11`); our method outperforms the baselines on the three more challenging tasks. Of particular note is the success on `sawyer push` and `sawyer bin`: while the success rate of our method remains below 50%, no baselines make any progress on learning these tasks. Taken together, these results suggest that contrastive RL (NCE) is a competitive goal-conditioned RL algorithm.

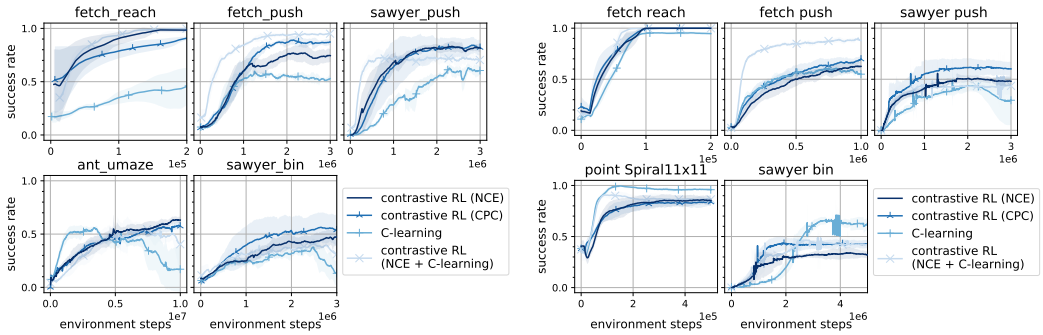
5.2 Comparing to prior representation learning methods

We hypothesize that contrastive RL may automatically learn good representations. To test this hypothesis, we compare contrastive RL (NCE) to techniques proposed by prior work for representation learning. These include data augmentation [64, 68, 131] ("DrQ") and auxiliary objectives based on an autoencoder [29, 86, 88, 132] ("AE") and a contrastive learning objective ("CURL") that generates positive examples using data augmentation, similar to prior work [84, 110, 112]. Because prior work has demonstrated these techniques in combination with actor-critic RL algorithms, we will use these techniques in combination with the actor-critic baseline from the previous section ("TD3 + HER"). While contrastive RL (NCE) resembles a contrastive representation learning method, it does not include any data augmentation or auxiliary representation learning objectives.

We show results in Fig. 4, with error bars again showing the mean and standard deviation across 5 random seeds. While adding the autoencoder improves the baseline on the `fetch reach` and adding DrQ improves the baseline on the `sawyer push`, contrastive RL (NCE) outperforms the prior methods on all tasks. Unlike these methods, contrastive RL does not use auxiliary objectives or additional domain knowledge in the form of image-appropriate data augmentations. These experiments do not show that representation learning is never useful, and do not show that contrastive RL cannot be improved with additional representation learning machinery. Rather, they show that designing RL algorithms that structurally resemble contrastive representation learning yields bigger improvements than simply adding representation learning tricks on top of existing RL algorithms.

5.3 Probing the dimensions of contrastive RL

Up to now, we have focused on the specific instantiation of contrastive RL spelled out in Alg. 1. However, there is a whole family of RL algorithms with contrastive characteristics. C-learning is a contrastive RL algorithm that uses temporal difference learning (Sec. 4.6). Contrastive RL (CPC) is



(a) state-based observations (b) image-based observations

Figure 5: **Contrastive RL design decisions.** Generalizing C-learning to a family of contrastive RL algorithms allowed us to identify algorithms that are much simpler (contrastive RL (NCE)) and that consistently achieve higher performance (contrastive RL (NCE + C-learning)).

a variant of Alg. 1 based on the infoNCE objective [90] that we derive in Appendix C Contrastive RL (NCE + C-learning) is variant that combines C-learning with Alg. D (see Appendix D.). The aim of these experiments are to study whether generalizing C-learning to a family of contrastive RL algorithms was useful: do the simpler methods achieve similar performance, and do other methods achieve better performance?

We present results in Fig. 5, again plotting the mean and standard deviation across five random seeds. Contrastive RL (CPC) outperforms contrastive RL (NCE) on three, suggesting that swapping one mutual information estimator for another can sometimes improve performance, though both estimators can be effective. C-learning outperforms contrastive RL (NCE) on three tasks but performs worse on other tasks. Contrastive RL (NCE + C-learning) consistently ranks among the best methods. These experiments demonstrate that the prior contrastive RL method, C-learning [27], achieves good results on most tasks; generalizing C-learning to a family of contrastive RL algorithms resulting in new algorithms that achieve higher performance and can be much simpler.

5.4 Linear regression with the learned features

To study the learned representations in isolation we take the state-action representations $\phi(s, a)$ trained on the image-based `point NineRooms` task, and run a linear probe [3, 47] experiment to see whether the representations have learned to encode task-relevant information (the shortest path distance to the goal). As shown in Fig. 6, contrastive RL (NCE) learns representations that achieve lower test error than those learned by TD3+HER and by a random CNN encoder.

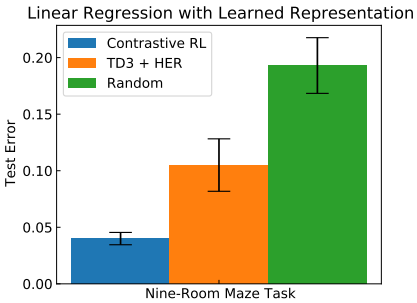


Figure 6: **Linear probe experiment.**

6 Conclusion

In this paper, we showed how contrastive representation learning can be used for goal-conditioned RL. This connection not only lets us re-interpret a prior RL method as performing contrastive learning, but also suggests a family of contrastive RL methods, which includes simpler algorithms, as well as algorithms that attain better overall performance. While this paper might be construed to imply that RL is more or less important than representation learning [67, 70, 106, 108], we have a different takeaway: that it may be enough to build RL algorithms that *look like* representation learning.

One limitation of this work is that it looks only at the goal-conditioned RL problems. How these methods might be applied to arbitrary RL problems remains an open problem, though we note that recent algorithms for this setting [26] already bear a resemblance to contrastive RL. Whether the rich set of ideas from contrastive learning might be used to construct even better RL algorithms likewise remains an open question.

References

[1] Achiam, J., Edwards, H., Amodei, D., and Abbeel, P. (2018). Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*.

[2] Achiam, J., Knight, E., and Abbeel, P. (2019). Towards characterizing divergence in deep q-learning. *arXiv preprint arXiv:1903.08894*.

[3] Alain, G. and Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

[4] Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M.-A., and Hjelm, R. D. (2019). Unsupervised state representation learning in atari. *Advances in Neural Information Processing Systems*, 32.

[5] Andreas, J., Klein, D., and Levine, S. (2017). Modular multitask reinforcement learning with policy sketches. In *International Conference on Machine Learning*, pages 166–175. PMLR.

[6] Andrychowicz, M., Crow, D., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. (2017). Hindsight experience replay. In *NeurIPS*.

[7] Annasamy, R. M. and Sycara, K. (2019). Towards better interpretability in deep q-networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4561–4569.

[8] Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. (2017). Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30.

[9] Blier, L., Tallec, C., and Ollivier, Y. (2021). Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*.

[10] Borsa, D., Barreto, A., Quan, J., Mankowitz, D., Munos, R., Van Hasselt, H., Silver, D., and Schaul, T. (2018). Universal successor features approximators. *arXiv preprint arXiv:1812.07626*.

[11] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

[12] Brown, D., Goo, W., Nagarajan, P., and Niekum, S. (2019). Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pages 783–792. PMLR.

[13] 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.

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

[15] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. (2020). A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

[16] Chen, X. and He, K. (2021). Exploring simple siamese representation learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15745–15753.

[17] Choi, J., Sharma, A., Lee, H., Levine, S., and Gu, S. S. (2021). Variational empowerment as representation learning for goal-conditioned reinforcement learning. In *International Conference on Machine Learning*, pages 1953–1963. PMLR.

[18] Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*.

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

[20] Ding, Y., Florensa, C., Abbeel, P., and Phielipp, M. (2019). Goal-conditioned imitation learning. *Advances in Neural Information Processing Systems*, 32:15324–15335.

[21] Dosovitskiy, A. and Koltun, V. (2016). Learning to act by predicting the future. *arXiv preprint arXiv:1611.01779*.

[22] Du, Y., Gan, C., and Isola, P. (2021). Curious representation learning for embodied intelligence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10408–10417.

[23] 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*.

[24] Eysenbach, B., Geng, X., Levine, S., and Salakhutdinov, R. (2020). Rewriting history with inverse RL: Hindsight inference for policy improvement. *ArXiv*, abs/2002.11089.

[25] 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*.

[26] Eysenbach, B., Levine, S., and Salakhutdinov, R. R. (2021a). Replacing rewards with examples: Example-based policy search via recursive classification. *Advances in Neural Information Processing Systems*, 34.

[27] Eysenbach, B., Salakhutdinov, R., and Levine, S. (2021b). C-learning: Learning to achieve goals via recursive classification. *ArXiv*, abs/2011.08909.

[28] 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.

[29] Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. (2016). Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE.

[30] Fischinger, D., Vincze, M., and Jiang, Y. (2013). Learning grasps for unknown objects in cluttered scenes. In *2013 IEEE international conference on robotics and automation*, pages 609–616. IEEE.

[31] Florensa, C., Degrave, J., Heess, N., Springenberg, J. T., and Riedmiller, M. (2019). Self-supervised learning of image embedding for continuous control. *arXiv preprint arXiv:1901.00943*.

[32] 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.

[33] Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. (2020). D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.

[34] Fu, J., Luo, K., and Levine, S. (2017). Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*.

[35] Fu, J., Singh, A., Ghosh, D., Yang, L., and Levine, S. (2018). Variational inverse control with events: A general framework for data-driven reward definition. In *NeurIPS*.

[36] 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.

[37] 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*.

[38] Gregor, K., Rezende, D. J., and Wierstra, D. (2016). Variational intrinsic control. *arXiv preprint arXiv:1611.07507*.

[39] Grill, J.-B., Strub, F., Altch'e, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. Á., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv*, abs/2006.07733.

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

[41] 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.

[42] 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).

[43] Ha, D. and Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*.

[44] 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.

[45] Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. (2019a). Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.

[46] Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. (2019b). Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR.

[47] Han, T., Xie, W., and Zisserman, A. (2020). Self-supervised co-training for video representation learning. *Advances in Neural Information Processing Systems*, 33:5679–5690.

[48] Hansen, S., Dabney, W., Barreto, A., Van de Wiele, T., Warde-Farley, D., and Mnih, V. (2019). Fast task inference with variational intrinsic successor features. *arXiv preprint arXiv:1906.05030*.

[49] 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.

[50] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2018). Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

[51] Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573.

[52] Hoffer, E. and Ailon, N. (2015). Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer.

[53] Hoffman, M., Shahriari, B., Aslanides, J., Barth-Maron, G., Behbahani, F., Norman, T., Abdolmaleki, A., Cassirer, A., Yang, F., Baumli, K., Henderson, S., Novikov, A., Colmenarejo, S. G., Cabi, S., Gulcehre, C., Paine, T. L., Cowie, A., Wang, Z., Piot, B., and de Freitas, N. (2020). Acme: A research framework for distributed reinforcement learning. *arXiv preprint arXiv:2006.00979*.

[54] Hong, Z.-W., Yang, G., and Agrawal, P. (2022). Bilinear value networks. *arXiv preprint arXiv:2204.13695*.

[55] Ichter, B., Sermanet, P., and Lynch, C. (2020). Broadly-exploring, local-policy trees for long-horizon task planning. *arXiv preprint arXiv:2010.06491*.

[56] Janner, M., Mordatch, I., and Levine, S. (2020). gamma-models: Generative temporal difference learning for infinite-horizon prediction. *Advances in Neural Information Processing Systems*, 33:1724–1735.

[57] Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

[58] Kaelbling, L. P. (1993). Learning to achieve goals. In *IJCAI*, pages 1094–1099. Citeseer.

[59] Kalashnikov, D., Varley, J., Chebotar, Y., Swanson, B., Jonschkowski, R., Finn, C., Levine, S., and Hausman, K. (2021). Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *ArXiv*, abs/2104.08212.

[60] Kish, L. (1965). Survey sampling. new york: John wiley& sons. *Inc. KishSurvey Sampling1965*.

[61] Klingemann, M. (2016). Raster fairy. `https://github.com/bmcfee/RasterFairy`.

[62] Konda, V. and Tsitsiklis, J. (1999). Actor-critic algorithms. *Advances in neural information processing systems*, 12.

[63] Konyushkova, K., Zolna, K., Aytar, Y., Novikov, A., Reed, S., Cabi, S., and de Freitas, N. (2020). Semi-supervised reward learning for offline reinforcement learning. *arXiv preprint arXiv:2012.06899*.

[64] Kostrikov, I., Yarats, D., and Fergus, R. (2020). Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*.

[65] Kumar, A., Agarwal, R., Ghosh, D., and Levine, S. (2020). Implicit under-parameterization inhibits data-efficient deep reinforcement learning. *arXiv preprint arXiv:2010.14498*.

[66] Lange, S. and Riedmiller, M. (2010). Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[67] Langford, J. (2010). Specializations of the master problem.

[68] Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. (2020). Reinforcement learning with augmented data. *Advances in Neural Information Processing Systems*, 33:19884–19895.

[69] Laskin, M., Liu, H., Peng, X. B., Yarats, D., Rajeswaran, A., and Abbeel, P. (2021). CIC: Contrastive intrinsic control for unsupervised skill discovery. In *Deep RL Workshop NeurIPS 2021*.

[70] LeCun, Y. (2016). Predictive learning. `https://www.youtube.com/watch?v=0unt2Y4qxQo`. Keynote Talk.

[71] Levy, A., Konidaris, G., Platt, R., and Saenko, K. (2017). Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*.

[72] Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27.

[73] Li, A., Pinto, L., and Abbeel, P. (2020). Generalized hindsight for reinforcement learning. *Advances in neural information processing systems*, 33:7754–7767.

[74] Liang, Y., Machado, M. C., Talvitie, E., and Bowling, M. (2015). State of the art control of atari games using shallow reinforcement learning. *arXiv preprint arXiv:1512.01563*.

[75] Lin, X., Baweja, H. S., and Held, D. (2019). Reinforcement learning without ground-truth state. *ArXiv*, abs/1905.07866.

[76] Liu, H. and Abbeel, P. (2021). Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pages 6736–6747. PMLR.

[77] Liu, K., Kurutach, T., Tung, C., Abbeel, P., and Tamar, A. (2020). Hallucinative topological memory for zero-shot visual planning. In *International Conference on Machine Learning*, pages 6259–6270. PMLR.

[78] 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.

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

[80] 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.

[81] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

[82] Mnih, A. and Teh, Y. W. (2012). A fast and simple algorithm for training neural probabilistic language models. In *ICML*.

[83] 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*.

[84] Nachum, O., Gu, S., Lee, H., and Levine, S. (2018a). Near-optimal representation learning for hierarchical reinforcement learning. In *International Conference on Learning Representations*.

[85] Nachum, O., Gu, S. S., Lee, H., and Levine, S. (2018b). Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31.

[86] 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*, 31:9191–9200.

[87] Nair, S., Mitchell, E., Chen, K., Savarese, S., Finn, C., et al. (2022). Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning*, pages 1303–1315. PMLR.

[88] Nasiriany, S., Pong, V. H., Lin, S., and Levine, S. (2019). Planning with goal-conditioned policies. In *NeurIPS*.

[89] Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29.

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

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

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

[93] Pong, V. H., Dalal, M., Lin, S., Nair, A., Bahl, S., and Levine, S. (2019). Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*.

[94] Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. (2019). On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR.

[95] Rakelly, K., Gupta, A., Florensa, C., and Levine, S. (2021). Which mutual-information representation learning objectives are sufficient for control? *ArXiv*, abs/2106.07278.

[96] Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degrave, J., Wiele, T., Mnih, V., Heess, N., and Springenberg, J. T. (2018). Learning by playing solving sparse reward tasks from scratch. In *International conference on machine learning*, pages 4344–4353. PMLR.

[97] 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.

[98] Rybkin, O., Zhu, C., Nagabandi, A., Daniilidis, K., Mordatch, I., and Levine, S. (2021). Model-based reinforcement learning via latent-space collocation. In *International Conference on Machine Learning*, pages 9190–9201. PMLR.

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

[100] Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR.

[101] 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.

[102] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

[103] Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., and Brain, G. (2018). Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE.

[104] Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. (2019). Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*.

[105] Shu, R., Nguyen, T., Chow, Y., Pham, T., Than, K., Ghavamzadeh, M., Ermon, S., and Bui, H. (2020). Predictive coding for locally-linear control. In *International Conference on Machine Learning*, pages 8862–8871. PMLR.

[106] Silver, D., Singh, S., Precup, D., and Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, 299:103535.

[107] Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*.

[108] Srinivas, A. and Abbeel, P. (2021). Unsupervised learning for reinforcement learning. Tutorial.

[109] Srinivas, A., Jabri, A., Abbeel, P., Levine, S., and Finn, C. (2018). Universal planning networks. *ArXiv*, abs/1804.00645.

[110] Srinivas, A., Laskin, M., and Abbeel, P. (2020). Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*.

[111] 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*.

[112] Stooke, A., Lee, K., Abbeel, P., and Laskin, M. (2021). Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pages 9870–9879. PMLR.

[113] Such, F. P., Madhavan, V., Liu, R., Wang, R., Castro, P. S., Li, Y., Zhi, J., Schubert, L., Bellemare, M. G., Clune, J., et al. (2018). An atari model zoo for analyzing, visualizing, and comparing deep reinforcement learning agents. *arXiv preprint arXiv:1812.07069*.

[114] 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.

[115] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[116] Teh, Y., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., and Pascanu, R. (2017). Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, 30.

[117] Tian, Y., Krishnan, D., and Isola, P. (2020). Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer.

[118] Tsai, Y.-H., Zhao, H., Yamada, M., Morency, L.-P., and Salakhutdinov, R. (2020). Neural methods for point-wise dependency estimation. In *Proceedings of the Neural Information Processing Systems Conference (Neurips)*.

[119] Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., and Lucic, M. (2019). On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*.

[120] Venkattaramanujam, S., Crawford, E., Doan, T. V., and Precup, D. (2019). Self-supervised learning of distance functions for goal-conditioned reinforcement learning. *ArXiv*, abs/1907.02998.

[121] Wang, H., Miahi, E., White, M., Machado, M. C., Abbas, Z., Kumaraswamy, R., Liu, V., and White, A. (2022). Investigating the properties of neural network representations in reinforcement learning. *arXiv preprint arXiv:2203.15955*.

[122] Warde-Farley, D., Van de Wiele, T., Kulkarni, T., Ionescu, C., Hansen, S., and Mnih, V. (2018). Unsupervised control through non-parametric discriminative rewards. *arXiv preprint arXiv:1811.11359*.

[123] Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. (2015). Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28.

[124] Weinberger, K. Q. and Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *NIPS*.

[125] Wilson, A., Fern, A., Ray, S., and Tadepalli, P. (2007). Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*, pages 1015–1022.

[126] Wu, Y., Tucker, G., and Nachum, O. (2018a). The laplacian in rl: Learning representations with efficient approximations. *arXiv preprint arXiv:1810.04586*.

[127] Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018b). Unsupervised feature learning via non-parametric instance discrimination. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742.

[128] Xie, A., Singh, A., Levine, S., and Finn, C. (2018). Few-shot goal inference for visuomotor learning and planning. In *Conference on Robot Learning*, pages 40–52. PMLR.

[129] Xu, D. and Denil, M. (2019). Positive-unlabeled reward learning. *arXiv preprint arXiv:1911.00459*.

[130] Yang, G., Ajay, A., and Agrawal, P. (2021). Overcoming the spectral bias of neural value approximation. In *International Conference on Learning Representations*.

[131] Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. (2021a). Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*.

[132] Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., and Fergus, R. (2021b). Improving sample efficiency in model-free reinforcement learning from images. In *AAAI*.

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

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

[135] Zhang, A., McAllister, R. T., Calandra, R., Gal, Y., and Levine, S. (2020a). Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*.

[136] Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M., and Levine, S. (2019). Solar: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, pages 7444–7453. PMLR.

[137] Zhang, S., Liu, B., and Whiteson, S. (2020b). Gradientdice: Rethinking generalized offline estimation of stationary values. In *International Conference on Machine Learning*, pages 11194–11203. PMLR.

[138] Zhao, R., Sun, X., and Tresp, V. (2019). Maximum entropy-regularized multi-goal reinforcement learning. In *International Conference on Machine Learning*, pages 7553–7562. PMLR.

[139] Ziebart, B. D. (2010). *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University.

[140] Zolna, K., Reed, S., Novikov, A., Colmenarejo, S. G., Budden, D., Cabi, S., Denil, M., de Freitas, N., and Wang, Z. (2019). Task-relevant adversarial imitation learning. *arXiv preprint arXiv:1910.01077*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The main claims are that (1) contrastive learning can be used to learn a Q-function (Proof in Appendix B) and that (2) contrastive RL methods can outperform non-contrastive RL algorithms on goal-conditioned RL tasks (results in Fig. 2).
   (b) Did you describe the limitations of your work? **[TODO]**
   (c) Did you discuss any potential negative societal impacts of your work? [No] While RL broadly might be used for applications with both positive and negative outcomes, our algorithmic contributions are not tied to any particular application.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] E.g., Assumptions 1 and 2
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We have included all experimental details in Appendix E; code will be released upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix E.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All figures show 5 random seeds, witht error bars corresponding to the mean and standard deviation across these seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Sec. 4.4 describes the training speed on one TPUv2.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]