Dense Dynamics-Aware Reward Synthesis:
Integrating Prior Experience with Demonstrations

Cevahir Koprulu[1][*]　　　　　　　　　　　　　　　　CEVAHIR.KOPRULU@UTEXAS.EDU
Po-han Li[1][*]　　　　　　　　　　　　　　　　　　　　POHANLI@UTEXAS.EDU
Tianyu Qiu[1][*]　　　　　　　　　　　　　　　　　　　TIANYUQIU@UTEXAS.EDU
Ruihan Zhao[1]　　　　　　　　　　　　　　　　　　　RUIHAN.ZHAO@UTEXAS.EDU
Tyler Westenbroek[2]　　　　　　　　　　　　　　　WESTENBROEKT@GMAIL.COM
David Fridovich-Keil[1]　　　　　　　　　　　　　　　　　DFK@UTEXAS.EDU
Sandeep Chinchali[1]　　　　　　　　　　　　　　　SANDEEPC@UTEXAS.EDU
Ufuk Topcu[1]　　　　　　　　　　　　　　　　　　　UTOPCU@UTEXAS.EDU
[1]*University of Texas at Austin,*　　[2]*University of Washington*

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

Many continuous control problems can be formulated as sparse-reward reinforcement learning (RL) tasks. In principle, online RL methods can automatically explore the state space to solve each new task. However, discovering sequences of actions that lead to a non-zero reward becomes exponentially more difficult as the task horizon increases. Manually shaping rewards can accelerate learning for a fixed task, but it is an arduous process that must be repeated for each new environment. We introduce a systematic reward-shaping framework that distills the information contained in 1) a task-agnostic prior data set and 2) a small number of task-specific expert demonstrations, and then uses these priors to synthesize dense dynamics-aware rewards for the given task. This supervision substantially accelerates learning in our experiments, and we provide analysis demonstrating how the approach can effectively guide online learning agents to faraway goals.

Keywords: Imitation Learning, Learning from Demonstrations, Reward Shaping.

1. Introduction

Intelligent decision-making requires temporally extended reasoning across a multitude of environments. In principle, established exploration techniques for online deep reinforcement learning (RL) (Haarnoja et al., 2018; Schulman et al., 2017; Chentanez et al., 2004; Bellemare et al., 2016) can achieve the spatial data coverage needed to carry out each new task. However, exploration for long-horizon tasks, especially those with sparse rewards, remains a significant bottleneck. In contrast, offline RL attempts to bootstrap learning from diverse data sets (Kumar et al., 2020; Janner et al., 2021). While both paradigms can benefit from amortizing learning across many tasks (Chane-Sane et al., 2021; Andrychowicz et al., 2017), solving novel long-horizon tasks remains challenging.

Learning from Demonstrations (LfD) frameworks can leverage expert trajectories to learn effective actions over long horizons. However, they struggle to obtain the necessary spatial data coverage to learn robust policies. This is due to compounding errors, wherein small differences in initial conditions or actions cause rollouts to rapidly diverge from the expert training distribution (Ke et al.,

---

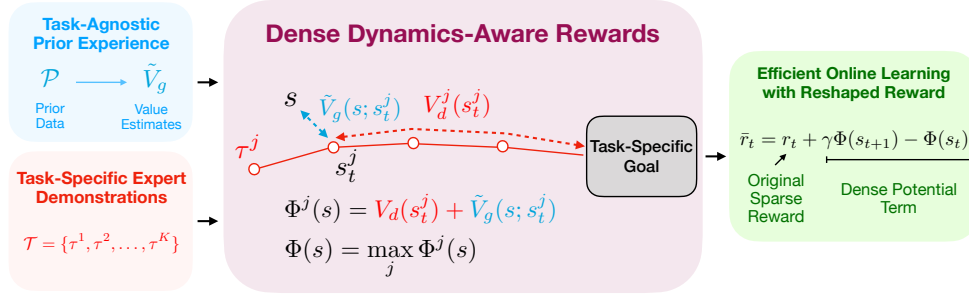[*]Equal contribution. Supplementary material can be found at https://tinyurl.com/mtz5cn2r.

Figure 1: A depiction of our framework. (Cyan) We distill information about the dynamics of a prior data set into a goal-conditioned value estimator $\tilde{V}_g$. (Red) An expert provides demonstrations for a new target task. (Purple) The framework combines them to construct a potential function $\Phi(s)$ (7), implicitly estimating the number of steps needed to reach the task-specific goal from state s. For this estimate, $\Phi^j(s)$ first measures (6) the number of steps needed to reach the j-th demonstration τ^j via $\tilde{V}_g$, and then follow it to the goal via V_d^j (5). (Green) We use the overall potential to synthesize dense dynamics-aware rewards for the target task.

2023; Ross et al., 2011; Block et al., 2023). Thus, LfD techniques may require an impractical number of expensive expert demonstrations (Ross et al., 2011). Low-quality or sub-optimal demonstrations exacerbate this challenge, degrading the performance of existing methods.

This paper introduces a novel reward-shaping framework that embraces the strengths of these paradigms. Our approach, see Figure 1, combines task-agnostic prior experience with a handful of task-specific expert demonstrations to synthesize dense dynamics-aware rewards that substantially accelerate online exploration and learning. Our approach encodes information about the dynamics during an initial pre-training phase by learning a goal-conditioned value function from task-agnostic data. Then, given a few temporally extended demonstrations for a new task, we leverage the potential-based reward-shaping (PBRS) formalism (Ng et al., 1999) to construct a potential with two terms: 1) a term incentivizing the agent to reach the nearest point on a demonstration, via the goal-conditioned value function, 2) a term encouraging the agent to follow the demonstration from this point to the task-specific goal. The shaped reward provides dense supervision, biasing the exploration towards nearby expert demonstrations to faraway goals. In summary, our approach leverages expert demonstrations for temporally extended reasoning, task-agnostic data to 'fill the gaps' around demonstrations with dense rewards, and online learning to smooth over potential deficiencies in these data sources. Beyond accelerating learning, our approach has the following benefits:

Overcoming Sub-optimal Demonstrations and Dynamics Shift: The potential in PBRS can be viewed as an approximation to the optimal value function of the original objective (Ng et al., 1999). While the reshaped reward guides the exploration, the new objective preserves the optimal policies for the original problem. In our setting, the potential will be a sub-optimal approximation when the demonstrations are poor. Similarly, when dynamics shift between the prior data and the target task, the goal-conditioned value functions may be inaccurate. Nonetheless, as our experiments demonstrate, our approach still accelerates learning optimal policies in these scenarios.

Learning from (Partial) Observations: In practice, the agent may not have access to the expert's actions, only partial observations, and may even need to infer states from high-dimensional observations such as visual inputs. For example, when demonstrating how to push an object across a table, a human expert may use their hands rather than teleoperating a robot arm. Our reward-shaping approach only requires the expert's states, and we demonstrate how the approach extends to partial (and potentially high-dimensional) observations. However, because we implement our approach

with off-policy RL, when actions are available, we benefit from including expert transitions in the replay buffer (Nair et al., 2018; Rajeswaran et al., 2017).

Shortening Learning Horizons: During the initial pre-training phase, we focus on learning accurate goal-conditioned value estimates for goals that take fewer steps to reach in comparison to temporally extended expert demonstrations. Shortening the learning horizon enables us to leverage tractable amounts of prior data effectively while reusing it for many downstream tasks. We also investigate the effects of annealing the discount factor (Cheng et al., 2021) for the reshaped objective. Theoretically and empirically, we show how using a small discount factor early in training will cause the agent to greedily exploit the prior information baked into the dense rewards during initial exploration. Our experiments indicate this can further accelerate learning by a substantial margin.

2. Related Work

Overcoming Distribution Shift in Imitation Learning: A challenge for imitation learning is distribution shift, wherein small errors compound over time and cause the state distribution generated by the agent to diverge from that of the expert. Techniques such as DAGGER (Ross et al., 2011) and DART (Laskey et al., 2017) use an interactive expert model to query expert trajectories that recover from mistakes made by the agent; however, this process may be overly arduous or impossible to implement in many practical scenarios. Frameworks such as GAIL (Ho and Ermon, 2016), SQIL (Reddy et al., 2019), and AIRL (Fu et al., 2017) allow the agent to interact with the environment independently and attempt to match the expert distribution over long horizons. The approach in (Reichlin et al., 2022) learns a recovery policy that takes the agent back to the set of demonstrated states but requires that the dynamics are known in the testing environment. Generative techniques such as (Ke et al., 2023) use a learned model to perform data augmentation by generating actions returning the agent to the expert distribution. While these approaches are highly effective at imitating the expert, their performance is fundamentally limited by the quality of the demonstrations.

Reinforcement Learning with Demonstrations: Most similar to our approach are techniques that use expert demonstrations to accelerate base off-policy RL methods. Approaches such as (Nair et al., 2018; Rajeswaran et al., 2017; Hester et al., 2018) add demonstrations to replay buffers of off-policy RL methods and use techniques such as prioritized experience replay and pre-training with behavior cloning to accelerate and stabilize learning. These methods are directly compatible with our core reward-shaping technique. Other contributions such as (Brys et al., 2015; Zhao et al., 2023; Pari et al., 2021) have performed reward-shaping to guide agents back to demonstrated states but rely on heuristic hand-designed measures of the distance between a given state and a nearby demonstration (such as simple L2 distances). Wilcox et al. (2022) use suboptimal but task-specific demonstrations. Our method bakes task-agnostic prior knowledge about the dynamics into the reward, which, as we discuss in Sec. 4.4, is essential to accelerate learning.

3. Problem Setting

3.1. Sparse Reward Reinforcement Learning Problem

Our ultimate objective is to efficiently solve long-horizon sparse reward reinforcement learning problems defined by an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ where $s \in \mathcal{S}$ is the state, $a \in \mathcal{A} \subset \mathbb{R}^a$ is the action, $s_{t+1} \sim P(\cdot|s_t, a_t)$ is the transition dynamics, $r(s, a)$ is the rewards, and $\gamma \in [0, 1)$ is

the discount factor. We will focus on the case where the agent aims to reach a goal set of states $\mathcal{G} \subset \mathcal{S}$ in as few steps as possible by optimizing over Π, the space of all policies π. For ease of analysis, we assume that the dynamics and policies are deterministic to simplify the exposition, as in prior works studying the structure of goal-conditioned problems (Ma et al., 2022). Namely, P can be described by a single-valued map f such that $s_{t+1} = f(s_t, a_t)$ and actions are generated according to $a_t = \pi(s_t)$. However, our method also applies to stochastic scenarios, as demonstrated in our experiments with stochastic policies. To incentivize the desired goal-reaching behavior, we formulate a policy optimization problem of the form:

$$\max_{\pi \in \Pi} \mathbb{E}_{s_0 \sim d_0} V^\pi(s_0), \quad V^\pi(s_0) := \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t), \tag{1}$$

where d_0 is a distribution over initial conditions. We use a sparse reward of $r(s_t, a_t) = 1$ if $s_{t+1} \in G$ to award successes and apply a reward of $r(s_t, a_t) = 0$ otherwise. Episodes terminate when the agent first reaches $\mathcal{G}$. The larger the magnitude of $V^\pi(s)$, the fewer steps it will take π to reach the goal set $\mathcal{G}$ when starting from s and using π. We define the optimal value function as:

$$V^*(s_0) := \max_{\pi \in \Pi} V^\pi(s_0). \tag{2}$$

The magnitude of $V^*(s)$ captures the optimal number of steps needed to reach the goal from s.

3.2. Goal-conditioned Policies

Consider a goal-conditioned controller of the form $a_t = \pi_g(s; s_d)$ where $s \in \mathcal{S}$ is the current state, and s_d is the desired state the agent attempts to reach. Let Π_g denote the space of all deterministic goal-conditioned policies. For each policy π_g, we define the value function:

$$V_g^{\pi_g}(s_0; s_d) := \sum_{t=0}^{\infty} \gamma^t r_g(s_t, a_t; s_d), \tag{3}$$

where we use the sparse rewards $r(s_t, a_t; s_d) = 1$ if $s_{t+1} = s_d$ to award the agent for successfully reaching the desired state, and apply a reward of $r(s_t, a_t; s_d) = 0$ otherwise. Episodes for the goal-conditioned agent terminate once the agent reaches the goal state and the magnitude of $V_g^{\pi_g}(s; s_d)$ captures the number of steps the given controller will take to reach the goal s_d from s.

4. Systematic Reward-Shaping from Demonstrations and Prior Experience

4.1. Learning Approximate Goal-conditioned Value Estimates from Prior Data

Our approach begins with a pre-training phase, wherein the agent leverages a broad prior data set of transitions $\mathcal{P} = \{(s_t^p, a_t^p, s_{t+1}^p)\}_{t=1}^{T_p}$, which can be generated by one or multiple prior MDPs where data have been collected. We assume that each of these MDPs has the same state and action space as the target MDP $\mathcal{M}$, but the dynamics of the prior MDPs can differ substantially from those encountered in $\mathcal{M}$. For example, in our experiments in Sec. 5, the prior data set comes from an environment without a slot. We use $\mathcal{P}$ to fit an approximation $\tilde{V}_g(s; s_d)$ to the goal-conditioned value function $V_g^{\pi_g}$ associated to some fixed policy π_g (but not necessarily the optimal goal-reaching policy). Our framework is agnostic to how $\tilde{V}_g$ is learned. For example, offline RL can be used to

learn both π_g and $\tilde{V}_g$ if $\mathcal{P}$ is a static data set (Mezghani et al., 2023). Or, as in our experiments, these quantities can be learned by generating $\mathcal{P}$ using online exploration and extensive play data.

Regardless of how $\tilde{V}_g$ is learned during the pre-training phase, it encodes valuable information about the dynamics that generated the prior data that we will use to accelerate learning for the downstream task defined by the target MDP $\mathcal{M}$. However, $\tilde{V}_g$ may be flawed due to 1) dynamics shift between the prior training data and the target environment and 2) limited data coverage in $\mathcal{P}$. In particular, 2) can make obtaining accurate value estimates for faraway goals extremely difficult. Thus, in practice, we aim only to learn value estimates for goals that require a relatively small number of steps to reach, reducing the horizon and sample complexity for the initial pre-training phase. For example, in our pushing experiments, we train controllers that can push a puck to goals up to 0.1 meter away during the pre-training phase and then use these primitives to track much longer expert demonstrations to distant goals. For concreteness, for each starting state $s \in \mathcal{S}$, let $\Delta(s) \subset \mathcal{S}$ denote the range of goals that we use to fit $\tilde{V}_g$ during the pre-training phase. For our analysis, it is convenient to assume that the sets of valid goals are of the form:

$$\Delta(s) := \{s_d \in S : \tilde{V}_g(s; s_d) \geq \beta\}, \tag{4}$$

where $\beta \in [0, 1)$ is uniform across all states.

We also highlight related works such as (Eysenbach et al., 2019), which explicitly attempt to estimate whether certain goals are out of distribution for goal-conditioned value estimators. While defining $\Delta(s)$ for the examples we consider is straightforward, these approaches are compatible with our method when it is more difficult to ascertain whether certain states are out of distribution.

4.2. Collecting Task-Specific Demonstrations

Our method assumes access to K expert demonstrations τ_d^j for the target MDP $\mathcal{M}$. When introducing our base method, we will initially assume that the learner has full access to the transitions performed by the expert. Thus, the expert data set will be of the form: $\mathcal{T} = \{\tau_d^1, \tau_d^2, \ldots, \tau_d^K\}, \tau_d^j = \{(s_t, a_t, s_{t+1}^j)\}_{t=0}^{H^j}$, where H^j is the number of transitions in the j-th demonstration. We assume that $s_{H_j+1}^j \in \mathcal{G}$ and $s_t^j \notin \mathcal{G}$ for each $t < H_j + 1$, so that the expert first reaches the goal set.

4.3. Sythesizing Dense Rewards

We now introduce our reward-shaping method, which uses the prior experience encapsulated in $\tilde{V}_g$ to synthesize dense dynamics-aware rewards for the target MDP $\mathcal{M}$. We begin by defining:

$$V_d^j(s_t^j) := \gamma^{H_j - t}, \tag{5}$$

This is simply the discounted sum of rewards incurred in the original MDP $\mathcal{M}$ by starting at state s_t^j and following the j-th expert demonstration to G. This value captures the number of steps needed to reach the goal by following the demonstration. Next, we define:[1]

$$\Phi^j(s) := \max_{s_t^j \in \Delta(s)} \left[V_d^j(s_t^j) + \tilde{V}_g(s; s_t^j) \right], \tag{6}$$

---

[1]We apply the convention that $\Phi^j(s) = 1$ if $s \in G$ and $\Phi^j(s) = 0$ if there does not exist a state s_t^j on the demonstration such that $s_t^j \in \Delta(s)$ (*i.e.*, if the j-th demonstration is too far away from s).

where the maximization is taken over the set of states that appear in τ^j and are in-distribution goals for $\tilde{V}_g$ for the starting state s. It balances choosing states on the demonstration, which will take a small number of steps to reach from s (as measured by $\tilde{V}_g(s; s_t^j)$) against picking states that are closer to the end of the demonstration (as measured by $V_d^j(s_t^j)$). We can view Φ^j as an estimate of the optimal value V^*, which is constructed by approximating how many steps it will take to reach τ^j and then follow the demonstration to G. We aggregate these estimates by defining:

$$\Phi(s) := \max_j \Phi^j(s), \tag{7}$$

which captures the optimal demonstration to reach and then follow from the state s. We then leverage the PBRS formalism (Ng et al., 1999) and define the dense rewards:

$$\bar{r}(s_t, a_t) = r(s_t, a_t) + \gamma\Phi(s_{t+1}) - \Phi(s_t), \tag{8}$$

and propose the reshaped policy optimization problem:

$$\max_{\pi \in \Pi} \mathbb{E}_{s_0 \sim d_0} \left[\bar{V}^\pi(s) \right], \quad \text{s.t.} \ \bar{V}^\pi(s) = \sum_{t=0}^{\infty} \gamma^t \bar{r}(s_t, a_t), \tag{9}$$

along the dynamics of the target environment. Φ is referred to as a *potential function* and can be viewed as an initial guess for the optimal value V^* injected into the reshaped reward structure to guide exploration. In particular, the new rewards incentivize choosing actions that increase the value of Φ at the following state, guiding learning agents to reach and follow the expert training distribution. This supervision provides a much stronger learning signal than the original spare rewards, which only provides feedback to the agent when it successfully reaches the goal set $\mathcal{G}$. It is extremely difficult, especially when the goal takes many steps to reach.

However, the PBRS formulation preserves the set of optimal policies between the original and reshaped MDPs (Ng et al., 1999). Thus, this formulation enables us to leverage prior experience and a small number of demonstrations to guide training without tying the end performance of the policies we learn to these potentially flawed data sources. Indeed, as our experiments demonstrate, our approach can still substantially accelerate learning even when highly sub-optimal demonstrations are used and there is a significant dynamics shift between the pre-training data set and target domain.

4.4. Guiding Exploration by Decreasing and Annealing Discount Factors

To see why the reshaped objective preserves the optimal policies defined by the original MDP, recall that for a given policy $\pi \in \Pi$, the associated Q-functions in the original and reshaped MDPs are:

$$Q^\pi(s, a) = \gamma V^\pi(s') + r(s, a), \quad \bar{Q}^\pi(s, a) = \gamma\bar{V}^\pi(s') + r(s, a) + \gamma\Phi(s') - \Phi(s). \tag{10}$$

By unrolling the expression for the reshaped value function (9), one may show that $\bar{V}^\pi(s) = V^\pi(s) - \Phi(s)$. Combining (9) and (10), $\bar{Q}^\pi(s, a) = \gamma V^\pi(s') + r(s, a) - \Phi(s) = Q^\pi(s, a) - \Phi(s)$. Because the $-\Phi(s)$ term does not depend on the action taken by the agent, $\bar{Q}^\pi$ preserves the ordering over actions defined by Q^π. When we take the given policy to be the optimal policy π^*, this argument shows that the reshaped MDP preserves the optimal ordering of actions in the original MDP. However, in practice, many reinforcement learning algorithms use an approximation to the Q-function learned from experience. When these data are limited, and the estimate for the Q-function is inaccurate, the shaped rewards can bias the agent towards actions that follow the potential, which enables the reshaped rewards to guide exploration early in the learning process.

We can more aggressively and directly incentivize the agent to follow the potential during exploration by altering the discount factor used during online training (Westenbroek et al., 2022; Cheng et al., 2021). In particular, suppose that we instead apply the long-term return $\bar{V}^\pi(s_0) = \sum_{t=0}^\infty \bar{\gamma}^t \bar{r}(s,a)$, where $\bar{\gamma} \in [0, \gamma]$ is a smaller discount factor (but we leave the original discount factor γ unchanged when it is used to construct Φ and $\bar{r}$). In general, reducing the discount factor for an MDP shortens the effective time horizon over which the agent searches while learning and can reduce the sample complexity and variance of many search methods (Westenbroek et al., 2022; Cheng et al., 2021). However, this regularization may also cause the agent to behave too myopically to achieve the desired long-horizon task. In our setting, the addition of the potential function in the reshaped rewards can guide the agent to act effectively over long horizons even when acting greedily. Consider the Q-function for the reshaped objective under the new discount factor $\bar{Q}^\pi(s,a) = \bar{\gamma}\bar{V}^\pi(s') + \gamma\Phi(s') - \Phi(s) + r(s,a)$. Because the original rewards are sparse, at most states, the only terms affected by the agent's actions are $\bar{\gamma}\bar{V}^\pi(s')$ and $\gamma\Phi(s')$. As $\bar{\gamma}$ gets smaller, the agent prefers to follow the potential over the value function (which must be bootstrapped from data in practice). In the extreme case where $\bar{\gamma} = 0$, the agent greedily follows the potential, relying on the prior information baked into Φ to guide exploration. It may be advantageous early in training when data coverage in the new domain is limited (and bootstrapped estimates for $\bar{V}^\pi(s')$ are poor) but will tie the final performance of the policy to the quality of the data used to synthesize Φ.

One compromise we leverage in our experiments is to train the agent with an increasing sequence of discount factors $\{\bar{\gamma}_k\}_{k=1}^M$ where $\bar{\gamma}_1 = 0$ and $\bar{\gamma}_M = \gamma$. By annealing the discount factor, the agent relies more heavily on the potential for guidance when encountering a new environment but still recovers the optimal policy for the environment at convergence. The following result provides sufficient conditions to ensure the online learner can discover trajectories that reach the goal by greedily following the potential. The supplementary material provides the proof in the Appendix C.

Proposition 1 *Assume each state $s \in \mathcal{S}$, in-distribution goal $s_d \in \Delta(s)$, and value estimator $\tilde{V}_g$ satisfies the following conditions for some $\epsilon > 0$:*
1. $\tilde{V}_g(s; s_d) \in [0, 1]$ and $\tilde{V}_g(s; s_d) = 1$ iff $s = s_d$; and,
2. $\max_a \tilde{V}_g(f(s,a); s_d) > (1 + \epsilon)\tilde{V}_g(s; s_d)$.
Further assume that the expert training set $\mathcal{T}$ satisfies the following condition:
3. for each state $s \in \mathcal{S}$ there exists some state s_t^j on some demonstration τ^j such that $s_t^j \in \Delta(s)$.
Finally, assume that $\bar{\gamma} = 0$ is used to optimize the reshaped rewards, and let $\bar{\pi}^*$ denote the optimal policy. Then, every trajectory generated by $\bar{\pi}^*$ will reach the goal set G in a finite number of steps.

The first assumption in the proposition ensures that the value estimator $\tilde{V}_g$ has a well-defined maximum at the goal state. The second condition says that we can reach each in-distribution goal for the state s by greedily following $\tilde{V}_g$ at each time step. Note that this does not imply that $\tilde{V}_g$ accurately predicts the number of steps needed to reach s_d, but rather that it captures the rough structure of the dynamics in the target MDP. Finally, the third condition requires that at every state s, there exists an in-distribution goal for $\tilde{V}_g$, which lies on an expert demonstration. In particular, this condition captures the fact that there is a "tube" around each expert demonstration on which our dense rewards provide useful information (with the radius of the tube corresponding to the range of in-distribution goals for $\tilde{V}_g$). The third condition implies that these tubes cover the entire state space, ensuring sufficient data coverage from the demonstrations and the prior data set to rely on greedily following the potential at every point in the state space. However, while these conditions are strong, we em-

phasize that we can always ensure that our approach recovers the optimal policy at convergence by annealing the discount factor back to the original value during training.

4.5. Learning From (Partial) Observations

While our general reward-shaping technique can be incorporated with many paradigms, in this paper, we apply this method in the context of online off-policy RL. Like many other approaches, if the learner can directly observe the actions and states that occur in the expert demonstrations, then the demonstrations can be added to the replay buffer of the off-policy RL algorithm (Nair et al., 2018; Rajeswaran et al., 2017). Importance sampling can replay the demonstrations more frequently, which can help propagate reward information over long horizons.

However, note that our reward-shaping approach, as introduced in Sec. 4.3, only requires access to the states generated by the expert and thus can be applied when only state trajectories are available. Indeed, in our experimental results in Sec. 5, we observe that applying our reward shaping approach without access to expert actions can lead to faster learning when compared to an agent that has access to the expert actions but does not benefit from our informative reward structure.

Our approach extends to scenarios where the expert only has access to partial state information or must infer certain states from high-dimensional observations. Rather than specifying goals with full desired states, in many scenarios, only a subset of states is crucial for defining goals. For example, in many manipulation tasks, success only depends on the final state of the object but not the manipulator. Moreover, in many practical scenarios, such as when a robot observes a human manipulating an object, only the state of the object is observable. Still, there are no concrete actions or states of the robot that can be directly associated with the demonstration.

We can formalize this scenario by breaking down the state into $s = (s_1, s_2)$, where s_1 are the goals-relevant states, and we define goal sets of the form $g(s_d^1) = \{s = (s_1, s_2) : s_1 = s_d^1\}$ when learning a goal-conditioned value function $\tilde{V}_g(s; g(s_d^1))$. That is when charged with reaching the goal $g(s_d^1)$, the agent receives the reward $r(s, a) = 1$ if $s' \in g(s_d^1)$ and the reward $r(s, a) = 0$ otherwise. In this case, the demonstration $\tau^j = (s_0^{1,j}, \ldots, s_t^{1,j})$ above consists of only the goal-relevant states, and we construct the potential via:

$$V_d(s_t^{1,j}) = \gamma^{H^j - t}, \quad \Phi^j(s) = \max_{s_t^{1,j} \in \tau^j} \left[V_d(s_t^{1,j}) + \tilde{V}_g(s; g(s_t^{1,j})) \right]. \tag{11}$$

The overall potential function can again be defined by $\Phi(s) = \max_j \Phi^j(s)$ to generate dense rewards with PBRS as before. In our experiments, this reward structure drives the agent to push the object to positions demonstrated by the expert. While the demonstration may not contain a sequence of states for the arm that is needed to realize the desired motion for the object, this information is baked into $\tilde{V}_g$, which (approximately) encodes sequences of states for both the object and the arm that are needed to move the object to nearby goal positions. This extension also applies to situations where a subset of the states can only be inferred through high-dimensional observations o_t. Indeed, the goal sets can be defined in the observation space in this case.

5. Experiments

In this section, we investigate 1) the speed our method learns tasks from few demonstrations and 2) its adaptability to sub-optimal demonstrations and high-dimensional observations. We conduct

empirical studies in the following tasks: pushing a puck to a U-shaped slot, utilizing both 1) perfect state observations and 2) imperfect, high-dimensional point cloud observations to verify the effectiveness of our proposed reward-shaping approach with binary rewards. We implement our approach on top of the off-policy Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018). See Appendix B for experimental details.

Baselines: We combine our reward-shaping approach with an importance sampling buffer (Sec. 4.5) and the annealing discount factor technique (Sec. 4.4). The importance sampling buffer has a probability of 10% to include expert demonstrations in each batch, and the annealing trick linearly increases the discount factor from 0 to γ during training. The label *potential* refers to an agent trained with our reshaped rewards. The label *buffer* refers to an agent having direct access to the expert states and actions and has included them in the importance sampling buffer as *buffer*, and *annealing* indicates the annealing trick in the following plots and tables. For example, Buffer + Potential + Annealing refers to an agent trained with all three modes active.

We compared our methods with 6 baselines: the Buffer Only mode described above, Behavior Cloning (BC) (Torabi et al., 2018), GAIL (Ho and Ermon, 2016), PWIL (Dadashi et al., 2021), DRIL (Brantley et al., 2020), and AdRIL (Swamy et al., 2021). The Buffer Only mode does not incorporate reward-shaping and helps us to ablate how our method can improve base methods, which incorporate demonstrations into the replay buffer (Nair et al., 2018; Rajeswaran et al., 2017). The other baselines are standard imitation learning methods that learn from

(*a*) Push slot   (*b*) Potential heatmap

Figure 2: (a) Push slot environment in Robosuite. (b) Potential heatmap for the push slot with two reset grids.

demonstrations. We exclude comparisons of agents that solve the original sparse reward problems without reward-shaping or demonstrations, as their performance is significantly worse than all other methods tested. We implemented the baselines from (Arulkumaran and Ogawa Lillrank, 2023).

Setup: We created a puck-pushing environment in the Robosuite (Zhu et al., 2020) simulator, where a Franka Panda robot arm aims to push a puck into a U-shaped slot that is blocked by three walls, shown in Fig. 2. The puck is initialized on one or multiple reset grids (gray), and an RL agent controls the 2-dimensional displacement of the end effector. Success occurs when the puck enters the slot. The agent observes the puck's displacement and speed relative to the end effector and the goal. Demonstrations are collected with a sub-optimal, hand-coded expert policy that pushes the puck to predetermined anchor points around the slot to complete the task. However, as mentioned in Sec. 4.5, for this task, the goal sets depend only on the position of the puck but not the robot arm. Thus, we simulated situations where a human demonstrator moves the puck with their hand, and only the position of the puck is recorded in the demonstrations. We tested 3 variants of the task: 1) the agent with perfect state estimates for the puck, with either a) one reset grid or b) two reset grids, and 2) the agent infers the puck's position from high-dimensional point-cloud observations. The two-reset-grid variant is depicted in Fig. 2, and the one-reset-grid in Fig. 4 in the appendix.
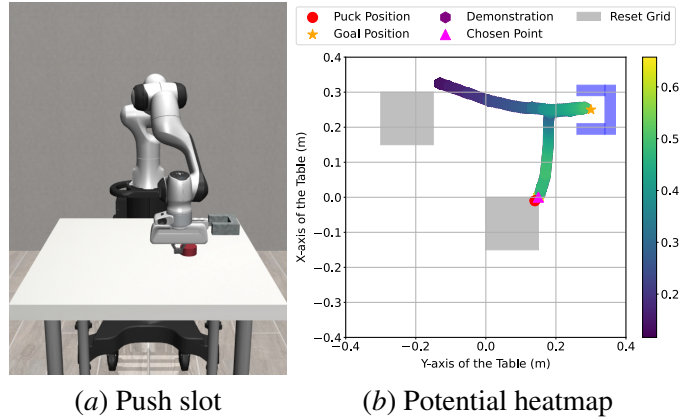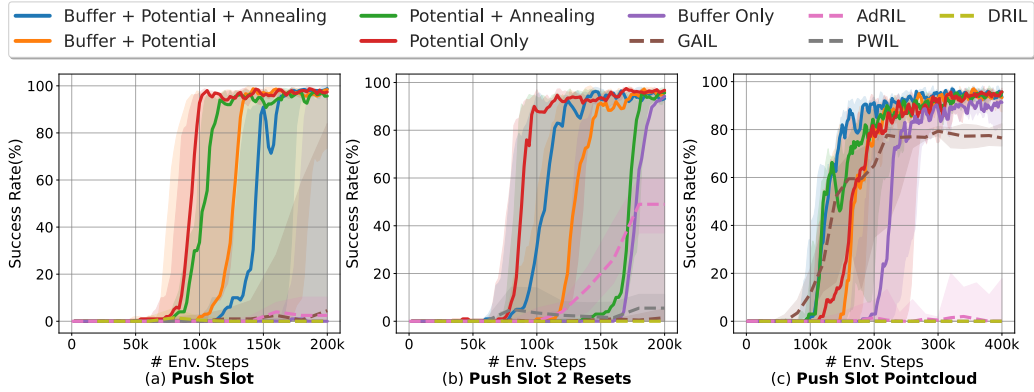
Figure 3: Training progression in push slot tasks. We present the median (bold) and quartiles (shaded area) of success rates. Our approach leads to faster learning and more stable performance across tasks.

The pointcloud environment has only one reset grid. We provided **one demonstration from each active reset grid** for all experiments, as shown in Fig. 2.

Results: Fig. 3 shows the progression of the success rates. We showed the detailed statistics in the appendix (Tab. 1). Learning is accelerated substantially for the agents which have access to our shaped rewards. In contrast, previous online learning methods, GAIL, PWIL, AdRIL, and DRIL, cannot master the task. This is because the learned value estimates $\tilde{V}_g$ encode these sorts of short-horizon pushing primitives *a priori*, enabling the agent to quickly adapt to the new long-horizon task. Note that we do not include BC in Fig. 3 since it is not an online learning method, and thus no learning progress can be visualized. However, Tab. 1 shows that while BC converges, it fails to master any pushing task even with 100 demonstrations. Our approaches effectively learn the tasks even with high-dimensional pointcloud observations, while other baselines fail. Also, when successful, GAIL learns sub-optimal behaviors, requiring around 90 steps to complete the tasks. In contrast, our approaches complete the tasks in approximately 30 steps, showcasing their ability to learn optimal behaviors from sub-optimal demonstrations with 100 steps.

6. Discussion

This paper presents a general reward-shaping approach that ties together prior task-agnostic data and a small number of expert demonstrations to synthesize informative rewards for new tasks. We demonstrated that this core technique can substantially accelerate learning for new downstream tasks. However, the scope of the current study is limited in a few key ways, which leaves many avenues for future work. First, while we have demonstrated our approach with high-dimensional pointcloud observations, it will be important to demonstrate that it can extend to other rich sensory modalities, such as RGB images. Fortunately, there has been extensive work on learning and effectively leveraging latent representations and goal-conditioned value functions in those representations (Ma et al., 2022; Eysenbach et al., 2019). Next, we are not fully leveraging the prior data set in the present framework. Several works have investigated using goal-conditioned controllers to accelerate exploration (Nasiriany et al., 2019) and mixing in samples from prior data sets when training on new tasks (Kumar et al., 2022), rather than training from scratch on each new task. Fortunately, our reward-shaping framework is agnostic to these techniques, and the pathway to integrating our approach with these contributions leaves a clear avenue for future work.

Acknowledgments

References

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

Kai Arulkumaran and Dan Ogawa Lillrank. A pragmatic look at deep imitation learning. In *Asian Conference on Machine Learning*, 2023.

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

Adam Block, Ali Jadbabaie, Daniel Pfrommer, Max Simchowitz, and Russ Tedrake. Provable guarantees for generative behavior cloning: Bridging low-level stability and high-level behavior. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=PhFVF0gwid.

Kiante Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized imitation learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgbYyHtwB.

Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E Taylor, and Ann Nowé. Reinforcement learning from demonstration through shaping. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning*, pages 1430–1440. PMLR, 2021.

Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. Heuristic-guided reinforcement learning. *Advances in Neural Information Processing Systems*, 34:13550–13563, 2021.

Nuttapong Chentanez, Andrew Barto, and Satinder Singh. Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17, 2004.

Robert Dadashi, Leonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal wasserstein imitation learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=TtYSU29zgR.

Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

Liyiming Ke, Yunchu Zhang, Abhay Deshpande, Siddhartha Srinivasa, and Abhishek Gupta. Ccil: Continuity-based data augmentation for corrective imitation learning. *arXiv preprint arXiv:2310.12972*, 2023.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Aviral Kumar, Anikait Singh, Frederik Ebert, Mitsuhiko Nakamoto, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. *arXiv preprint arXiv:2210.05178*, 2022.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pages 143–156. PMLR, 2017.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

Lina Mezghani, Sainbayar Sukhbaatar, Piotr Bojanowski, Alessandro Lazaric, and Karteek Alahari. Learning goal-conditioned policies offline with self-supervised reward shaping. In *Conference on Robot Learning*, pages 1401–1410. PMLR, 2023.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6292–6299. IEEE, 2018.

Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. *Advances in Neural Information Processing Systems*, 32, 2019.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, volume 99, pages 278–287, 1999.

Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: Imitation learning via reinforcement learning with sparse rewards. *arXiv preprint arXiv:1905.11108*, 2019.

Alfredo Reichlin, Giovanni Luca Marchetti, Hang Yin, Ali Ghadirzadeh, and Danica Kragic. Back to the manifold: Recovering from out-of-distribution states. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8660–8666. IEEE, 2022.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Gokul Swamy, Sanjiban Choudhury, J Andrew Bagnell, and Steven Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pages 10022–10032. PMLR, 2021.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018. ISBN 9780999241127.

Tyler Westenbroek, Fernando Castaneda, Ayush Agrawal, Shankar Sastry, and Koushil Sreenath. Lyapunov design for robust and efficient robotic reinforcement learning. *arXiv preprint arXiv:2208.06721*, 2022.

Albert Wilcox, Ashwin Balakrishna, Jules Dedieu, Wyame Benslimane, Daniel Brown, and Ken Goldberg. Monte carlo augmented actor-critic for sparse reward deep reinforcement learning from suboptimal demonstrations. *Advances in neural information processing systems*, 35:2254–2267, 2022.

Ruihan Zhao, Ufuk Topcu, Sandeep Chinchali, and Mariano Phielipp. Learning sparse control tasks from pixels by latent nearest-neighbor-guided explorations. *arXiv preprint arXiv:2302.14242*, 2023.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiri-any, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.