D-Shape: Demonstration-Shaped Reinforcement Learning via Goal Conditioning

Caroline Wang
The University of Texas at Austin
Austin, Texas, United States
caroline.l.wang@utexas.edu

Garrett Warnell
Army Research Laboratory and
The University of Texas at Austin
Austin, Texas, United States
garrett.a.warnell.civ@army.mil

Peter Stone
The University of Texas at Austin and
Sony AI
Austin, Texas, United States
pstone@cs.utexas.edu

ABSTRACT

While combining imitation learning (IL) and reinforcement learning (RL) is a promising way to address poor sample efficiency in autonomous behavior acquisition, methods that do so typically assume that the requisite behavior demonstrations are provided by an expert that behaves optimally with respect to a task reward. If, however, suboptimal demonstrations are provided, a fundamental challenge appears in that the demonstration-matching objective of IL conflicts with the return-maximization objective of RL. This paper introduces D-Shape, a new method for combining IL and RL that uses ideas from reward shaping and goal-conditioned RL to resolve the above conflict. D-Shape allows learning from suboptimal demonstrations while retaining the ability to find the optimal policy with respect to the task reward. We experimentally validate D-Shape in sparse-reward gridworld domains, showing that it both improves over RL in terms of sample efficiency and converges consistently to the optimal policy in the presence of suboptimal demonstrations.

KEYWORDS

reinforcement learning; goal-conditioned reinforcement learning; imitation from observation; suboptimal demonstrations

ACM Reference Format:
Caroline Wang, Garrett Warnell, and Peter Stone. 2023. D-Shape: Demonstration-Shaped Reinforcement Learning via Goal Conditioning. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 13 pages.

1 INTRODUCTION AND BACKGROUND

A longstanding goal of artificial intelligence is enabling machines to learn new behaviors. Towards this goal, the research community has proposed both imitation learning (IL) and reinforcement learning (RL). In IL, the agent is given access to a set of state-action expert demonstrations and its goal is either to mimic the expert's behavior, or infer the expert's reward function and maximize the inferred reward. In RL, the agent is provided with a reward signal, and its goal is to maximize the long-term discounted reward. While RL algorithms can potentially learn optimal behavior with respect to the provided reward signal, in practice, they often suffer from high sample complexity in large state-action spaces, or spaces with sparse reward signals. On the other hand, IL methods are typically more sample-efficient than RL methods, but require expert demonstration data.

It seems natural to consider using techniques from imitation learning and demonstration data to speed up reinforcement learning. However, many IL algorithms implicitly perform divergence minimization with respect to the provided demonstrations, with no notion of an extrinsic task reward [11]. When we have access to both demonstration data *and* an extrinsic task reward, we have the opportunity to combine IL and RL techniques, but must carefully consider whether the demonstrated behavior conflicts with the extrinsic task reward — especially when demonstrated behavior is suboptimal. Moreover, standard IL algorithms are only valid in situations when demonstrations contain both state and action information, which is not always the case.

The community has recently made progress in the area of imitation from observation (IfO), which extends IL approaches to situations where demonstrator action information is unavailable, difficult to induce, or not appropriate for the task at hand. This last situation may occur if the demonstrator's transition dynamics differ from the learner's—for instance, when the expert is a human and the agent is a robot [20, 35]. While there has been some work on performing IL or IfO with demonstrations that are suboptimal with respect to an implicit task that the demonstrator seeks to accomplish [4, 5, 7, 38], to date, relatively little work has considered the problem of combining IfO and RL, where the learner's true task is explicitly specified by a reward function [32].

This paper introduces the D-Shape algorithm, which combines IfO and RL in situations with suboptimal demonstrations. D-Shape requires only a single, suboptimal, state-only expert demonstration, and treats demonstration states as goals. To ensure that the optimal policy with respect to the task reward is not altered, D-Shape uses potential-based reward shaping to define a goal-reaching reward. We show theoretically that D-Shape preserves optimal policies, and show empirically that D-Shape improves sample efficiency over related approaches with both optimal and suboptimal demonstrations.

2 PRELIMINARIES

This section introduces our notation, and technical concepts that are key to this work: reinforcement learning with state-only demonstrations, goal-conditioned reinforcement learning, and potential-based reward shaping.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom. © 2023 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

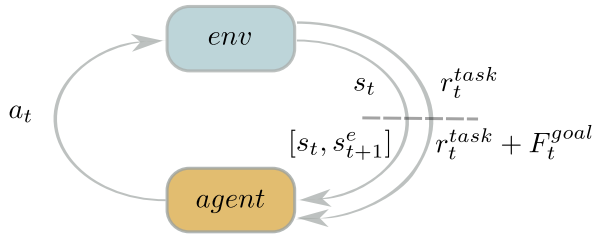

Figure 1: D-Shape's interaction with the environment. The state s_t and the task reward r_t^{task} come from the environment. s_t is concatenated with the demonstration state s_{t+1}^e as the goal, and r^{task} is augmented with the potential-based goal reaching function, F_t^{goal}

.

2.1 Reinforcement Learning with State-Only Demonstrations

Let $M = (S, A, P, r^{task}, \gamma)$ be a finite-horizon Markov Decision Process (MDP) with horizon H, where S and A are the state and action spaces, $P(s' \mid s, a)$ is the transition dynamics of the environment, $r^{task}(s, a, s')$ is a deterministic extrinsic task reward, and $\gamma \in (0, 1)$ is the discount factor. The objective of reinforcement learning is to discover a policy $\pi(\cdot \mid s)$ that maximizes the expected reward induced by π, $\mathbb{E}_\pi[\sum_{t=0}^{H-1} \gamma^t r^{task}(s_t, a_t, s_{t+1})]$. In this work, we seek to maximize the same objective, but to do so more efficiently by incorporating additional information in the form of a single, state-only demonstration $D^e = \{s_t^e\}_{t=1}^H$, that may be suboptimal with respect to the task reward. The extent to which the demonstration can improve learning efficiency may depend on its degree of suboptimality. However, incorporating the demonstration into the reinforcement learning procedure should not alter the optimal policy with respect to r^{task}, no matter how suboptimal the demonstration is. Prior literature has referred to this desideratum as *policy invariance* [22].

2.2 Goal-Conditioned Reinforcement Learning

Goal-conditioned reinforcement learning (GCRL) further considers a set of goals G [18, 28]. While standard RL aims to find policies that can be used to execute a single task, the objective of GCRL is to learn a goal-conditioned policy $\pi(\cdot \mid [s, g])$, where the task is to reach any goal $g \in G$. Typically, G is a predefined set of desirable *states*, and the reward function depends on the goal. A common choice of reward function is the sparse indicator function for when a goal has been reached, $r_t^g = \mathbb{1}_{s_t = g}$.

Since it is challenging for RL algorithms to learn under sparse rewards, Andrychowicz et al. [2] introduced hindsight experience replay (HER). In the setting considered by Andrychowicz et al. [2], the goal is set at the beginning of an episode and remains fixed throughout. HER relies on the insight that even if a trajectory fails to reach the given goal g, transitions in the trajectory are successful examples of reaching *future* states in the trajectory. More formally, given a transition with goal g, $([s_t, g], a_t, r_t^g, [s_{t+1}, g])$, HER samples a set of goals from future states in the episode, $\mathcal{G}$. For all goals $g' \in \mathcal{G}$, HER relabels the original transition to $([s_t, g'], a_t, r_t^{g'}, [s_{t+1}, g'])$ and stores the relabelled transition in a replay buffer. An off-policy RL algorithm is used to learn from the replay buffer.

In this work, we use demonstration states as goals, allowing the goal to change dynamically throughout the episode, and employ the relabelling technique from HER.

2.3 Potential-Based Reward Shaping

Define a *potential function* $\phi : S \mapsto \mathbb{R}$. Let $F(s, s') \coloneqq \gamma\phi(s') - \phi(s)$; F is called a *potential-based shaping function*. Consider the MDP $M' = (S, A, P, R' \coloneqq r^{task} + F, \gamma)$, where r^{task} is an extrinsic task reward. We say R' is a *potential-based reward function*. Ng et al. [22] showed that F being a potential-based shaping function is both a necessary and sufficient condition to guarantee that (near) optimal policies learned in M' are also (near) optimal in M — that is, policy invariance holds. This work leverages potential-based reward functions as goal-reaching rewards, to bias the learned policy towards the demonstration trajectory.

3 METHOD

We now introduce D-Shape, our approach to improving sample efficiency while leaving the optimal policy according to the task reward unchanged. The training procedure is summarized in Algorithm 1.

D-Shape requires only a single, possibly suboptimal, state-only demonstration. We are inspired by model-based IfO methods that rely on inverse dynamics models (IDMs) — models that, given a current state and a target state, return the action that induces the transition. We observe that an IDM can be viewed as a single-step goal-reaching policy. Although D-Shape does not assume access to an IDM, we hypothesize that providing expert demonstration states as goals to the reinforcement learner might be a useful inductive bias. HER is used to form an implicit curriculum to learn to reach demonstration states. As such, there are three components of D-Shape: state augmentation, reward shaping, and a goal-relabelling procedure. Figure 1 depicts the state augmentation and reward shaping process of a D-Shape learner as it interacts with the environment. Each component is described below.

3.0.1 State augmentation. During training, the policy gathers data with demonstration states as behavior goals: $a_t \sim \pi_\theta(a_t \mid [s_t, s_{t+1}^e])$. The agent observes the next state s_{t+1} and the task reward r_t^{task}. The next state s_{t+1} is augmented with the demonstration state s_{t+2}^e as the goal. Note that our method employs dynamic goals, as the goal changes from time step t to time step $t + 1$.

3.0.2 Reward shaping. The task reward r_t^{task} is summed with a potential-based shaping function to form a potential-based goal-reaching reward. Define the potential function $\phi([s_t, g_t]) \coloneqq -d(s_t, g_t)$, where d is a distance function, s_t is the state observed by the agent at time t, and g_t is the provided goal. Because the goal is defined as part of the state, $\phi(\cdot)$ only depends on the state, as required by the formulation of potential-based reward shaping considered by Ng et al. [22]. The potential-based shaping function is then

$$F^{goal}([s_t, g_t], [s_{t+1}, g_{t+1}]) \coloneqq \gamma\phi([s_{t+1}, g_{t+1}]) - \phi([s_t, g_t]). \quad (1)$$

Algorithm 1 D-Shape

Require: Single, state-only demonstration $D^e := \{s_t^e\}_{t=1}^H$
Initialize θ at random
while θ is not converged **do**
  for $t = 0 : H-1$ **do**
    Execute $a_t \sim \pi_\theta(\cdot \mid [s_t, s_{t+1}^e])$, observe r_t^{task}, s_{t+1}
    Compute $r_t^{goal} = r_t^{task} + F_t^{goal}$ using Equation (2)
    Store transition
    $([s_t, s_{t+1}^e], a_t, r_t^{goal}, [s_{t+1}, s_{t+2}^e])$ in buffer
  end for
  for $t = 1 : H-1$ **do**
    Sample set of consecutive goal states $\mathcal{G}$ uniformly from episode
    for $(g, g') \in \mathcal{G}$ **do**
      Recompute F_t^{goal} component of r_t^{goal} using (g, g')
      Relabel transition to
      $([s_t, g], a_t, r_t^{goal'}, [s_{t+1}, g'])$
      Store relabelled transition in replay buffer
    end for
  end for
  Update θ using off-policy RL algorithm
end while
return θ^*

The potential-based reward function is

$$r_t^{goal} := r_t^{task} + F_t^{goal}. \tag{2}$$

Note that r_t^{goal} is a goal-reaching reward that can be recomputed with new goals, suitable for goal relabelling. The above procedure results in "original" transitions of the form $([s_t, s_{t+1}^e], a_t, r_t^{goal}, [s_{t+1}, s_{t+2}^e])$, which are stored in a replay buffer.

Goal Relabelling. To encourage the policy to reach provided goals, we perform the following goal-relabelling procedure on original transitions using previously achieved states as goals. We adopt a similar technique to HER, with a slight modification to the goal sampling strategy. D-Shape's goal sampling strategy consists of sampling consecutive pairs of achieved states (g, g') from the current episode. The original transitions are then relabelled to $([s_t, g], a_t, r^{goal'}, [s_{t+1}, g'])$, where the reward is recomputed as $r^{goal'} = r_t^{task} + F_t^{goal}([s_t, g], a_t, [s_{t+1}, g'])$. As the goals are imaginary, even if the goal states change, the task reward r_t^{task} remains unchanged.

The policy π_θ is updated by applying an off-policy RL algorithm to the original data combined with the relabelled data. In our experiments, we use Q-learning [36]. At inference time, the policy once more acts with demonstration states as goals, i.e., $a_t \sim \pi_\theta(a_t \mid [s_t, s_{t+1}^e])$.

4 CONSISTENCY WITH THE TASK REWARD

In this section, we prove the claim that D-Shape preserves the optimal policy with respect to the task reward by showing that we can obtain optimal policies on M from an optimal policy learned by D-Shape. The analysis treats D-Shape as the composition of goal relabelling and potential-based reward shaping and formalizes this composition as the MDP transformation, $M \to M^\dagger \to M^\ddagger$ (Figure 2).

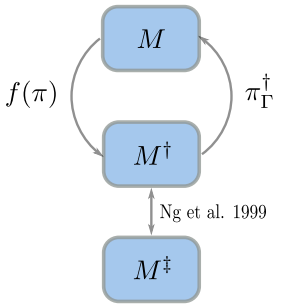

Figure 2: Our theoretical analysis considers D-Shape as the composition of goal relabelling ($M \to M^\dagger$) and potential-based reward shaping ($M^\dagger \to M^\ddagger$). D-Shape learns in $M^\dagger$. That optimal policies are preserved from $M^\dagger$ to $M^\ddagger$ follows directly from the policy invariance results of Ng et al. [22]. The theory provided considers a policy π in M in $M^\dagger$ via the natural extension, $f(\pi)$, and considers a policy $\pi^\dagger$ in $M^\dagger$ as operating in M via $\Gamma(s)$.

Let $M = (S, A, P, r^{task}, \gamma)$ be an MDP with horizon H, where S and A are the state and action spaces, $P(s' \mid s, a)$ is the transition dynamics of the environment, $r^{task}(s, a, s')$ is the deterministic extrinsic task reward, and $\gamma \in (0, 1)$ is the discount factor. Modify M to $M^\dagger$ as follows: $M^\dagger = (S \times G, A, P^\dagger, \gamma, r^\dagger)$, where G is a discrete set of goals, and

$$P^\dagger([s', g'] \mid [s, g], a) = P(s' \mid s, a)P(g' \mid [s, g], a),$$
$$r^\dagger([s, g], a, [s', g']) = r^{task}(s, a, s').$$

We make two independence assumptions in our definition of $P^\dagger$. First, that the random variable $(s' \mid [s, g], a)$ is independent of $(g' \mid [s, g], a)$, allowing us to factorize $P^\dagger([s', g'] \mid [s, g], a) = P(s' \mid [s, g], a)P(g' \mid [s, g], a)$. Second, that $(s' \mid s, g, a)$ is independent of $(g \mid s, a)$, allowing us to rewrite $P(s' \mid [s, g], a) = P(s' \mid s, a)$. In the context of D-Shape, the above assumptions simply mean that a goal in the replay buffer must be independent of all states, goals, and actions other than the previous state, goal, and action. We also require that $r^\dagger$ is independent of goals. We justify that our implementation of D-Shape approximately satisfies these assumptions in the Supplemental Material.

Now define $M^\ddagger = (S \times G, A, P^\dagger, \gamma, r^\dagger + F^{goal})$, where F^{goal} is defined as in Equation 1. $M^\ddagger$ is identical to $M^\dagger$, except for the addition of the potential-based shaping function, F^{goal}.

D-Shape learns a goal-conditioned policy $\pi^\ddagger(\cdot \mid [s, g])$ in $M^\ddagger$. To perform inference with the goal-conditioned policy in M, we must specify a state-goal mapping $\Gamma : S \mapsto G$. Then $\pi^\ddagger(\cdot \mid [s, \Gamma(s)])$ can be executed in M. Suppose that $\pi^{\ddagger *}(\cdot \mid [s, g])$ is an optimal policy in $M^\ddagger$. Is there a Γ such that $\pi^{\ddagger *}(\cdot \mid [s, \Gamma(s)])$ is optimal in M? We show next that the answer is positive, and that an arbitrary Γ suffices.

That $\pi^{\ddagger *}$ is optimal in $M^\dagger$ follows from the policy invariance results proven by Ng et al. [22]. By their result, as long as the

potential function F^{goal} depends only on the states $([s,g],[s',g'])$ and has the form $F([s,g],[s',g']) = \gamma\phi([s',g']) - \phi([s,g])$, the optimal policy will not be altered by learning under $r^{task} + F^{goal}$.

Thus, the main result reduces to showing that one can obtain an optimal policy in M from an optimal policy in $M^\dagger$(Theorem 1). The proof of Theorem 1 relies on some supporting results, which are stated below and proven in the Supplemental Material. [1]

Let us first define a way to map policies in M to $M^\dagger$. Define the policy transformation map, $f(\pi)([s,g]) = \pi(\cdot \mid s)$. Under the policy transformation map, policies expressible in M can be naturally expressed in $M^\dagger$.

Proposition 1. Let $\pi(\cdot \mid s)$ be a policy defined in M, and let $f(\pi)(\cdot \mid [s,g])$ be the extension of $\pi(\cdot \mid s)$ to $M^\dagger$. Then $V_\pi(s_t) = V_{f(\pi)}([s_t,g_t])\ \forall s_t \in S, g_t \in G$, and $t \in \{1,\dots H\}$.

Corollary 1.1. $Q_{f(\pi)}([s_t,g_t],a_t) = Q_\pi(s_t,a_t)$ for all $s_t \in S, g_t \in G, a_t \in A, t \in \{1,\dots H\}$.

Proposition 2. Let $\pi^*(\cdot \mid s)$ be an optimal policy in M, and let $f(\pi^*)(\cdot \mid [s,g])$ be its extension to $M^\dagger$. Then $f(\pi^*)$ is an optimal policy in $M^\dagger$.

Proposition 1 states that the value of π in M is equal to the value of $f(\pi)$ in $M^\dagger$ for all $s \in S, g \in G$, and follows from the observation that the reward function does not depend on goals. Corollary 2 makes the same claim, but for state-action value functions. Proposition 2 states that if a policy π^* is optimal in M, then $f(\pi^*)$ must also be optimal in $M^\dagger$, and is proven by showing that the policy $f(\pi^*)$ cannot be improved.

Theorem 1. *Let $\pi^{\dagger *}(\cdot \mid [s,g])$ be an optimal policy in $M^\dagger$. Define an arbitrary state-goal mapping $\Gamma : S \mapsto G_0 \subseteq G$, where G_0 is the set of goals in $M^\dagger$ that has positive probability of being reached by any policy. Then $\pi^{\dagger *}_\Gamma(\cdot \mid s) := \pi^{\dagger *}(\cdot \mid [s,\Gamma(s)])$ is an optimal policy in M.*

Proof. We prove the statement by considering the optimal state-action value functions in M and $M^\dagger$. Denote the set of optimal actions at state-goal $[s,g] \in M^\dagger$ by $A_{s,g} = \arg\max_a Q^{\dagger *}([s,g],a)$.

Let $\pi^*(\cdot \mid s)$ be an optimal policy in M, and let $f(\pi^*)$ denote the policy induced by π^* in $M^\dagger$ as in Proposition 2. By Proposition 2, $f(\pi^*)$ is an optimal policy in $M^\dagger$. Since the optimal value function in $M^\dagger$ is unique, $Q_{f(\pi^*)}([s,g],a) = Q^{\dagger *}([s,g],a)$ for all $s \in S, g \in G$, and $a \in A$. By Corollary 1.1, $Q_{f(\pi^*)}([s,g],a) = Q_{\pi^*}(s,a)$. Thus,

$$\begin{aligned} A_{s,g} &:= \arg\max_a Q^{\dagger *}([s,g],a) \\ &= \arg\max_a Q_{f(\pi^*)}([s,g],a) \\ &= \arg\max_a Q_{\pi^*}(s,a). \end{aligned}$$

The computation shows that the set of optimal actions in $M^\dagger$ at some $[s,g] \in S \times G$ is the same as the set of optimal actions in M at s — that is, the set of optimal actions is independent of the goal g. Since the set of actions with positive probability under $\pi^{\dagger *}(\cdot \mid [s,g])$ is a subset of $A_{s,g}$ for $g \in G_0$, we have $\pi^{\dagger *}_\Gamma(\cdot \mid s) := \pi^{\dagger *}(\cdot \mid [s,\Gamma(s)]) \subset A_{s,\Gamma(s)} = \arg\max_a Q_{\pi^*}(s,a)$ for all $s \in S$. Thus, $\pi^{\dagger *}_\Gamma$ is an optimal policy in M. □

[1] Supplemental Material is included in the arXiv version, https://arxiv.org/abs/2210.14428.

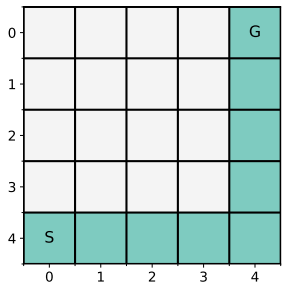

Figure 3: A 5×5 example of the gridworld task and the states traversed by the optimal demonstration (shown in blue). The start position is marked by "S", and the goal position by "G". The optimal demonstration starts at the "S" and goes to "G" along the blue states in the fewest possible number of steps.

The condition that $\Gamma(S) \subseteq G_0 \subseteq G$ avoids the technical issue that at a state $[s,g]$ which has zero probability of being reached, any action is trivially optimal. Thus an optimal policy in $M^\dagger$ at such a state $[s,g]$ might specify an action that is not optimal at s in M. The intuition behind the result is that, although the state space and transition probabilities for $M^\dagger$ incorporate a goal distribution, the reward function $r^\dagger$ is goal-independent. We can show that the set of optimal actions at $[s,g] \in S \times G$, operating in $M^\dagger$, is identical to the set of optimal actions at s, operating in M.[2] Then, since the set of actions considered by $\pi^{\dagger *}$ at $[s,g]$ is optimal for all $[s,g] \in S \times G$, it is certainly optimal at $[s,\Gamma(s)]$.

Theorem 1 establishes that it is reasonable to execute $\pi^{\dagger *}_\Gamma$ in M with any $\Gamma(s)$, so long as $\Gamma(S) \subseteq G_0$. In our experiments, we assume that the time step is part of the state and adopt $\Gamma(s_t) = s^e_{t+1}$.[3] Since D-Shape gathers data from the environment using the set of demonstration states as goals, and stores those transitions in the replay buffer, the technical condition that the goals G_0 were seen during learning is satisfied. While our choice of $\Gamma(s)$ may not matter for optimal policies, nevertheless, we use demonstration states as inference-time goals, based on the intuition that demonstration states are useful goals.

5 EXPERIMENTAL EVALUATION

We now empirically study the behavior of D-Shape in gridworld domains. We investigate the following questions:

- Does D-Shape improve sample efficiency over RL alone?
- How are D-Shape's convergence properties affected by sub-optimal demonstrations?

Our results show that D-Shape improves sample efficiency over baseline RL methods, and can converge to the optimal policy more

[2] Indeed, the set of optimal actions at $[s,g]$ is also identical to the set of optimal actions at $[s,g']$ for any g' (operating in $M^\dagger$), but our proof relies on the set of optimal actions at s (operating in M).

[3] Appending the time step to the state does not alter optimal policies in an MDP [24]. This trick is commonly used to ensure that time-limited RL environments maintain the Markov property. Since the time step is defined as part of the state in M, the fact that our potential function is time-varying is not a concern. Nevertheless, note that Devlin and Kudenko [8] showed that the policy invariance property still holds under time-varying potentials.

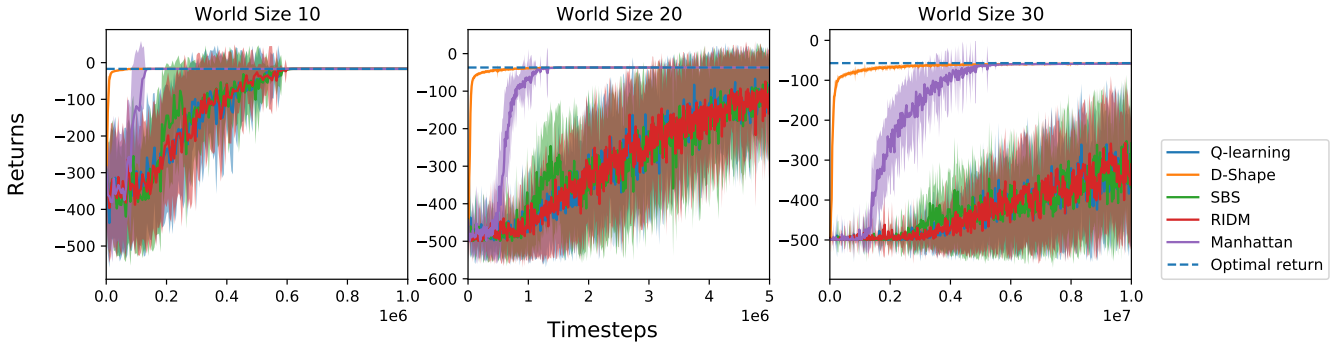

Figure 4: Learning curves of D-Shape (our method) using the optimal quality demonstrations, as compared to Q-learning (the base RL method), SBS, RIDM, and Manhattan. D-Shape improves sample efficiency over Q-learning and other baselines by a significant margin over all timesteps.

rapidly than baseline methods, despite suboptimal demonstrations that do not fully solve the task at hand. We also provide the results from an ablation study designed to evaluate the state augmentation and potential-based reward shaping components of D-Shape separately and together with goal-relabelling.

The main experimental procedures are described below; further experimental details are provided in the Supplemental Material.

5.0.1 Environments. The experiments are conducted on gridworlds with side lengths $s \in \{10, 20, 30\}$, where the task is for the agent to navigate from a fixed starting position to some goal position g. Unless otherwise specified, in our experiments, the starting position is one corner of the gridworld, while the goal position is the opposite corner. The state space consists of the agent's current position, and the action space consists of moving one square up, down, left, or right. If the agent attempts to move into the boundaries of the gridworld, its position will not change. The reward function is $r(s) = -\mathbb{1}_{s \neq g}$, and episodes terminate either when the agent reaches the goal position or when the episode time limit (500 timesteps) is reached.

These simple gridworld domains were chosen because they possess the key characteristics of having a sparse reward function, and therefore being challenging to explore. Further, it is easy to construct optimal and suboptimal demonstrations for this setting, and the state/action spaces are discrete and finite, matching the setting described in the theoretical analysis of D-Shape provided above. Experiments were also conducted on a stochastic version of this gridworld, with similar results and findings (see Supplemental Material).

5.0.2 Baselines. For fair comparison, Q-learning [36] is used as the base RL algorithm for all methods and experiments. D-Shape is compared against RL alone as well as RL+IfO methods that share similar policy invariance guarantees and assumptions. D-Shape is also compared against RL with a naive RL+IfO reward, where the reward is defined as the sum of the task reward and an imitation reward. Applying RL with a hybrid task and imitation reward is a common technique to combine RL and IL, but typically requires hyperparameter search and does not provide any robustness guarantees [14, 39, 40]. The specific algorithms used as baselines in our experiments are listed below:

- *Q-learning* [36]: a classic model-free, off-policy RL method.
- *Reinforced Inverse Dynamics Modelling (RIDM)* [26]: a model-based RL+IfO algorithm that also assumes only a single demonstration.
- *Similarity Based Shaping (SBS)* [6] : an RL+IL algorithm that incorporates demonstrations through a potential-based shaping reward. We adapt SBS to the discrete setting, and for state-only demonstrations.
- *Q-learning + Manhattan (Manhattan)*: We construct an imitation reward term that consists of the negative Manhattan distance between the state s_t and the demonstration state s^e_{t+1}. Q-learning optimizes for the sum of the imitation reward and the task reward.

5.0.3 Demonstrations. Each algorithm is run with only a single demonstration. To study the ability of each algorithm to utilize suboptimal demonstrations, the experiments are performed with four demonstrations of varying qualities: optimal, good, medium, and worst. For the goal-based gridworld tasks considered, any state trajectory from the start position (which is always in the bottom left corner) to the goal position that decreases its distance from the goal by going right or up at each timestep, is an optimal demonstration. Thus, suboptimal demonstrations must consist of trajectories that either (1) go to an incorrect goal, or (2) do not decrease their distance to the goal at some timestep.

The experiments consider optimal demonstrations—which are manually computed—and suboptimal demonstrations of the first type. More specifically, the optimal demonstrations in the experiments are those that always follow the bottom and right edges of the gridworld; see Figure 3 for an example of the task and style of optimal demonstration used. The good, medium, and worst suboptimal demonstrations consist of state trajectories that go to alternative goals that are a Manhattan distance of 2, 4, and 6 steps away from the true goal g for the size 10 and size 20 gridworlds. For the size 30 gridworld, the alternative goals are a Manhattan distance of 4, 8 and 12 steps away from the goal. Experiments with further types

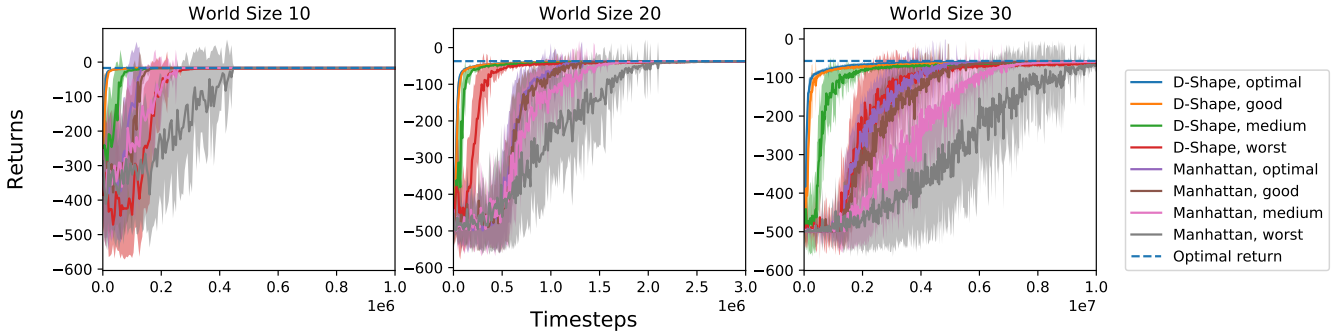

Figure 5: Learning curves of D-Shape (our method) and the Manhattan baseline when provided optimal, good, medium, and worst demonstrations. D-Shape converges to the optimal task reward consistently, despite being provided suboptimal demonstrations, and with better sample efficiency than Manhattan.

of suboptimal demonstrations (e.g. suboptimal demonstrations of the second type, demonstrations with missing states) may be found in the Supplemental Material.

5.0.4 Evaluation. D-Shape and baseline models are trained for a fixed number of time steps. To measure learning speed, the return achieved by each method is evaluated at regular intervals by averaging test returns from the current policy, and is plotted as a learning curve. The shaded region around each learning curve is the standard deviation of the average return. All curves in the following results are computed over 30 independent training runs. The optimal task return is plotted as a horizontal line.

5.1 D-Shape Improves Sample Efficiency

Figure 4 compares D-Shape to the baseline methods, where all methods are trained with a demonstration of optimal quality. It shows that, for all environments, D-Shape was able to improve sample efficiency over Q-learning, SBS, RIDM, and Manhattan. The closest competing method is Q-learning with the Manhattan distance IfO reward. The strong performance of this baseline is not surprising, since the Manhattan IfO reward computed with an optimal demonstration is similar to the Manhattan distance of a state to the true goal state—where the latter quantity is actually the value function for this set of tasks [22]. This same Manhattan IfO reward is also the potential function used by D-Shape, yet D-Shape also includes the component of goal relabelling, which may explain why D-Shape's sample efficiency is much higher than that of Manhattan's. The relative contribution of each of D-Shape's components to sample efficiency is further studied in the ablation section.

Figure 6 shows the state visitation distribution of a D-Shape agent on the 20×20 task, trained with an optimal demonstration analogous to the one shown in Figure 3. The visitation distribution has highest concentration along the bottom and right edges of the gridworld, which is precisely the path that the demonstration traverses. This confirms that the D-Shape algorithm biases the learning agent to follow the path of the demonstration (rather than following any of the other optimal trajectories for this task).

5.2 D-Shape's Convergence and Sample Efficiency is Robust to Demonstration Quality

The theory provided in Section 4 suggests that D-Shape should converge to the optimal policy even when provided suboptimal demonstrations. This section examines (1) whether the prior theoretical property is exhibited empirically by our implementation of D-Shape, and (2) how the sample efficiency demonstrated by D-Shape in the prior section is affected by suboptimal demonstrations.

More precisely, the theory states that the set of goals D-Shape explores and learns with should not affect D-Shape's ability to learn an optimal policy, as long the goal g at time $t+1$ depends only on the goal, state, and action at time t. We hypothesize that this theoretical guarantee means that D-Shape will still converge to the optimal task reward, no matter the demonstration quality used during training.

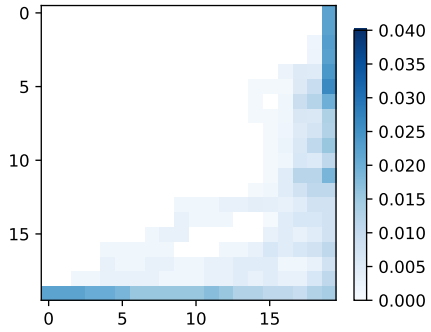

Figure 6: State visitation of D-Shape trained with the optimal demonstration for the 20×20 gridworld, where the start position is in the lower leftmost position, and the goal position is in the upper rightmost position. The provided demonstration followed the lower edge to the right, and then went up to the goal state, which is reflected by the state visitation of the learned D-Shape policy.

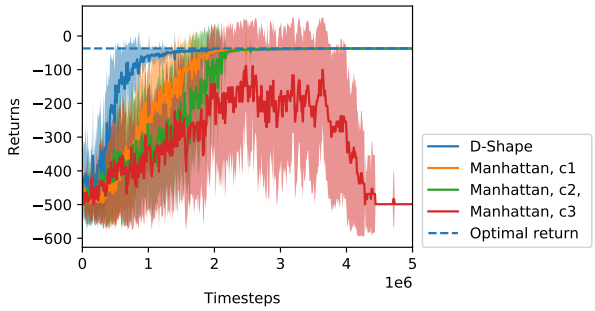

Figure 7: Varying the coefficient on the Manhattan distance IfO reward for the Q-learning + Manhattan distance baseline, for $\vec{c} = [1, 20, 25]$. The learning curve for D-Shape is shown for reference. Note that for c_3, Q-learning+Manhattan actually diverges from the optimal task return.

To test the hypothesis, D-Shape is trained with four demonstration qualities: optimal, good, medium, and worst. In this experiment, D-Shape is compared to the naive RL + IfO baseline, and Q-learning with a Manhattan distance reward. The same hyperparameters are used for all four demonstration qualities; the hyperparameter tuning procedure is described in the Supplemental Material.

Figure 5 shows that D-Shape and Q-learning+ Manhattan both converge to the optimal task return for all demonstrations, despite the fact that only the optimal demonstration fully completes the task. Given the demonstration quality, D-Shape has strictly better sample efficiency and lower learning variance than Q-learning+Manhattan. While D-Shape's theoretical properties guarantee that it should be able to converge to the optimal policy regardless of the demonstration quality, the Q-learning+Manhattan baseline boasts no such convergence guarantees. In fact, given a poorly chosen value of the c hyperparameter and a suboptimal demonstration, the learning process of the Q-learning+Manhattan baseline might actually diverge, as the optimal policy has changed (see Figure 7).

5.3 Ablation Study

D-Shape draws a connection to GCRL and uses *goal-relabelling* (GR) to unify *state augmentation* and *potential-based reward shaping* from demonstrations (shaping). As such, D-Shape can be thought of as the unification of these three components (GR, state-augmentation, and shaping).

This section investigates three ablations of D-Shape: [D-Shape − GR], [D-Shape − GR − shaping], and [D-Shape − GR − state augmentation]. Note that these are the only three ablations worth studying. [D-Shape − shaping] is not viable because GR requires a goal-reaching reward; similarly [D-Shape − state augmentation] cannot be studied because goals must be presented to the agent. [D-Shape − shaping − GR − state augmentation] is equivalent to baseline Q-learning.

The results are shown in Figure 8. In all domains, D-Shape dominates all three ablations. Two of the ablations— [D-Shape − GR − state augmentation] and [D-Shape − GR] — perform similarly. The final ablation, [D-Shape − GR − shaping] performed much more poorly than the others, showing similar sample efficiency to the baseline of Q-learning. Therefore, this ablation is not shown on Figure 8; see the Supplemental Material to view all ablations together. The results here imply that for the gridworld domain studied, a large part of D-Shape's improvement in sample efficiency over baseline methods comes from potential-based shaping with demonstrations, and the remaining benefit comes from its GR component. While state augmentation alone does not seem to produce any gains in sample efficiency, note that it is not possible to perform goal-relabelling without augmenting the state with the goal description. Thus, we conclude that GR is a crucial component of D-Shape and has a synergistic effect on state augmentation and potential-based reward shaping.

6 RELATED WORK

D-Shape is closely related to methods from the RL+IfO literature and the GCRL literature. Here, we review work from the RL+IfO literature that is robust to demonstration quality, and work from the GCRL literature that uses demonstrations.

6.0.1 Combining RL and IL. A common approach to combining RL and IL is to define a hybrid reward function that is the sum of the task reward and an imitation reward [14, 39]. Imitation rewards proposed in the literature are derived from similarity measures [30], behavior cloning [3, 33], inverse RL [1, 5], or generative adversarial IL [9, 17, 34]. However, if demonstrations are suboptimal, the imitation objective may conflict with the task objective. The research community has proposed various techniques to remedy this problem. Simpler solutions consist of tuning the relative weights between the imitation and task reward and/or annealing the imitation reward to zero as training progresses [9]. Zolna et al. [40] introduce a method to automatically control the tradeoff between imitation and task objectives during training with Bernoulli random variables. While each of these methods has demonstrated some experimental success, each introduces a new set of hyperparameters that must be carefully tuned.

Others in the research community have defined hybrid rewards using potential-based reward shaping (PBRS) [22], and have demonstrated that the policy invariance guarantees of PBRS result in robustness to demonstration quality. Brys et al. [6] defines a potential function for reward shaping, where the potential function is a similarity measure between agent states and expert states. Similarly, Suay et al. [31] use inverse RL to recover a linear reward function from the demonstration, and formulate a dynamic potential function from the recovered reward. Wu et al. [37] explore using generative modelling techniques to construct a potential-based shaping function. Combining PBRS with goal relabelling has been relatively understudied. This work explores using potential-based reward functions as goal-reaching rewards, where the goal is given by demonstration states.

There are several other approaches for combining RL and IL that avoid potential conflict between imitation and task rewards by optimizing only the task reward, and incorporating demonstration knowledge elsewhere in the learning process. For instance, state augmentation (SA) approaches concatenate the agent state with a representation of the demonstration state(s) [23, 26]. Initialization approaches incorporate demonstration knowledge through

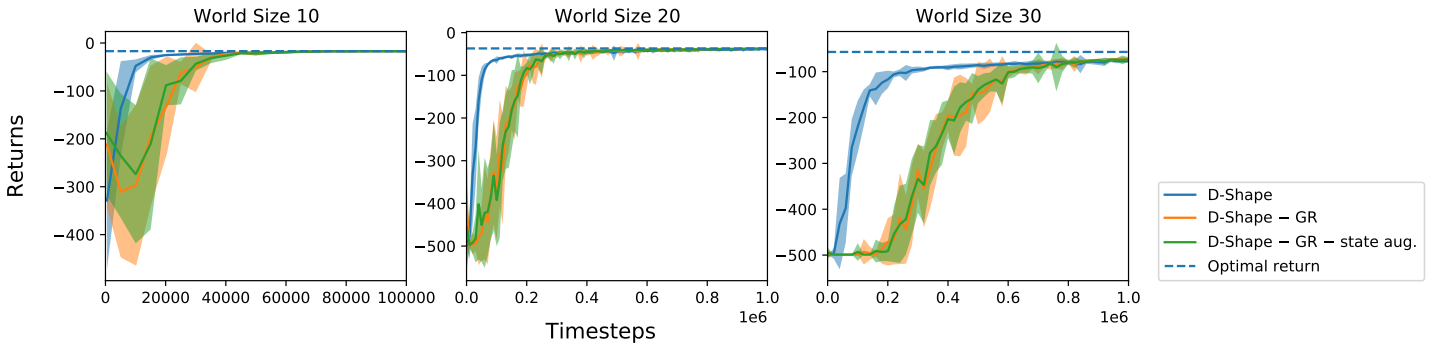

Figure 8: D-Shape compared to two ablations of D-Shape using the optimal demonstration. D-Shape displays better sample-effiency than all ablations. The last ablation ([D-Shape − GR − shaping]) performed no better than the baseline of Q-learning alone. It is not shown here because the difference between the other ablations would not be visible; see the Supplemental Material for a version with all ablations.

pretraining the policy or value functions [16, 32]. Resetting to demonstration states is another technique that has been shown to be effective for improving the sample efficiency and asymptotic performance of RL [10, 21, 27]. While the aforementioned techniques have demonstrated success with optimal demonstrations, they have not been explicitly evaluated for their robustness to suboptimal demonstrations. This work uses SA as a component of D-Shape, and our theoretical results directly apply to SA, showing that SA should not modify the optimal policy with respect to the task reward. Therefore, this work provides theoretical grounding for the intuition that SA is robust to demonstration quality.

6.0.2 Accelerating GCRL with Demonstrations. There is some work in the GCRL literature that investigates using demonstrations to speed up learning in classic GCRL tasks (e.g., maze tasks, target-reaching tasks). For example, Nair et al. [21] leverage state-action demonstrations in a GCRL algorithm through a demonstration buffer, an auxiliary behavioral cloning loss with a Q-filter, and resetting to demonstration states. Ding et al. [9] incorporate demonstrations into a GCRL algorithm by adding an imitation reward derived from generative adversarial IL to a sparse goal-reaching reward. Plan-based reward shaping methods consider designing potential-based shaping rewards when given access to an approximate plan or demonstration[6, 29]. Most similar to this work, Paul et al. [25] learn subgoals from a set of demonstration trajectories and define a potential-based shaping function from the subgoals. Unlike Paul et al. [25], we study goal-relabelling, demonstrate the efficacy of D-Shape using only a single expert demonstration, and examine using demonstration states directly as goals in a time-aligned manner.

7 CONCLUSION

In this work, we investigated using techniques from goal-conditioned reinforcement learning to combine potential-based reward shaping and state augmentation, and introduced D-Shape, a novel RL+IfO algorithm. We showed theoretically that an optimal policy learned using D-Shape can be executed optimally with an arbitrary sequence of goals, and demonstrated empirically that D-Shape improves sample efficiency and is robust to suboptimal demonstrations in gridworld domains. In our experiments, D-Shape was able to (1) improve sample efficiency over all baselines, and (2) consistently converge to the optimal policy with better sample efficiency than baselines, when provided with suboptimal demonstrations.

This paper evaluates D-Shape based on a sample efficiency criterion in discrete, goal-based gridworlds. Future work could assess D-Shape's ability to generalize to new demonstrations on more complex gridworld tasks, such as those with obstacles. Another direction of future work is extending D-Shape to continuous state-action spaces. In this work, we augment agent states with demonstration states as goals, where the demonstration states have the same representation as the agent states, but learned goal representations may be necessary to extend D-Shape to continuous state-action spaces. A limitation of D-Shape is that if the provided potential function does not improve the reinforcement learner's performance, then D-Shape also will not improve performance. Thus, learning potential functions—an active research area [13, 19]—is an important future direction. Finally, this work investigates D-Shape with a single demonstration only, but there is a rich set of techniques from GCRL that could potentially be used to incorporate multiple demonstrations into D-Shape.

ACKNOWLEDGMENTS

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by the National Science Foundation (CPS-1739964, IIS-1724157, FAIN-2019844), the Office of Naval Research (N00014-18-2243), Army Research Office (W911NF-19-2-0333), DARPA, Lockheed Martin, General Motors, Bosch, and Good Systems, a research grand challenge at the University of Texas at Austin. The views and conclusions contained in this document are those of the authors alone. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *ICML*. PMLR.

[2] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. 2017. Hindsight Experience Replay. In *NeurIPS*. Curran Associates.

[3] Michael Bain and Claude Sammut. 1999. A Framework for Behavioural Cloning. *Machine Intelligence* 15 (1999), 103–129.

[4] Daniel S. Brown, Wonjoon Goo, P. Nagarajan, and S. Niekum. 2019. Extrapolating Beyond Suboptimal Demonstrations via Inverse Reinforcement Learning from Observations. In *ICML*. PMLR.

[5] Daniel S. Brown, Wonjoon Goo, and S. Niekum. 2019. Better-than-Demonstrator Imitation Learning via Automatically-Ranked Demonstrations. In *CoRL*. PMLR.

[6] T. Brys, A. Harutyunyan, Halit Bener Suay, S. Chernova, Matthew E. Taylor, and A. Nowé. 2015. Reinforcement Learning from Demonstration through Shaping. In *IJCAI*. IJCAI.

[7] Letian Chen, Rohan R. Paleja, and M. Gombolay. 2020. Learning from Suboptimal Demonstration via Self-Supervised Reward Regression. In *CoRL*. PMLR.

[8] Sam Devlin and D. Kudenko. 2012. Dynamic potential-based reward shaping. In *AAMAS*. IFAAMAS.

[9] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. 2019. Goal-conditioned Imitation Learning. In *NeurIPS*. Curran Associates.

[10] Adrien Ecoffet, Joost Huizinga, J. Lehman, Kenneth O. Stanley, and J. Clune. 2021. First return, then explore. *Nature* 590 (2021), 580–586.

[11] Seyed Kamyar Seyed Ghasemipour, R. Zemel, and S. Gu. 2019. A Divergence Minimization Perspective on Imitation Learning Methods. In *CoRL*. PMLR.

[12] M. Grzes. 2017. Reward Shaping in Episodic Reinforcement Learning. In *AAMAS*. IFAAMAS.

[13] Marek Grzes and Daniel Kudenko. 2010. Online learning of shaping rewards in reinforcement learning. *Neural networks : the official journal of the International Neural Network Society* 23 4 (2010), 541–50.

[14] Xiaoxiao Guo, Shiyu Chang, Mo Yu, G. Tesauro, and Murray Campbell. 2019. Hybrid Reinforcement Learning with Expert State Sequences. In *AAAI*. AAAI Press.

[15] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9, 2 (2001), 159–195.

[16] Todd Hester, Matej Vecerík, O. Pietquin, Marc Lanctot, T. Schaul, Bilal Piot, Dan Horgan, John Quan, A. Sendonaris, Ian Osband, Gabriel Dulac-Arnold, J. Agapiou, Joel Z. Leibo, and A. Gruslys. 2018. Deep Q-learning From Demonstrations. In *AAAI*. AAAI Press.

[17] Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. In *NeurIPS*. Curran Associates.

[18] Leslie Pack Kaelbling. 1993. Learning to Achieve Goals. In *IJCAI*. IJCAI.

[19] Seyed Mohammad Khansari-Zadeh and Oussama Khatib. 2017. Learning potential functions from human demonstrations with encapsulated dynamic and compliant behaviors. *Autonomous Robots* 41 (2017), 45–69.

[20] Yuxuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. 2018. Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation. In *ICRA*. IEEE.

[21] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and P. Abbeel. 2018. Overcoming Exploration in Reinforcement Learning with Demonstrations. In *ICRA*. IEEE.

[22] Andrew Y. Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*. PMLR.

[23] T. Paine, Sergio Gomez Colmenarejo, Ziyu Wang, Scott E. Reed, Yusuf Aytar, T. Pfaff, Matthew W. Hoffman, Gabriel Barth-Maron, Serkan Cabi, D. Budden, and N. D. Freitas. 2018. One-Shot High-Fidelity Imitation: Training Large-Scale Deep Nets with RL. *ArXiv* abs/1810.05017.

[24] Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. 2018. Time Limits in Reinforcement Learning. In *ICML*. PMLR.

[25] Sujoy Paul, Jeroen Vanbaar, and Amit Roy-Chowdhury. 2019. Learning from Trajectories via Subgoal Discovery. In *NeurIPS*. Curran Associates.

[26] Brahma S. Pavse, Faraz Torabi, Josiah Hanna, Garrett Warnell, and Peter Stone. 2020. RIDM: Reinforced Inverse Dynamics Modeling for Learning from a Single Observed Demonstration. In *IROS*. IEEE.

[27] Tim Salimans and Richard J. Chen. 2018. Learning Montezuma's Revenge from a Single Demonstration. In *Workshop on Deep Reinforcement learning at NeurIPS*.

[28] Tom Schaul, Dan Horgan, K. Gregor, and D. Silver. 2015. Universal Value Function Approximators. In *ICML*. PMLR.

[29] Ingmar Schubert, Ozgur Oguz, and Mark Toussaint. 2021. Plan-Based Relaxed Reward Shaping for Goal-Directed Tasks. In *ICLR*.

[30] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, S. Schaal, and Sergey Levine. 2018. Time-Contrastive Networks: Self-Supervised Learning from Video. In *ICRA*. IEEE.

[31] Halit Bener Suay, T. Brys, Matthew E. Taylor, and S. Chernova. 2016. Learning from Demonstration for Shaping through Inverse Reinforcement Learning. In *AAMAS*. IFAAMAS.

[32] Matthew E. Taylor, Halit Bener Suay, and S. Chernova. 2011. Integrating reinforcement learning with human demonstrations of varying ability. In *AAMAS*. IFAAMAS.

[33] Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Behavioral Cloning from Observation. In *IJCAI*. IJCAI.

[34] Faraz Torabi, Garrett Warnell, and Peter Stone. 2019. Generative Adversarial Imitation from Observation. In *Imitation, Intent, and Interaction (I3) Workshop at ICML*. PMLR.

[35] Faraz Torabi, Garrett Warnell, and Peter Stone. 2019. Recent Advances in Imitation Learning from Observation. In *IJCAI*. IJCAI.

[36] Christopher John Cornish Hellaby Watkins. 1989. *Learning from delayed rewards*. Ph.D. Dissertation. University of Cambridge, England.

[37] Yuchen Wu, Melissa Mozifian, and Florian Shkurti. 2021. Shaping Rewards for Reinforcement Learning with Imperfect Demonstrations using Generative Models. In *ICRA*. IEEE.

[38] Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. 2019. Imitation Learning from Imperfect Demonstration. In *ICML*. PMLR.

[39] Yuke Zhu, Ziyu Wang, Josh Merel, Andrei Rusu, Tom Erez, Serkan Cabi, Saran Tunyasuvunakool, János Kramár, Raia Hadsell, Nando de Freitas, and Nicolas Heess. 2018. Reinforcement and Imitation Learning for Diverse Visuomotor Skills. In *RSS*. RSS.

[40] Konrad Zolna, N. Rostamzadeh, Yoshua Bengio, Sungjin Ahn, and Pedro H. O. Pinheiro. 2019. Reinforced Imitation in Heterogeneous Action Space. In *Imitation Learning and its Challenges in Robotics Workshop at NeurIPS*.

SUPPLEMENT

7.1 Proofs

We consider the finite-horizon MDP $M = (S, A, P, r, \gamma)$ with horizon H and the goal-conditioned version of M, $M^\dagger = (S \times G, A, P^\dagger, r^\dagger, \gamma)$. All terms are defined as in Sections 2 and 3. We prove two supporting propositions before proving the main result (Theorem 1).

Proposition 1. Let $\pi(\cdot \mid s)$ be a policy defined in M, and let $f(\pi)(\cdot \mid [s, g])$ be the extension of $\pi(\cdot \mid s)$ to $M^\dagger$. Then $V_\pi(s_t) = V_{f(\pi)}([s_t, g_t])\ \ \forall s_t \in S, g_t \in G$, and $t \in \{1, \dots H\}$.

Proof. Let s_t^g denote a state-goal pair, $[s_t, g_t]$. Let τ_t denote trajectories in M, i.e. $\tau_t = (s_t, a_t, \dots, s_H, a_H)$. Let τ_t^G denote (goal-conditioned) trajectories in $M^\dagger$, i.e. $(s_t^g, a_t, \dots, s_H^g, a_H)$. The conditional probability of a trajectory is

$$p_\pi(\tau_t \mid s_t, a_t) = \prod_{t'=t}^{H-1} \pi(a_{t'} \mid s_{t'}) P(s_{t'+1} \mid s_{t'}, a_{t'}).$$

The reward of a trajectory is defined as follows:

$$r(\tau_t) = \sum_{t'=t}^{H-1} \gamma^{t'-t} r(s_{t'+1}, a_{t'}, s_{t'}).$$

Notice that as the reward function does not depend on goals, $r^\dagger(\tau_t^G) = r(\tau_t)$.

Computing the value of $f(\pi)$ at s_t^g from definition:

$$\begin{aligned}
V_{f(\pi)}(s_t^g) &= \mathbb{E}_{\tau_t^G \sim p_{f(\pi)}}\left[r^\dagger(\tau_t^G) \mid s_t^g\right] \\
&= \sum_{\tau_t^G}\left[p_{f(\pi)}(\tau_t^G \mid s_t^g) r^\dagger(\tau_t^G)\right] \\
&= \sum_{\tau_t^G}\left[\Big(\prod_{t'=t}^{H-1} f(\pi)(a_{t'} \mid s_{t'}^g) P^\dagger(s_{t'+1}^g \mid s_{t'}^g, a_{t'})\Big) r^\dagger(\tau_t^G)\right] \\
&= \sum_{\tau_t^G}\Big[\Big(\prod_{t'=t}^{H-1} \pi(a_{t'} \mid s_{t'}) P(s_{t'+1} \mid s_{t'}, a_{t'}) \\
&\qquad P(g_{t'+1} \mid [s_{t'}, g_{t'}], a_{t'})\Big) r(\tau_t)\Big].
\end{aligned}$$

The outermost summation is over τ_t^G, representing summing over all values of $(s_t^g, a_t, \dots, s_H^g, a_H)$ — a summation over all possible trajectories in $M^\dagger$. We organize these trajectories into sets where the variables $\{s_{t'}, a_{t'}\}_{t'=t}^{H-1}$ are fixed, and only $\{g_{t'}\}_{t'=t}^{H-1}$ may vary. Let $\tau_t^{s,a} := \{s_{t'}, a_{t'}\}_{t'=t}^{H-1}$ (state-action sequences) and $\tau_t^{g|sa} := \{g_{t'} \mid s_{t'}, a_{t'}\}_{t'=t}^{H-1}$ (sequences of goals). Thus, we split the outermost summation over τ_t^G into two summations where the outer summation is over $\tau_t^{s,a}$ and the inner summation is over $\tau_t^{g|sa}$. Reorganizing the sum is permissible by the commutative property of addition and the fact that we are dealing with finite sums. Since the trajectory rewards, transition probabilities, and action probabilities only depend on $\tau_t^{s,a}$, we may factor it out of the inner summation.

Continuing the derivation from above:

$$\begin{aligned}
V_{f(\pi)}(s_t^g) &= \sum_{\tau_t^{s,a}} \sum_{\tau_t^{g|sa}}\Big[\Big(\prod_{t'=t}^{H-1} \pi(a_{t'} \mid s_{t'}) P(s_{t'+1} \mid s_{t'}, a_{t'}) \\
&\qquad P(g_{t'+1} \mid [s_{t'}, g_{t'}], a_{t'})\Big) r(\tau_t)\Big] \\
&= \sum_{\tau_t^{s,a}}\Big(\Big(\prod_{t'=t}^{H-1} \pi(a_{t'} \mid s_{t'}) P(s_{t'+1} \mid s_{t'}, a_{t'})\Big) r(\tau_t)\Big) \\
&\qquad \sum_{\tau_t^{g|sa}}\Big(\prod_{t'=t}^{H-1} P(g_{t'+1} \mid [s_{t'}, g_{t'}], a_{t'})\Big).
\end{aligned}$$

In the innermost summation over $\tau_t^{g|sa}$, we are left only with the product of conditional goal probabilities. This is precisely the probability of a goal sequence $p(\tau_t^{g|sa})$, given that the states and actions are fixed, which sums to 1. The remaining terms depend only on the sequence of states and actions, and are precisely $V_\pi(s_t)$, completing the proof.

$$\begin{aligned}
V_{f(\pi)}(s_t^g) &= \sum_{\tau_t^{s,a}}\Big(\Big(\prod_{t'=t}^{H-1} \pi(a_{t'} \mid s_{t'}) P(s_{t'+1} \mid s_{t'}, a_{t'})\Big) r(\tau_t)\Big) \\
&\qquad \sum_{\tau_t^{g|sa}} p(\tau_t^{g|sa}) \\
&= \sum_{\tau_t^{s,a}}\Big(\Big(\prod_{t'=t}^{H-1} \pi(a_{t'} \mid s_{t'}) P(s_{t'+1} \mid s_{t'}, a_{t'})\Big) r(\tau_t)\Big) \\
&= V_\pi(s_t).
\end{aligned}$$

□

Corollary 1.1. $Q_{f(\pi)}([s_t, g_t], a_t) = Q_\pi(s_t, a_t)$ for all $s_t \in S, g_t \in G, a_t \in A, t \in \{1, \dots H\}$.

Proof. The proof follows the same method as the proof for Proposition 1. □

Proposition 2. Let $\pi^*(\cdot \mid s)$ be an optimal policy in M, and let $f(\pi^*)(\cdot \mid [s, g])$ be its extension to $M^\dagger$. Then $f(\pi^*)$ is an optimal policy in $M^\dagger$.

Proof. We will prove the statement by showing that the value of $f(\pi^*)$ is already maximal.

By Corollary 1.1, $Q_{f(\pi^*)}([s, g], a) = Q_{\pi^*}(s, a)$. By definition of optimality, $\max_a Q_{\pi^*}(s, a) = V_{\pi^*}(s)$. By Proposition 1, $V_{\pi^*}(s) = V_{f(\pi^*)}(s)$. Applying these three facts,

$$\begin{aligned}
\max_a Q_{f(\pi^*)}([s, g], a) &= \max_a Q_{\pi^*}(s, a) \\
&= V_{\pi^*}(s) \\
&= V_{f(\pi^*)}([s, g]).
\end{aligned}$$

Since $\max_a Q_{f(\pi^*)}([s, g], a) = V_{f(\pi^*)}([s, g])$, the value of $f(\pi^*)([s, g])$ is already maximal for all $[s, g] \in S \times G$. Thus, $f(\pi^*)$ cannot be improved and must already be optimal in $M^\dagger$.

□

For reader convenience, the statement and proof of Theorem 1 is reproduced below.

Theorem 1. *Let $\pi^{\dagger *}(\cdot \mid [s, g])$ be an optimal policy in $M^\dagger$. Define an arbitrary state-goal mapping $\Gamma : S \mapsto G_0 \subseteq G$, where G_0 is the set of goals in $M^\dagger$ that has positive probability of being reached by any policy. Then $\pi_\Gamma^{\dagger *}(\cdot \mid s) := \pi^{\dagger *}(\cdot \mid [s, \Gamma(s)])$ is an optimal policy in M.*

Proof. We prove the statement by considering the optimal state-action value functions in M and $M^\dagger$. Denote the set of optimal actions at state-goal $[s, g] \in M^\dagger$ by $A_{s,g} = \arg\max_a Q^{\dagger *}([s, g], a)$.

Let $\pi^*(\cdot \mid s)$ be an optimal policy in M, and let $f(\pi^*)$ denote the policy induced by π^* in $M^\dagger$ as in Proposition 2. By Proposition 2, $f(\pi^*)$ is an optimal policy in $M^\dagger$. Since the optimal value function in $M^\dagger$ is unique, $Q_{f(\pi^*)}([s, g], a) = Q^{\dagger *}([s, g], a)$ for all $s \in S$, $g \in G$, and $a \in A$. By Corollary 1.1, $Q_{f(\pi^*)}([s, g], a) = Q_{\pi^*}(s, a)$. Thus,

$$\begin{aligned} A_{s,g} &\coloneqq \arg\max_a Q^{\dagger *}([s, g], a) \\ &= \arg\max_a Q_{f(\pi^*)}([s, g], a) \\ &= \arg\max_a Q_{\pi^*}(s, a). \end{aligned}$$

The computation shows that the set of optimal actions in $M^\dagger$ at some $[s, g] \in S \times G$ is the same as the set of optimal actions in M at s — that is, the set of optimal actions is independent of the goal g. Since the set of actions with positive probability under $\pi^{\dagger *}(\cdot \mid [s, g])$ is a subset of $A_{s,g}$ for $g \in G_0$, we have $\pi^{\dagger *}_\Gamma(\cdot \mid s) \coloneqq \pi^{\dagger *}(\cdot \mid [s, \Gamma(s)]) \subset A_{s,\Gamma(s)} = \arg\max_a Q_{\pi^*}(s, a)$ for all $s \in S$. Thus, $\pi^{\dagger *}_\Gamma$ is an optimal policy in M.

□

We note that there is an alternative way to think about obtaining optimal policies in M from an optimal policy in $M^\dagger$. We could prove, given that the rewards in $M^\dagger$ are agnostic to the goals, that an optimal policy in $M^\dagger$ is still optimal if we change the goal transition probabilities (while maintaining that the same $[s, g]$ have positive support). Then we could modify $M^\dagger$ to an MDP that is equivalent to M by just changing the the goal transition probabilities, and thus we would obtain an optimal policy in M.

7.2 Further Experimental Results

Figure 9 reproduces the ablation study presented in the main paper, but *with* RIDM as well. The remainder of this section provides further experiments supporting the results in the main paper.

7.2.1 Stochastic Environments. We also ran the main experiments on a stochastic version of the gridworld environment in the main paper, with similar findings. The results are shown in Figure 10. The main difference from the deterministic setting is that the learning curves are higher variance, demonstrating the validity of our approach even in stochastic environments.

7.2.2 Further Types of Suboptimal Demonstrations. The experiments in Section 5.2 show that for the first type of suboptimal demonstrations mentioned in the main paper (those that go to an incorrect goal state), D-Shape's performance declines less than that of the Manhattan baseline. Further, DShape always converges to the optimal return, unlike Manhattan. Here, we examine whether D-Shape is robust to the second type of suboptimal demonstrations (those that do not decrease their distance to the goal at some timestep).

Two demonstration styles that satisfy this second type of suboptimality are considered here. Figure 11 shows how D-Shape handles demonstrations consisting of an optimal path to the goal tile, where each state is repeated N times (N in $\{1, 3, 5\}$), as compared to the Manhattan baseline. Similar to Figure 5, we find that D-Shape's convergence and sample efficiency is more robust to suboptimal demonstrations than Manhattan.

We also consider demonstrations that are *missing* states, an interesting type of suboptimality that may come up in real applications (Figure 12). Interestingly, we find that D-Shape is strongly robust to this type of suboptimality, showing almost no decline in performance. We hypothesize this is because D-Shape treats each state as its own goal, so it is be able to handle demonstration trajectories with missing states.

7.3 Experimental Details

7.3.1 Base RL Algorithm (Q-learning). We use our own implementation of Q-learning with a tabular representation of the value function, where updates are performed based on data sampled from a replay buffer. Hyperparameters for baseline Q-learning are given in Table 1. Since all the methods in this paper use Q-learning as the base RL algorithm, these hyperparameters are also used there.

Hyperparameter	Value
training timesteps	250000
max timesteps per episode	500
ϵ	0.2
α	0.1
γ	1
updates per step	20
buffer size	5000

Table 1: Hyperparameters for the baseline of Q-learning. α is the learning rate, ϵ is the parameter for ϵ-greedy exploration, and γ is the discount factor.

7.3.2 PBRS in episodic settings. Ng et al. [22] proved their result in the continuing setting. Grzes [12] showed that in order for the policy invariance result to hold in the episodic setting, we must set $\phi(s_N) = 0$ for all terminal states s_N. We follow this recommendation for all methods that use a potential-based reward function (D-Shape, D-Shape ablations with potential-based shaping, and SBS).

7.3.3 D-Shape (our method). Transitions in the replay buffer are of the form, $([s_t, g_t], a_t, r_t, [s_{t+1}, g_{t+1}])$. One assumption in the theory behind D-Shape is that g_{t+1} depends only on $([s_t, g_t], a_t)$. Goals in the replay buffer come from two sources: the expert demonstration, or are sampled using the goal sampling strategy. We argue below that our implementation of D-Shape approximately satisfies the independence assumption.

Goals either originate from the goal-sampling strategy or are expert states. D-Shape is implemented with a variant of the `episode` goal sampling strategy [2]. The variant consists of uniformly sampling subsequent states at time k, (s_k, s_{k+1}) from the current episode to use as goal states at time t, (g_t, g_{t+1}). Since s_{k+1} depends on s_k, a_k, thus, g_{t+1} depends only on (g_t, a_k). In future work, we plan to investigate alternative goal sampling strategies that do not rely on the action a_k. The remaining transitions have time-aligned expert states as goal states, i.e. $(g_t, g_{t+1}) = (s^e_t, s^e_{t+1})$. Since our implementation

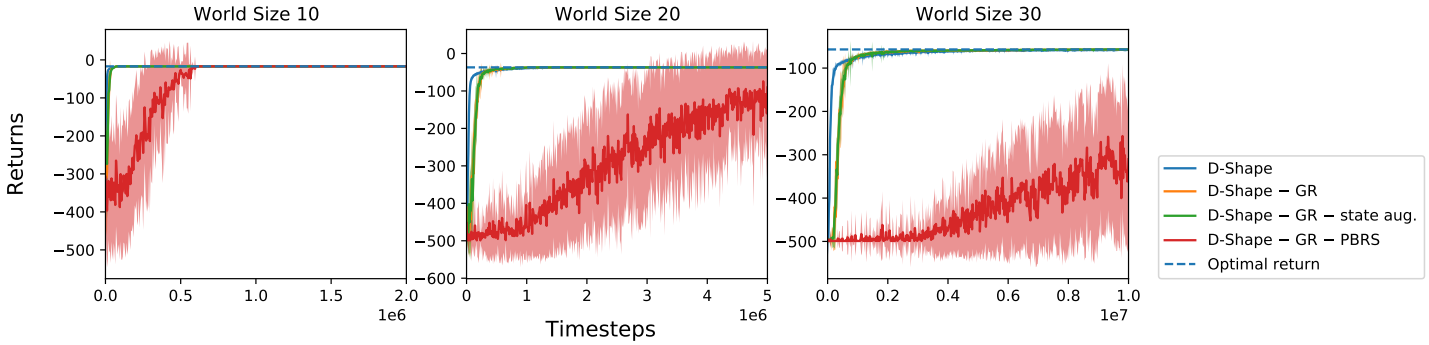

Figure 9: D-Shape compared to all three ablations of D-Shape using the optimal demonstration. D-Shape displays better sample-effiency than all ablations. The last ablation (DShape without PBRS or goal relabelling, i.e. state augmentation only) is shown on this plot. Optimal task return is shown as a horizontal line. Means and standard deviations are computed over 30 training runs.

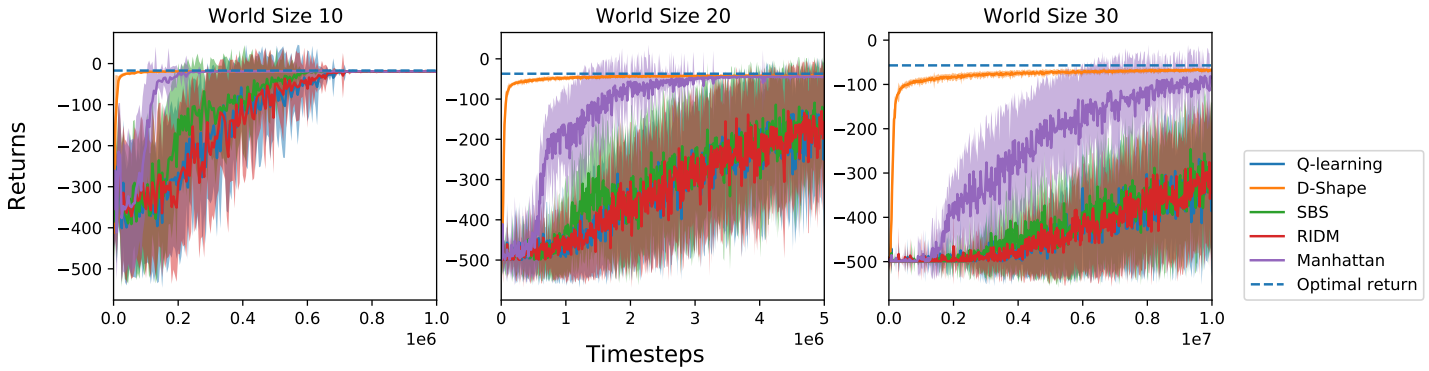

Figure 10: Learning curves of D-Shape with optimal quality demonstrations as compared to baseline methods, on a gridworld with stochastic transitions, where with 0.1 probability, a random action will be taken rather than that selected by the policy.

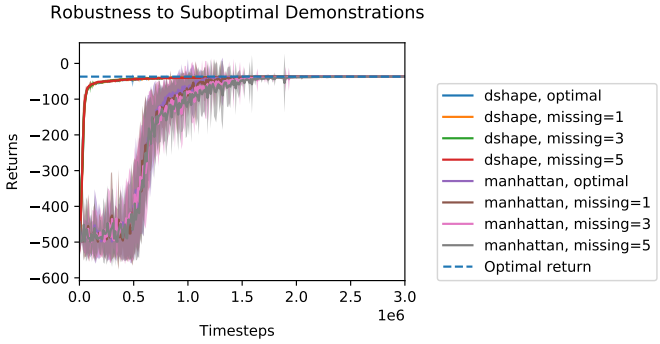

Figure 12: Learning curves of D-Shape compared to Manhattan with suboptimal demonstrations that are missing information (i.e. trajectories are missing states). Results shown are for the 20×20 gridworld.

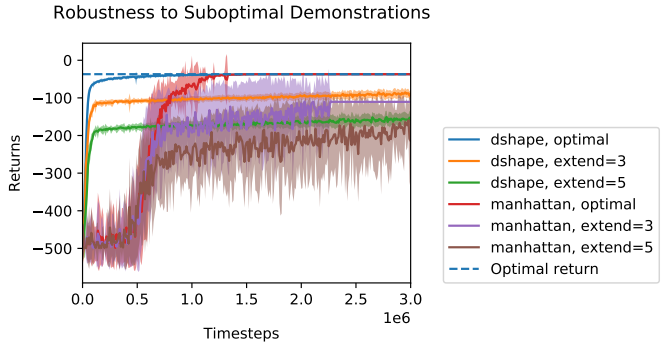

Figure 11: Learning curves of D-Shape compared to Manhattan with suboptimal demonstrations that never increase distance from the goal, but "linger" at each state for $N = \{1, 3, 5\}$ steps. Results shown are for the 20×20 gridworld.

includes timesteps are part of the state, s^e_{t+1} can be determined from s_t.

In the experiments, each sampled transition is relabelled with 3 goals. We found that relabelling with more goals did not improve performance.

7.3.4 RIDM [26]. Pavse et al. [26] implemented RIDM with CMA-ES, an evolutionary optimization algorithm [15]. Since the optimization technique is not fundamental to their method, for fair comparison, we instead implement RIDM using the base RL algorithm (Q-learning). This turns out to be equivalent to Q-learning with state augmentation. RIDM does not introduce any hyperparameters.

7.3.5 SBS [6]. The original SBS method defines a potential function in terms of a similarity measure between states from the demonstration s^d, and the agent's current state s, $m(s^d, s)$. Denote the set of demonstrations by D. The potential function is of the following form:

$$\phi^D(s, a) = \max_{(s^d, a) \in D} m(s, s^d).$$

This function computes the maximal similarity between agent states s and demonstration states s^d that come from samples with the same action as the agent. To implement SBS with state-only demonstrations, we simply remove the dependency on a:

$$\phi^D(s) = \max_{s^d \in D} m(s, s^d).$$

The particular similarity measure used by Brys et al. [6] is a non-normalized multi-variate Gaussian with mean s^d and covariance Σ,

$$g(s, s^d, \Sigma) = e^{(-\frac{1}{2}(s-s^d)^T \Sigma^{-1}(s-s^d))}.$$

In Brys et al. [6], $\Sigma \coloneqq \sigma I$, where σ is a constant, and state variables are mapped to $[0, 1]$. Thus, in practice, the similarity metric is the exponentiated negative Euclidean distance with a scaling parameter σ. Since the Euclidean distance metric is not a suitable distance metric for the discrete gridworld environments considered in our experiments, we instead replace it with the L_1 (Manhattan) distance. We also map the state variables into $[0, 1]$ and tune σ for each gridworld task. As suggested by Brys et al. [6], demonstration states are stored in a k-d tree and g is computed by querying the k-d tree.

Hyperparameter search procedure: We perform a hyperparameter search over σ and a reward scaling parameter for the shaping reward, c, using the 10×10 gridworld domain. The performance of each parameter combination is summarized by the sum of all returns along the training curve, an evaluation metric that balances between sample efficiency and asymptoptic return.

The search range is given below, and the final hyperparameter choice is bolded:

$$\sigma \in [0.1, 1, \mathbf{10}]$$
$$c \in [0.1, \mathbf{1}, 10].$$

7.3.6 Q-learning + Manhattan. This baseline consists of running baseline Q-learning with the following hybrid reward function:

$$r_t^{task} - c||s_t - s_{t+1}^e||_1.$$

The second term in the above reward function is the Manhattan distance of the current agent state from the next expert state, and is scaled by c.

The hybrid reward function has three hyperparameters: $\alpha, \beta, \gamma \in [0, 1]$. We perform a hyperparameter search over the following values using the same procedure as for SBS. The final selected value is bolded:

$$c \in [0.01, 0.1, \mathbf{1}].$$