Outcome-Driven Reinforcement Learning via Variational Inference

Tim G. J. Rudner [*1] **Vitchyr H. Pong** [*2] **Rowan McAllister** [2] **Yarin Gal** [1] **Sergey Levine** [2]

Abstract

While reinforcement learning algorithms provide automated acquisition of optimal policies, practical application of such methods requires a number of design decisions, such as manually designing reward functions that not only define the task, but also provide sufficient shaping to accomplish it. In this paper, we discuss a new perspective on reinforcement learning, recasting it as the problem of inferring actions that achieve desired outcomes, rather than a problem of maximizing rewards. To solve the resulting outcome-directed inference problem, we establish a novel variational inference formulation that allows us to derive a well-shaped reward function which can be learned directly from environment interactions. From the corresponding variational objective, we also derive a new probabilistic Bellman backup operator reminiscent of the standard Bellman backup operator and use it to develop an off-policy algorithm to solve goal-directed tasks. We empirically demonstrate that this method eliminates the need to design reward functions and leads to effective goal-directed behaviors.

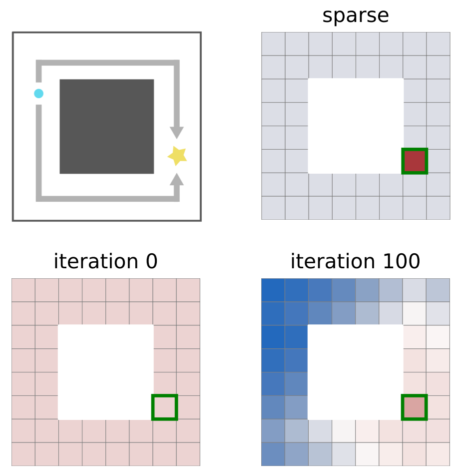


Figure 1: Illustration of the shaping effect of the reward function derived from the goal-directed variational inference objective. The top left figure shows a 2-dimensional grid world with a desired outcome marked by a star. The top right figure shows the corresponding sparse reward function, which is zero everywhere except at the desired outcome. The figures in the bottom row show the reward function derived from our variational inference formulation at initialization and at convergence. Unlike in the sparse reward setting, the derived reward function provides a dense reward signal everywhere in the state space.

1. Introduction

Reinforcement learning (RL) provides an appealing formalism for automated learning of behavioral skills, but requires considerable care and manual design to use in practice. One particularly delicate decision is the design of the reward function, which has a significant impact on the resulting policy but is largely heuristic in practice, often lacks theoretical grounding, can make effective learning difficult, and may lead to reward mis-specification.

To avoid these shortcomings, we propose to circumvent the process of manually specifying a reward function altogether: Instead of framing the reinforcement learning problem as finding a policy that maximizes a heuristically-defined reward function, we express it probabilistically, as inferring a state–action trajectory distribution conditioned on a desired future outcome. From this inference problem, we derive a tractable variational objective, an off-policy temporal-difference algorithm that provides a shaping-like effect for effective learning, as well as a reward function that captures the semantics of the underlying decision problem and facilitates effective learning.

We demonstrate that the temporal-difference algorithm, Outcome-Driven Actor–Critic (ODAC), is applicable to complex, high-dimensional continuous control tasks over finite and infinite horizons and has desirable theoretical and empirical properties. We show that in tabular settings, ODAC is guaranteed to converge to an optimal policy, and that in non-tabular settings with linear Gaussian transition dynamics, the derived optimization objective is convex in the policy, facilitating easier learning. In high-dimensional and non-linear domains, our method can be combined with deep

[*]Equal contribution [1]Department of Computer Science, University of Oxford, Oxford, United Kingdom [2]University of California, Berkeley, United States. Correspondence to: Tim G. J. Rudner <tim.rudner@cs.ox.ac.uk>, Vitchyr H. Pong <vitchyr@eecs.berkeley.edu>.

neural network function approximators to yield a deep reinforcement learning method that does not require manual specification of rewards, and leads to good performance on a range of benchmark tasks. Figure 1 shows the shaping-like effect under the proposed framework in a discrete state space.

Contributions. The core contribution of this paper is the probabilistic formulation of a general framework for inferring policies that lead to desired outcomes. We show that this formulation gives rise to a variational objective from which we derive a novel outcome-driven Bellman backup operator with a shaping-like effect that ensures a clear and dense learning signal even in early stages of training. Crucially, this "shaping" emerges automatically from a variational lower bound, rather than the heuristic approach for incorporating shaping that is often used in standard RL. We demonstrate that the resulting variational objective leads to an off-policy temporal-difference algorithm and evaluate it on a range of reinforcement learning tasks without having to manually specify task-specific reward functions. In our experiments, we find that our method results in significantly faster learning across a variety of robot manipulation and locomotion tasks than alternative approaches.

2. Preliminaries

Standard reinforcement learning (RL) addresses reward maximization in a Markov decision process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, p_0, p_d, r, \gamma)$ (Sutton and Barto, 1998; Szepesvári, 2010), where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action space, respectively, p_0 denotes the initial state distribution, p_d is a state transition distribution, r is an immediate reward function, and γ is a discount factor. To sample trajectories, an initial state $\mathbf{S}_0$ is sampled according to p_0, and successive states are sampled from the state transition distribution $\mathbf{S}_{t+1} \sim p_d(\cdot \,|\, \mathbf{s}_t, \mathbf{a}_t)$ and actions from a policy $\mathbf{A}_t \sim \pi(\cdot \,|\, \mathbf{s}_t)$. We will write $\boldsymbol{\tau}_{0:t} = \{\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \dots, \mathbf{s}_t\}$ to represent a finite-horizon and $\boldsymbol{\tau} \stackrel{\text{def}}{=} \{\mathbf{a}_t, \mathbf{s}_{t+1}\}_{t=0}^{\infty}$ to represent an infinite-horizon state–action trajectory realization. Given a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and discount factor $\gamma \in (0, 1)$, the objective in reinforcement learning is to find a policy π that maximizes the returns, defined as $\mathbb{E}_{p_\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right]$, where p_π denotes the distribution of states induced by a policy π.

Goal-Conditioned Reinforcement Learning. In goal-conditioned reinforcement learning (Kaelbling, 1993), which can be considered a special case of the broader class of stochastic boundary value problems (Aly and Chan, 1974; Goebel and Raitums, 1990), the objective is for an agent to reach some pre-specified goal state, $\mathbf{g} \in \mathcal{S}$, so that the policy and reward function introduced above become dependent on the goal and are expressed as $\pi(\mathbf{a} \,|\, \mathbf{s}, \mathbf{g})$ and $r(\mathbf{s}, \mathbf{a}, \mathbf{g})$, respectively. Typically, such a reward function needs to be defined manually, with a common choice being to use a sparse indicator reward $r(\mathbf{s}, \mathbf{g}) = \mathbb{I}\{\mathbf{s} = \mathbf{g}\}$. However, this approach presents a number of challenges both in theory and in practice. From a theoretical perspective, the indicator reward will almost surely equal zero for environments with continuous goal spaces and non-trivial stochastic dynamics. From a practical perspective, such sparse rewards can be slow to learn from, as most transitions provide no reward supervision, while manually designing dense reward functions that provide a better learning signal is time-consuming and often based on heuristics. In Section 3, we will present a framework that addresses these practical and theoretical considerations by casting goal-conditioned RL as probabilistic inference.

Q-Learning. Off-policy Q-learning algorithms (Watkins and Dayan, 1992) allow training policies from data collected under alternate decision rules by estimating the expected return Q^π conditioned on a state–action pair:

$$Q^\pi(\mathbf{s}, \mathbf{a}) \stackrel{\text{def}}{=} \mathbb{E}_{p_\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \middle| \mathbf{S}_0 = \mathbf{s}, \mathbf{A}_0 = \mathbf{a}\right].$$

Crucially, the expected return given a state–action pair can be expressed recursively as

$$Q^\pi(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \gamma\, \mathbb{E}_{p_\pi}[Q^\pi(\mathbf{s}_1, \mathbf{a}_1) \,|\, \mathbf{S}_0 = \mathbf{s}, \mathbf{A}_0 = \mathbf{a}], \tag{1}$$

which makes it possible to estimate the expectation on the right-hand side from single-step transitions. The resulting estimates can then be used to find a policy that results in actions which maximize the expected return $Q^\pi(\mathbf{s}, \mathbf{a})$ for all available state–action pairs.

3. Outcome-Driven Reinforcement Learning

In this section, we derive a variational inference objective to infer an approximate posterior policy for achieving desired outcomes. Instead of using the heuristic goal-reaching rewards discussed in Section 2, we will derive a general framework for *inferring actions that lead to desired outcomes* by formulating a probabilistic objective, using the tools of variational inference. As we will show in the following sections, we use this formulation to translate the problem of inferring a policy that leads to a desired outcome into a tractable variational optimization problem, which we show corresponds to an RL problem with a well-shaped, dense reward signal from which the agent can learn more easily. We build off of prior work on probabilistic approaches to RL (Attias, 2003; Toussaint and Storkey, 2006; Toussaint, 2009; Ziebart et al., 2008; Kappen et al., 2012; Rawlik et al., 2013; Fellows et al., 2019) and formulate a general probabilistic objective from which we then derive an entire off-policy RL algorithm—including both the reward and discount factor—solely by specifying a desired outcome.

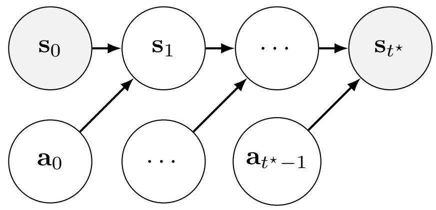


Figure 2: A Probabilistic graphical model of a state–action trajectory with observed random variables $\mathbf{s}_0$ and $\mathbf{s}_{t^\star}$.

We start with a warm-up problem that demonstrates how to frame the task of achieving a desired outcome as an inference problem in a simplified setting. We then describe how to extend this approach to more general settings. Finally, we show that the resulting variational objective can be expressed as a recurrence relation, which allows us to derive an outcome-driven variational Bellman operator and prove an outcome-driven probabilistic policy iteration theorem.

3.1. Warm-up: Achieving a Desired Outcome at a Fixed Time Step

We first consider a simplified problem, where the desired outcome is to reach a specific state $\mathbf{g} \in \mathcal{S}$ at a specific time step $t^\star$ when starting from initial state $\mathbf{s}_0$. We can think of the starting state $\mathbf{s}_0$ and the desired outcome $\mathbf{g}$ as boundary conditions, and the goal is to learn a stochastic policy that induces a trajectory from $\mathbf{s}_0$ to $\mathbf{g}$. To derive a control law that solves this stochastic boundary value problem, we frame the problem probabilistically, as inferring a state–action trajectory posterior distribution *conditioned on the desired outcome and the initial state.* We will show that, by framing the learning problem this way, we obtain an algorithm for learning outcome-driven policies without needing to manually specify a reward function. We consider a model of the state–action trajectory up to and including the desired outcome $\mathbf{g}$,

$$p_{\boldsymbol{\tau}_{0:t},\mathbf{S}_{t+1}}(\boldsymbol{\tau}_{0:t}, \mathbf{g} \,|\, \mathbf{s}_0) \stackrel{\text{def}}{=} p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) p(\mathbf{a}_t \,|\, \mathbf{s}_t) \cdot \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})), \quad (2)$$

where $t \stackrel{\text{def}}{=} t^\star - 1$, $p(\mathbf{a}_t \,|\, \mathbf{s}_t)$ is a conditional action prior, and $p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)$ is the environment's state transition distribution. If the dynamics are simple (e.g., tabular or Gaussian), the posterior over actions can be computed in closed form (Attias, 2003), but we would like to be able to infer outcome-driven posterior policies in any environments, including those where exact inference may be intractable. To do so, we start by expressing posterior inference as the variational minimization problem

$$\boxed{\min_{q \in \mathcal{Q}} D_{\text{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0, \mathbf{S}_{t^\star} = \mathbf{g})),} \quad (3)$$

where $\tilde{\boldsymbol{\tau}}_{0:t}$ is the state–action trajectory up to $t^\star$, but *excluding* $\mathbf{s}_0$ and $\mathbf{s}_{t^\star}$, $D_{\text{KL}}(\cdot \,\|\, \cdot)$ is the KL divergence, and $\mathcal{Q}$ denotes the variational family over which to optimize. We consider a family of distributions parameterized by a policy π and defined by

$$q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0) \stackrel{\text{def}}{=} \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}), \quad (4)$$

where $\pi \in \Pi$, a family of policy distributions, and where $\prod_{t=0}^{t-1} p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)$ is the true action-conditional state transition distribution up to but excluding the state transition at $t^\star - 1$, which we exclude from the variational distribution, since $\mathbf{S}_{t^\star} = \mathbf{g}$ is observed. Under this variational family, the inference problem in Equation (3) can be equivalently stated as the problem of maximizing the following objective with respect to the policy π:

Proposition 1 (Fixed-time Outcome-Driven Variational Objective)**.** *Let $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)$ be as defined in Equation (4). Then, given any initial state $\mathbf{s}_0$, termination time $t^\star$, and outcome $\mathbf{g}$, solving Equation (3) is equivalent to maximizing the following objective with respect to $\pi \in \Pi$:*

$$\boxed{\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) \stackrel{\text{def}}{=} \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, \mathbf{s}_0)}\Big[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \sum_{t'=0}^{t-1} D_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,||\, p(\cdot \,|\, \mathbf{s}_{t'}))\Big].} \quad (5)$$

Proof. See Appendix A.1. □

A variational problem of this form—which corresponds to finding a posterior distribution over actions)—can equivalently be viewed as a reinforcement learning problem:

Corollary 1. *The objective in Equation (5) corresponds to KL-regularized reinforcement learning with a time-varying reward function given by*

$$\boxed{r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') \stackrel{\text{def}}{=} \mathbb{I}\{t' = t\} \log p_d(\mathbf{g} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}).} \quad (6)$$

Corollary 1 illustrates how a reward function *emerges automatically* from a probabilistic framing of outcome-driven reinforcement learning problems where the sole specification is which variable ($\mathbf{S}_{t^\star}$) should attain which value ($\mathbf{g}$). In particular, Corollary 1 suggests that we ought to learn the environment's state-transition distribution, and view the log-likelihood of achieving the desired outcome given a state–action pair as a "reward" that can be used for off-policy learning as described in Section 2. Importantly—unlike in model-based RL—such a transition model would not have to be accurate beyond single-step predictions, as it would not be used for planning (see Appendix C). Instead, $\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t)$ only needs to be well shaped, which we expect to happen for commonly used model classes. For example, when the dynamics are linear-Gaussian, using a conditional Gaussian model (Nagabandi et al., 2018) yields a reward function that is quadratic in $\mathbf{S}_{t+1}$, making the objective convex and thus more amenable to optimization.

Thus far, we assumed that the time at which the outcome is achieved is given. In practice, however, we typically do not know (or care) when an outcome is achieved, and instead want to find policies that reach an outcome at *some* (possibly unknown) point in the future. This general setting is the primary focus of our work, and we will address it next.

3.2. Outcome-Driven Reinforcement Learning as Variational Inference

In this section, we present a variational inference perspective on achieving desired outcomes in settings where *no reward function* and *no termination time* are given, but only a desired outcome is provided. As in the previous section, we derive a variational objective and show that a principled algorithm and reward function emerge automatically when framing the problem as variational inference.

To derive such an objective, we need to modify the probabilistic model used in the previous section to accommodate settings where *the time at which the outcome is achieved is not known*. As before, we define an "outcome" as a point in the state space, but instead of defining the event of "achieving a desired outcome" as a realization $\mathbf{S}_{t^\star} = \mathbf{g}$ for a *known* $t^\star$, we define it as a realization $\mathbf{S}_{T^\star} = \mathbf{g}$ for an *unknown* "termination time" $T^\star$ at which the outcome is achieved.

We consider a probabilistic model of the state–action trajectory and the time step T immediately *before* the outcome is reached

$$p_{\tilde{\boldsymbol{\tau}}_{0:T},\mathbf{S}_T,T}(\tilde{\boldsymbol{\tau}}_{0:t},\mathbf{g},t\,|\,\mathbf{s}_0) = p_T(t)p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t)p(\mathbf{a}_t\,|\,\mathbf{s}_t) \cdot \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1}\,|\,\mathbf{s}_{t'},\mathbf{a}_{t'})p(\mathbf{a}_{t'}\,|\,\mathbf{s}_{t'}), \qquad (7)$$

where $t^\star = t+1$ and $p_T(t)$ is the probability of reaching the outcome at $t+1$. Since the trajectory length is itself a random variable, the joint distribution in Equation (7) is a *transdimensional* distribution defined on $\biguplus_{t=0}^{\infty}\{t\} \times \mathcal{S}^{t+1} \times \mathcal{A}^{t+1}$ (Hoffman et al., 2009).

Unlike in the warm-up, this problem of finding an outcome-driven policy corresponds to finding the posterior distribution over state–action trajectories *and* the termination time T conditioned on the desired outcome $\mathbf{S}_T$ and a starting state. This problem corresponds to jointly finding a trajectory distribution under which achieving the desired outcome is likely and a distribution over the time at which the outcome is achieved. Analogously to Section 3.1, we can express this inference problem variationally as

$$\min_{q \in \mathcal{Q}} D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t},T}(\cdot\,|\,\mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t},T}(\cdot\,|\,\mathbf{s}_0,\mathbf{S}_{T^\star}=\mathbf{g})), \qquad (8)$$

where t denotes the time immediately *before* the outcome is achieved and $\mathcal{Q}$ denotes the variational family. In general, solving this variational problem in closed form is challenging, but by choosing a variational family $q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\tilde{\boldsymbol{\tau}}_{0:t},t\,|\,\mathbf{s}_0) = q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t}\,|\,t,\mathbf{s}_0)q_T(t)$, where q_T is a distribution over T in some variational family $\mathcal{Q}_T$ and $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t}\,|\,t,\mathbf{s}_0)$ is given by Equation (4), we obtain an unknown-termination-time analogue to Proposition 1:

Proposition 2 (Unknown-time Outcome-Driven Variational Objective)**.** *Let $q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\tilde{\boldsymbol{\tau}}_{0:t},t\,|\,\mathbf{s}_0) = q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t}\,|\,t,\mathbf{s}_0)q_T(t)$, let $q_T(t)$ be a variational distribution defined on $t \in \mathbb{N}_0$, and let $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t}\,|\,t,\mathbf{s}_0)$ be as defined in Equation (4). Then, given any initial state $\mathbf{s}_0$ and outcome $\mathbf{g}$. Solving the variational optimization problem in Equation (8) is equivalent to maximizing the following variational objective with respect to $\pi \in \Pi$ and $q_T \in \mathcal{Q}_T$:*

$$\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g}) \stackrel{\text{def}}{=} \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_{0:t},t\,|\,\mathbf{s}_0)}\Big[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t) \qquad (9)$$
$$- D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot\,|\,\mathbf{s}_0)\,||\,p_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot\,|\,\mathbf{s}_0))\Big].$$

Proof. See Appendix A.2. □

Unfortunately, unlike in the fixed time step case, the objective in Proposition 2 cannot trivially be written in the form of Equation (1) and hence may not lend itself to off-policy reinforcement learning, making it of limited use in practice.

3.3. Deriving an Outcome-Driven Variational Objective

To derive a variational objective amenable to optimization via off-policy learning, we need to show that the variational objective in Equation (9) can be expressed in the form of Equation (1). To do so, we consider a variational family $\mathcal{Q}_T$ of variational distributions q_T parameterized by

$$q_T(t\,|\,\mathbf{s}_0) = q_{\Delta_{t+1}}(\Delta_{t+1}=1)\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0), \quad (10)$$

with Bernoulli random variables Δ_t denoting the event of "reaching $\mathbf{g}$ at time t given that $\mathbf{g}$ has not yet been reached by time $t-1$." With this variational distribution, we can now express a tractable variational bound on the objective in Equation (8) that is amenable to off-policy optimization:

Theorem 1 (Outcome-Driven Variational Inference)**.** *Let $q_T(t)$ and $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t}\,|\,t,\mathbf{s}_0)$ be as defined before, and define*

$$V^\pi(\mathbf{s}_t,\mathbf{g};q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}\left[Q^\pi(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_T)\right] - D_{\mathrm{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t)), \qquad (11)$$

$$Q^\pi(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_T) \stackrel{\text{def}}{=} r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1}=0)\,\mathbb{E}_{p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}\left[V^\pi(\mathbf{s}_{t+1},\mathbf{g};\pi,q_T)\right], \qquad (12)$$

$$r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) \stackrel{\text{def}}{=} q_{\Delta_{t+1}}(\Delta_{t+1}=1)\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t) - D_{\mathrm{KL}}(q_{\Delta_{t+1}}\,\|\,p_{\Delta_{t+1}}). \qquad (13)$$

Then given any initial state $\mathbf{s}_0$ *and outcome* $\mathbf{g}$*,*

$$\begin{aligned} &D_{\mathrm{KL}}(q_{q\tilde{\tau}_{0:t},T}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\tau}_{0:t},T}(\cdot \,|\, \mathbf{s}_0, \mathbf{S}_{T^\star} = \mathbf{g})) \\ &\quad = -\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) + C = -V^\pi(\mathbf{s}_0, \mathbf{g}; q_T) + C, \end{aligned}$$

where $C \stackrel{\text{def}}{=} \log p(\mathbf{g} \,|\, \mathbf{s}_0)$ *is independent of* π *and* q_T*, and hence maximizing* $V^\pi(\mathbf{s}_0, \mathbf{g}; \pi, q_T)$ *is equivalent to minimizing Equation (8). In other words,*

$$\begin{aligned} &\underset{\pi \in \Pi, q_T \in \mathcal{Q}_T}{\arg\min} \; \{D_{\mathrm{KL}}(q_{q\tilde{\tau}_{0:t},T}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\tau}_{0:t},T}(\cdot \,|\, \mathbf{s}_0, \mathbf{S}_{T^\star} = \mathbf{g}))\} \\ &\quad = \underset{\pi \in \Pi, q_T \in \mathcal{Q}_T}{\arg\max} \; \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) = \underset{\pi \in \Pi, q_T \in \mathcal{Q}_T}{\arg\max} \; V^\pi(\mathbf{s}_0, \mathbf{g}; q_T). \end{aligned}$$

Proof. See Appendix A.2. □

This theorem tells us that the maximizer of $V^\pi(\mathbf{s}_t, \mathbf{g}; q_T)$ is equal to the minimizer of Equation (8). In other words, Theorem 1 presents a variational objective with dense reward functions defined solely in terms of the desired outcome and the environment dynamics, which we can learn directly from environment interactions. Thanks to the recursive expression of the variational objective, we can find the optimal variational over T as a function of the current policy and Q-function analytically:

Proposition 3 (Optimal Variational Distribution over T)**.** *The optimal variational distribution* $q_T^\star$ *with respect to Equation (11) is defined recursively in terms of* $q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0) \forall t \in \mathbb{N}_0$ *by*

$$\begin{aligned} &q^\star_{\Delta_{t+1}}(\Delta_{t+1} = 0; \pi, Q^\pi) \\ &= \sigma\left(\Lambda(\mathbf{s}_t, \pi, q_T, Q^\pi) + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1} = 0)\right)\right), \end{aligned} \tag{14}$$

where

$$\begin{aligned} &\Lambda(\mathbf{s}_t, \pi, q_T, Q^\pi) \\ &\stackrel{\text{def}}{=} \mathbb{E}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T) - \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t)] \end{aligned} \tag{15}$$

with the expectation taken with respect to $\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t) \pi(\mathbf{a}_t \,|\, \mathbf{s}_t)$*, and* $\sigma(\cdot)$ *is the sigmoid function, that is,* $\sigma(x) = \frac{1}{e^{-x}+1}$ *and* $\sigma^{-1}(x) = \log \frac{x}{1-x}$*.*

Proof. See Appendix A.3 □

In the next section, we discuss how we can learn the Q-function in Theorem 1 using off-policy transitions by presenting a new type of Bellman operator.

4. Outcome-Driven Reinforcement Learning

In this section, we show that the variational objective in Theorem 1 is amenable to off-policy learning and that it can be estimated efficiently from single-step transitions. We then describe how to instantiate the resulting outcome-driven algorithm in large environments where function approximation is necessary.

4.1. Outcome-Driven Policy Iteration

To develop an outcome-directed model-free off-policy algorithm, we define the following Bellman operator:

Definition 1 (Variational Outcome-Driven Bellman Backup Operator)**.** *Let* $r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta)$ *be defined as in Theorem 1, let* $Q : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$*, and define the operator* $\mathcal{T}^\pi$ *as*

$$\begin{aligned} \mathcal{T}^\pi Q(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) &\stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \\ &\quad + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}\left[V(\mathbf{s}_{t+1}, \mathbf{g}; q_T)\right], \end{aligned} \tag{16}$$

where the expectation is w.r.t. $p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)$ *and*

$$\begin{aligned} V(\mathbf{s}_t, \mathbf{g}; q_T) &\stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)}\left[Q(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)\right] \\ &\quad + D_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)) \end{aligned} \tag{17}$$

Unlike the standard Bellman operator, the above operator has a varying weight factor $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$, with the optimal weight factor given by Equation (14). From Equation (14), we see that as the outcome likelihood $p_d(\mathbf{g} \,|\, \mathbf{s}, \mathbf{a})$ becomes large relative to the Q-function, the weight factor automatically adjusts the target to rely more on the rewards.

Below, we show that repeatedly applying the operator $\mathcal{T}^\pi$ (policy evaluation) and optimizing π with respect to Q^π (policy improvement) converges to a policy that maximizes the objective in Theorem 1.

Theorem 2 (Variational Outcome-Driven Policy Iteration)**.** *Assume* $|\mathcal{A}| < \infty$ *and that the MDP is ergodic.*

1. *Outcome-Driven Policy Evaluation (ODPE): Given policy* π *and a function* $Q^0 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$*, define* $Q^{i+1} = \mathcal{T}^\pi Q^i$*. Then the sequence* Q^i *converges to the lower bound in Theorem 1.*

2. *Outcome-Driven Policy Improvement (ODPI): The policy that solves*

$$\begin{aligned} \pi^+ = \underset{\pi' \in \Pi}{\arg\max} \{&\mathbb{E}_{\pi'(\mathbf{a}_t \,|\, \mathbf{s}_t)}\left[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)\right] \\ &- D_{\mathrm{KL}}(\pi'(\cdot \,|\, \mathbf{s}_t) \,||\, p(\cdot \,|\, \mathbf{s}_t)\} \end{aligned} \tag{18}$$

   and the variational distribution over T *recursively defined Equation (14) improve the variational objective. In other words,* $\mathcal{F}(\pi^+, q_T, \mathbf{s}_0) \geq \mathcal{F}(\pi, q_T, \mathbf{s}_0)$ *and* $\mathcal{F}(\pi, q_T^+, \mathbf{s}_0) \geq \mathcal{F}(\pi, q_T, \mathbf{s}_0)$ *for all* $\mathbf{s}_0 \in \mathcal{S}$*.*

3. *Alternating between ODPE and ODPI converges to a policy* $\pi^\star$ *and a variational distribution over* T*,* q_T*, such that* $Q^{\pi^\star}(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_T^\star) \geq Q^\pi(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_T)$ *for all* $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ *and any* $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$*.*

Proof. See Appendix B. □

This result tells us that alternating between applying the outcome-driven Bellman operator in Definition 1 and optimizing the bound in Theorem 1 using the resulting Q-function, which can equivalently be viewed as expectation maximization, will lead to a policy that induces an outcome-driven trajectory and solves the inference problem in Equation (8). As we discuss in Appendix B, this implies that Variational Outcome-Driven Policy Iteration is theoretically at least as good as or better than standard policy iteration for KL-regularized objectives.

4.2. Outcome-Driven Actor–Critic (ODAC)

We now build on previous sections to develop a practical algorithm that handles large and continuous domains. In such domains, the expectation in the Bellman operator in Definition 1 is intractable, and so we approximate the policy π_θ and Q-function Q_ϕ with neural networks parameterized by parameters θ and ϕ, respectively. We train Q-function to minimize

$$\mathcal{F}_Q(\phi) = \mathbb{E}\left[\Big(Q_\phi(\mathbf{s}, \mathbf{a}, \mathbf{g}) - (r(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_\Delta) + q_{\Delta_t}(\Delta_t = 0)\, \hat{V}(\mathbf{s}', \mathbf{g}))\Big)^2\right], \tag{19}$$

where the expectation is taken with respect to $(\mathbf{s}, \mathbf{a}, \mathbf{g}, \mathbf{s}')$ sampled from a replay buffer, $\mathcal{D}$, of data collected by a policy. We approximate the $\hat{V}$-function using a target Q-function $Q_{\bar{\phi}}$:

$$\hat{V}(\mathbf{s}', \mathbf{g}) \approx Q_{\bar{\phi}}(\mathbf{s}', \mathbf{a}', \mathbf{g}) - \log \pi(\mathbf{a}' \,|\, \mathbf{s}'; \mathbf{g}),$$

where $\mathbf{a}'$ are samples from the amortized variational policy $\pi_\theta(\cdot \,|\, \mathbf{s}'; \mathbf{g})$ The parameters $\bar{\phi}$ are updated to slowly track the parameters of ϕ at each time step via $\bar{\phi} \leftarrow \tau\bar{\phi} + (1-\tau)\phi$. We then train the policy to maximize the approximate Q-function and the entropy of the policy by performing gradient descent on

$$\mathcal{F}_\pi(\theta) = -\mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[Q_\phi(\mathbf{s}, \mathbf{a}, \mathbf{g}) - \log \pi_\theta(\mathbf{a} \,|\, \mathbf{s}; \mathbf{g})\right], \tag{20}$$

where $\mathbf{a}$ are samples from the amortized variational policy $\pi_\theta(\cdot \,|\, \mathbf{s}; \mathbf{g})$.

We estimate $\hat{q}_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ with a Monte Carlo estimate of Equation (14) obtained via a single Monte Carlo sample $(\mathbf{s}, \mathbf{a}, \mathbf{s}', \mathbf{a}', \mathbf{g})$ from the replay buffer. In practice, a value of $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) = 1$ can lead to numerical instabilities with bootstrapping, and so we also upper bound the estimated $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ by the prior distribution $p_{\Delta_{t+1}}(\Delta_{t+1} = 0)$.

To compute the rewards, we need to compute the likelihood of achieving the desired outcome. If the transition dynamics are unknown, we learn a dynamics model from environment interactions by training a neural network p_ψ that parameterizes the mean and scale of a factorized Laplace distribution. We train this model by minimizing the negative log-likelihood of the data collected by the policy,

$$\mathcal{F}_p(\psi) = -\mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}}[\log p_\psi(\mathbf{s}' \,|\, \mathbf{s}, \mathbf{a})], \tag{21}$$

and use it to compute the rewards

$$\hat{r}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \stackrel{\text{def}}{=} \hat{q}_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_\psi(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\Delta_t} \,\|\, p_{\Delta_t}). \tag{22}$$

Algorithm 1 ODAC: Outcome-Driven Actor–Critic

1: Initialize policy π_θ, replay buffer $\mathcal{R}$, Q-function Q_ϕ, and dynamics model p_ψ.
2: **for** iteration $i = 1, 2, \ldots$ **do**
3: Collect on-policy samples to add to $\mathcal{R}$ by sampling $\mathbf{g}$ from environment and executing π.
4: Sample batch $(\mathbf{s}, \mathbf{a}, \mathbf{s}', \mathbf{g})$ from $\mathcal{R}$.
5: Compute approximate reward and optimal weights with Equation (22) and Equation (14).
6: Update Q_ϕ with Equation (19), π_θ with Equation (20), and p_ψ with Equation (21).
7: **end for**

The complete algorithm is presented in Algorithm 1 and consists of alternating between collecting data via policy π and minimizing Equations 19, 20, and 21 via gradient descent. This algorithm alternates between approximating the lower bound in Equation (11) by repeatedly applying the outcome-driven Bellman operator on an approximate Q-function, and maximizing this lower bound by performing approximate policy optimization on Equation (20).

5. Related Work

Several prior works cast RL and control as probabilistic inference (Todorov, 2006; Ziebart et al., 2008; Hoffman et al., 2009; Rawlik et al., 2013; Levine, 2018; Fu et al., 2018; Singh et al., 2019; Fellows et al., 2019) or KL divergence minimization (Kárnỳ, 1996; Peters et al., 2010), but with the aim of reformulating standard reward-based RL, assuming that the reward function is given. In contrast, our work removes the need to manually specify a task-specific reward or likelihood function, and instead derives both an objective for learning an environment-specific likelihood function and a learning algorithm from the same inference problem.

Our problem definition is also related to goal-conditioned RL, where desired outcomes are often defined in terms of an exact goal-equality indicator (Kaelbling, 1993; Schaul et al., 2015). Unlike in our work, goal-conditioned RL typically requires specifying a reward function reflecting the desired outcome. The most natural choice for a reward function in

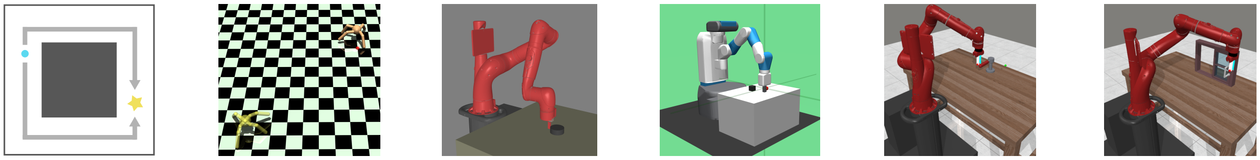


Figure 3: From left to right, we evaluate on: a 2D environment in which an agent must move around a box, a locomotion task in which a quadruped robot must match a location and pose (yellow), and four manipulation tasks in which the robot must push objects, rotate faucet valve, or open a window.

this setting is an exact goal–equality indicator, which gives non-zero reward whenever an outcome is reached. However, this type of reward function makes learning difficult, since the associated reward signal is sparse at best and impossible to attain at worst (in continuous state spaces).

To overcome this limitation, prior work has proposed heuristics for creating dense reward functions, such as the Euclidean distance or a thresholded distance to a goal (Andrychowicz et al., 2017; Levy et al., 2017; Pong et al., 2018; Nachum et al., 2018; Plappert et al., 2018; Schroecker and Isbell, 2020), or estimating auxiliary metrics to encourage learning, such as the mutual information or time between states and goal (Warde-Farley et al., 2019; Pong et al., 2019; Eysenbach et al., 2019; Venkattaramanujam et al., 2019; Hartikainen et al., 2020).

In contrast, a dense, generally-applicable reward function results automatically from our variational objective. In Section 6, we demonstrate that this reward function is substantially easier to optimize than sparse rewards, and that it removes the need to choose arbitrary thresholds or distance metrics needed in alternative approaches. Furthermore, the variational objective used to optimize the policy is a lower bound on the log-marginal likelihood of attaining the desired outcome under the environment dynamics and follows directly from the proposed variational approach for inferring a policy that leads to a desired outcome.

Our work extends closely related prior works that also frame control as inferring actions conditioned on reaching a goal or desired outcome (Attias, 2003; Toussaint and Storkey, 2006; Hoffman et al., 2009; Fu et al., 2018). Toussaint et al. (2006) and Hoffman et al. (2009) focus on exact inference methods that require time-varying tabular or time-varying Gaussian value functions. In contrast, we propose a variational inference method that eliminates the need to train a time-varying value function, and enables us to use expressive neural networks to represent an approximate value function, making our method applicable to high-dimensional, continuous, and non-linear domains. Unlike the approach in Attias (2003), our formulation is not constrained to fixed-horizon settings, obtains a closed-loop rather than open-loop policy, and is applicable to non-tabular dynamic models. More recently, Fu et al. (2018) proposed a probabilistic inference method for solving the unknown time-step formulation, but required on-policy trajectory samples. In contrast, we derive an off-policy method by introducing a variational distribution q_T over the time when the outcome is reached.

Lastly, the closely related problem of finding control laws that allow agents to move from an initial state to some desired goal state while incurring minimal cost has been studied in the stochastic control literature. Rawlik et al. (2010) consider a continuous-time setting and propose an expectation maximization algorithm that assumes linear Gaussian dynamics, while more recent work has explored finding control laws for non-linear stochastic system dynamics (Yi et al., 2020). In contrast to this strand of research, our framework considers a discrete-time setting and does not make assumptions of assumes knowledge of the system dynamics but only requires the ability to interact with the environment to learn an outcome-driven policy.

6. Empirical Evaluation

Our experiments compare ODAC to prior methods for learning goal-conditioned policies and evaluate how the different components of our variational objective impact the overall performance of the method. Specifically, we compare ODAC to prior methods on a wide range of manipulation and locomotion tasks that require achieving a desired outcome. To answer the second question, we conduct several ablation studies and visualize the behavior of $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$. In our experiments, we use a uniform action prior $p(\mathbf{a})$ and the time prior p_T is a geometric distribution with parameter 0.01, meaning that $p_{\Delta_{t+1}}(\Delta_{t+1} = 0) = 0.99$ for every t. We begin by describing prior methods and environments used for the experiments.

Baselines and prior work. We compare our method to hindsight experience replay (HER) (Andrychowicz et al., 2017), a goal-conditioned method, where the learner receives a reward of -1 if it is within an ϵ distance from the goal, and 0 otherwise, universal value density estimation (UVD) (Schroecker and Isbell, 2020), which also uses sparse rewards as well as a generative model of the future occupancy measure to estimate the Q-values, and DISCERN (Warde-Farley et al., 2019), which learns a reward function by training a discriminator and using a clipped log-likelihood as the reward. Lastly, include an oracle Soft Actor–Critic (SAC) baseline that uses a manually designed reward. For the MetaWorld tasks, this baseline uses the

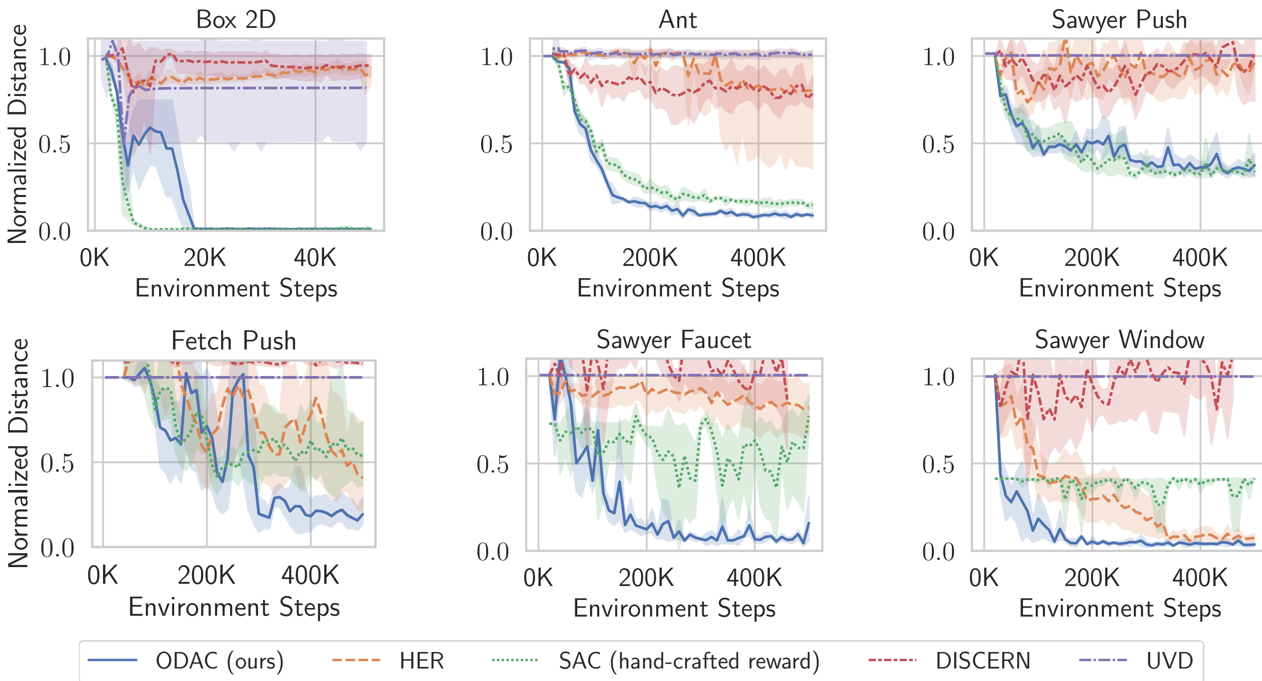


Figure 4: Learning curves showing final distance vs environment steps across all six environments. We see that only ODAC consistently performs well on all six tasks. Prior methods struggle to learn, especially in the absence of uniform goal sampling. See text for details.

benchmark reward for each task. For the remaining environments, this baseline uses the Euclidean distance between the agent's current and the desired outcome for the reward.

Environments. To avoid over-fitting to any one setting, we compare to these methods across several different robot morphologies and tasks, all illustrated in Figure 4. We compare ODAC to prior work on a simple 2D navigation task, in which an agent must take non-greedy actions to move around a box, as well as the *Ant*, *Sawyer Push*, and *Fetch Push* simulated robot domains, which have each been studied in prior work on reinforcement learning for reaching goals (Andrychowicz et al., 2017; Nair et al., 2018; Nachum et al., 2018; Pong et al., 2019; Schroecker and Isbell, 2020). For the Ant and Sawyer tasks, desired outcomes correspond to full states (i.e., desired positions and joints). For the Fetch task, we use the same goal representation as in prior work (Andrychowicz et al., 2017) and only represent $\mathbf{g}$ with the position of the object. Lastly, we demonstrate the feasibility of replacing manually designed rewards with our outcome-driven paradigm by evaluating the methods on the *Sawyer Window* and *Sawyer Faucet* tasks from the Meta-World benchmark (Yu et al., 2020). These tasks come with manually designed reward functions, which we replace by simply specify a desired outcome $\mathbf{g}$. Our plots show the mean and standard deviation of the normalized final Euclidean distance to the desired outcome across four random seeds. We normalize the distance to be 1 at the start of training. For further details, see Appendix D.1.

In all tasks, a fixed desired final goal is commanded as the exploration goal on each episode, and during training, the goals are relabeled using the future-style relabeling scheme from Andrychowicz et al. (2017). To challenge the methods, we choose the desired goal to be far from the starting state.

Results. In Figure 4, we see that ODAC outperforms virtually every method on all tasks, consistently learning faster and often reaching a final distance that is orders of magnitude closer to the desired outcome. The only exception is that the hand-crafted reward learns slightly faster on the 2D task, but this gap is closed within a few ten thousand steps.

6.1. Ablation Studies and Visualizations

Visualizing q_T. In Section 4, we presented a new Bellman operator, where the bootstrap value is multiplied by $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$. Since the bootstrap value is usually multiplied by a fixed discount factor γ, the state-dependent probability $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ effectively acts as a *dynamic discount factor* taking into account the agent's state relative to the desired outcome. In Figure 5, we investigate if $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ behaves as a sensible dynamic discount factor by plotting its value across an example trajectory in which the Ant robot flips on its back. We see that as the policy reaches an irrecoverable state, $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ drops in value, suggesting that ODAC *automatically* learns a discount factor that effectively terminates an episode when an irrecoverable state is reached.

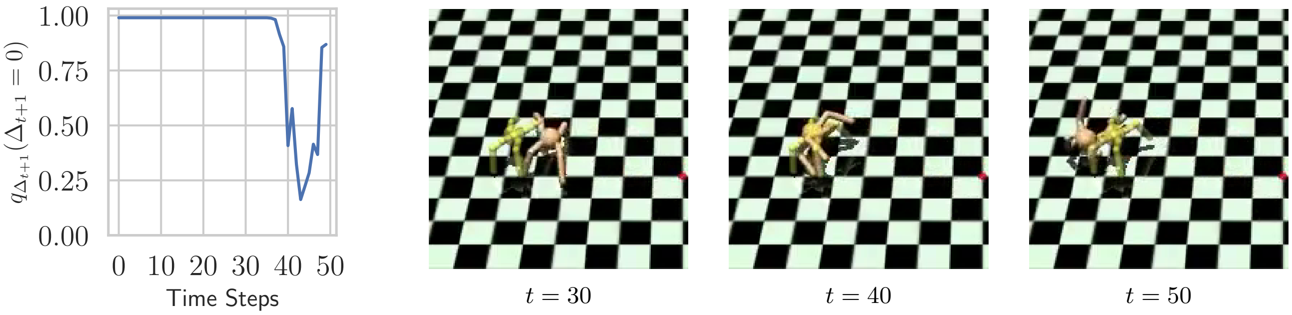


Figure 5: The inferred $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ versus time during an example trajectory in the Ant environment. As the ant robot falls over, $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ drops in value. We see that the optimal posterior $q^{\star}_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ given in Proposition 3 automatically assigns a high likelihood of terminating when this irrecoverable state is first reached, effectively acting as a dynamic discount factor.

Table 1: Ablation results, showing mean final normalized distance ($\times 100$) at the end of training across 4 seeds. Best mean is in bold and standard error in parentheses. ODAC is not sensitive to the dynamics models $\hat{p}_d$ but benefits from the dynamic q_T variant.

Env	ODAC	fix $\hat{p}_d$	fix q_T	fix q_T, $\hat{p}_d$
2D	1.7 (1.2)	1.2 (.14)	**1.0** (.24)	1.3 (.29)
Ant	**9.4** (.48)	11 (.57)	12 (.41)	13 (.20)
Push	35 (2.7)	**34** (1.5)	37 (1.5)	38 (3.1)
Fetch	19 (5.5)	**15** (2.5)	53 (13)	66 (15)
Window	5.4 (.62)	**5.0** (.62)	7.9 (.71)	6.0 (.12)
Faucet	**13** (4.2)	1 5 (3.3)	37 (8.3)	38 (7.2)

Ablation studies. Next, we study the importance of the dynamic discount factor $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ and the sensitivity of our method to the dynamics model. On all tasks, we evaluate the performance when the posterior exactly matches the prior, that is, $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) = 0.99$ (labeled **fixed** q_T in Table 1). Our analysis in Appendix A.3 suggests that this setting is sub-optimal, and this ablation will empirically evaluate its benefits. We also measure how the algorithm's performance depends on the accuracy of the learned dynamics model used for the reward in ODAC. To do this, we evaluate ODAC with the dynamics model fixed to a multivariate Laplace distribution with a fixed variance, centered at the previous state (labeled **fixed** $\hat{p}_d$ in Table 1). This ablation represents an extremely crude model, and good performance with such a model would indicate that our method does not depend on obtaining an particularly accurate model.

The results of both of these ablations are shown in Table 1. Across all six tasks, we see that ODAC with a learned and ODAC with a fixed dynamics model (second and third column from the left) generally perform the best, whereas fixing the variational distribution q_T generally leads to a deterioration in performance. These results suggest that the derived optimal variational distribution $q^{\star}_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ given in Proposition 3 is better not only in theory but also in practice, and that ODAC is not sensitive to the accuracy of the dynamics model.

6.2. Additional Experiments and Results

In Appendix C, we report the full learning curves for the ablation study and report the result of additional experiments. In particular, we find that prior methods perform much better when $\mathbf{g}$ is sampled uniformly from the set of possible desired outcomes. The large drop in performance when the desired outcome $\mathbf{g}$ is fixed may be due to the fact that uniformly sampling $\mathbf{g}$ implicitly provides a curriculum for learning. For example, in the Box 2D environment, goal states sampled above the box can train the agent to move around the obstacle, making it easier to learn how to reach the other side of the box. Without this guidance, prior methods often "got stuck" on the other side of the box. In contrast, ODAC consistently performs well in this more challenging setting, suggesting that the log-likelihood signal provides good guidance to the policy.

We also compare ODAC to a variant in which we use the learned dynamics model for model-based planning. In these experiments, the variant where the dynamics model is used only to compute rewards significantly outperformed the variant where it is used for both computing rewards and model-based planning. This result suggests that ODAC does not require learning a dynamics model that is accurate enough for planning, and that the derived Bellman updates are sufficient for obtaining policies that can achieve desired outcomes.

7. Conclusion

We proposed a probabilistic approach for achieving desired outcomes in settings where no reward function and no termination condition are given. We showed that by framing the problem of achieving desired outcomes as variational inference, we can derive an off-policy RL algorithm, a reward function learnable from environment interactions, and a novel Bellman backup that contains a state–action dependent dynamic discount factor for the reward and bootstrap term. Our experimental results further support that the resulting algorithm can lead to efficient outcome-driven approaches to RL.

Acknowledgements

We thank Marvin Zhang, Michael Janner, Abhishek Gupta, and various RAIL and OATML students for their discussions and feedback on early drafts of this paper. Tim G. J. Rudner is funded by the Rhodes Trust and the Engineering and Physical Sciences Research Council (EPSRC). This research was supported by Alan Turing Institute, the National Science Foundation, the DARPA Assured Autonomy Program, and ARL DCIST CRA W911NF-17-2-0181.

References

G.M Aly and W.C Chan. Numerical computation of optimal control problems with unknown final time. *Journal of Mathematical Analysis and Applications*, 45(2):274 – 284, 1974. ISSN 0022-247X. doi: https://doi.org/10.1016/0022-247X(74)90071-7. URL http://www.sciencedirect.com/science/article/pii/0022247X74900717.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob Mcgrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/pdf/1707.01495.pdf.

H. Attias. Planning by probabilistic inference. In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*, 2003.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. *arXiv preprint arXiv:1906.05253*, 2019. URL https://arxiv.org/pdf/1906.05253.pdf.

Matthew Fellows, Anuj Mahajan, Tim G. J. Rudner, and Shimon Whiteson. VIREL: A Variational Inference Framework for Reinforcement Learning. In *Advances in Neural Information Processing Systems 32*, 2019.

Justin Fu, Avi Singh, Dibya Ghosh, Larry Yang, and Sergey Levine. Variational inverse control with events: A general framework for data-driven reward definition. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Neural Information Processing Systems (NeurIPS)*, pages 8538–8547. 2018.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing Function Approximation Error in Actor-Critic Methods. *International Conference on Machine Learning (ICML)*, 2018.

Manfred Goebel and Uldis Raitums. Optimal control of two point boundary value problems. In H. J. Sebastian and K. Tammer, editors, *System Modelling and Optimization*, pages 281–290, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg. ISBN 978-3-540-47095-3.

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*, 2018a. URL https://arxiv.org/pdf/1801.01290.pdf.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications, 2018b.

Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical distance learning for unsupervised and semi-supervised skill discovery. In *International Conference on Learning Representations (ICLR)*, 2020.

Matthew Hoffman, Nando Freitas, Arnaud Doucet, and Jan Peters. An expectation maximization algorithm for continuous markov decision processes with arbitrary reward. In *Artificial intelligence and statistics*, pages 232–239, 2009.

Leslie P Kaelbling. Learning to achieve goals. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume vol.2, pages 1094 – 8, 1993.

H. J. Kappen, V. Gómez, and M. Opper. Optimal control as a graphical model inference problem. *Machine Learning*, 87(2):159–182, 2012.

Miroslav Kárnỳ. Towards fully probabilistic control design. *Automatica*, 32(12):1719–1722, 1996.

Sergey Levine. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. 2018. URL https://arxiv.org/pdf/1805.00909.pdf.

Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*, 2017.

Ofir Nachum, Google Brain, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-Efficient Hierarchical Reinforcement Learning. In *Neural Information Processing Systems (NeurIPS)*, 2018. URL https://sites.google.com/view/efficient-hrl.

Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. URL https://arxiv.org/pdf/1708.02596.pdf.

Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual Reinforcement Learning with Imagined Goals. In *Neural Information Processing Systems (NeurIPS)*, 2018. URL https://sites.google.com/site/visualrlwithimaginedgoals.

Jan Peters, Katharina Mülling, and Yasemin Altün. Relative Entropy Policy Search. In *AAAI Conference on Artificial Intelligence*, pages 1607–1612, 2010.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob Mcgrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-Goal Reinforcement Learning: Challenging Robotics Environments and Request for Research. *arXiv preprint arXiv:1802.09464*, 2018. URL http://fetchrobotics.com.

Vitchyr Pong, Shixiang Gu, Murtaza Dalal, and Sergey Levine. Temporal Difference Models: Model-Free Deep RL For Model-Based Control. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/pdf/1802.09081.pdf.

Vitchyr H. Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, abs/1903.03698, 2019. URL http://arxiv.org/pdf/1903.03698.pdf.

K. Rawlik, M. Toussaint, and S. Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Robotics: Science and Systems (RSS)*, 2013. URL http://roboticsproceedings.org/rss08/p45.pdf.

Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. An approximate inference approach to temporal optimization in optimal control. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/file/b5dc4e5d9b495d0196f61d45b26ef33e-Paper.pdf.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal Value Function Approximators. In *International Conference on Machine Learning (ICML)*, pages 1312–1320, 2015. ISBN 9781510810587. URL http://proceedings.mlr.press/v37/schaul15.pdf.

Yannick Schroecker and Charles Isbell. Universal value density estimation for imitation learning and goal-conditioned reinforcement learning. *arXiv preprint arXiv:2002.06473*, 2020.

Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. *arXiv preprint arXiv:1904.07854*, 2019.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. 1998. URL http://incompleteideas.net/sutton/book/bookdraft2016sep.pdf.

Csaba Szepesvári. *Algorithms for Reinforcement Learning*, volume 4. 2010. ISBN 9781608454921. doi: 10.2200/S00268ED1V01Y201005AIM009. URL https://sites.ualberta.ca/{~}szepesva/papers/RLAlgsInMDPs.pdf.

E. Todorov. Linearly-solvable Markov decision problems. In *Neural Information Processing Systems (NeurIPS)*, 2006. URL http://papers.nips.cc/paper/3002-linearly-solvable-markov-decision-problems.pdf.

M. Toussaint. Robot trajectory optimization using approximate inference. In *International Conference on Machine Learning (ICML)*, 2009. URL https://icml.cc/Conferences/2009/papers/271.pdf.

Marc Toussaint and Amos Storkey. Probabilistic inference for solving discrete and continuous state markov decision processes. In *Proceedings of the 23rd international conference on Machine learning*, pages 945–952, 2006.

Marc Toussaint, Stefan Harmeling, and Amos Storkey. Probabilistic inference for solving (po)mdps. Technical report, 2006.

Srinivas Venkattaramanujam, Eric Crawford, Thang Doan, and Doina Precup. Self-supervised learning of distance functions for goal-conditioned reinforcement learning. *arXiv:1907.02998*, 2019. URL https://arxiv.org/pdf/1907.02998.pdf.

David Warde-Farley, Tom Van De Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, & Volodymyr, and Mnih Deepmind. Unsupervised Control Through Non-Parametric Discriminative Rewards. In *International Conference on Learning Representations (ICLR)*, 2019. ISBN 1811.11359v1. URL https://arxiv.org/pdf/1811.11359.pdf.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Martha White. Unifying task specification in reinforcement learning. In *International Conference on Machine Learning*, pages 3742–3750. PMLR, 2017.

Z. Yi, Z. Cao, E. Theodorou, and Y. Chen. Nonlinear covariance control via differential dynamic programming. In *2020 American Control Conference (ACC)*, pages 3571–3576, 2020. doi: 10.23919/ACC45564.2020.9147531.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100. PMLR, 2020.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum Entropy Inverse Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, pages 1433–1438, 2008. ISBN 9781577353683 (ISBN). URL https://www.aaai.org/Papers/AAAI/2008/AAAI08-227.pdf.

Supplementary Material

A. Proofs & Derivations

A.1. Derivation of Variational Objectives

In this section, we present detailed derivations to complement the derivations and results included in Section 3.1 and Section 3.2.

Proposition 1 (Fixed-Time Outcome-Driven Variational Objective)**.** *Let* $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)$ *be as defined in Equation (4). Then, given any initial state* $\mathbf{s}_0$*, termination time* $t^\star$*, and outcome* $\mathbf{g}$*,*

$$D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0, \mathbf{S}_{t^\star} = \mathbf{g})) = \log p(\mathbf{g} \,|\, \mathbf{s}_0) - \bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}), \tag{A.1}$$

where

$$\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) \stackrel{\text{def}}{=} \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)}\left[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) + \sum_{t'=0}^{t-1} D_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,||\, p(\cdot \,|\, \mathbf{s}_{t'}))\right], \tag{A.2}$$

and since $\log p(\mathbf{g}|\mathbf{s}_0)$ *is constant in* π*,*

$$\arg\min_{\pi \in \Pi} D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0, \mathbf{S}_{t^\star} = \mathbf{g})) = \arg\max_{\pi \in \Pi} \bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}). \tag{A.3}$$

Proof. To find the (approximate) posterior $p_{\mathbf{a}_t}(\cdot \,|\, \mathbf{s}_t, \mathbf{S}_{t^\star} = \mathbf{g})$, we can use variational inference. To do so, we consider the trajectory distribution under $p_{\mathbf{a}_t}(\cdot \,|\, \mathbf{s}_t, \mathbf{S}_{t^\star} = \mathbf{g})$,

$$p(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, \mathbf{s}_0, \mathbf{S}_{t^\star} = \mathbf{g}) = p(\mathbf{a}_t \,|\, \mathbf{s}_t, \mathbf{s}_{t^\star} = \mathbf{g}) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_t, \mathbf{a}_t) p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}, \mathbf{S}_{t^\star} = \mathbf{g}), \tag{A.4}$$

where $t = t^\star - 1$ and we denote the state–action trajectory realization from state $\mathbf{s}_0$ and action $\mathbf{a}_0$ to time $t^\star - 1$ by $\tilde{\boldsymbol{\tau}}_{0:t} \stackrel{\text{def}}{=} \{\mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, ..., \mathbf{s}_t, \mathbf{a}_t\}$. Inferring a posterior distribution $p(\tilde{\boldsymbol{\tau}}_{0:t}|\mathbf{s}_{t^\star} = \mathbf{g})$ then becomes equivalent to finding a variational distribution $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)$, which induces a trajectory distribution $q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)$ that minimizes the KL divergence from $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)$ to $p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0, \mathbf{S}_{t^\star} = \mathbf{g})$:

$$\min_{q \in \bar{\mathcal{Q}}} D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0, \mathbf{S}_{t^\star} = \mathbf{g})). \tag{A.5}$$

If we find a distribution $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)$ for which the resulting KL divergence is zero, then $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)$ is the exact posterior. If the KL divergence is positive, then $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)$ is an approximate posterior. To solve the variational inference problem in Equation (A.5), we can define a factorized variational family as

$$q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0) \stackrel{\text{def}}{=} \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} q(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}), \tag{A.6}$$

where $\mathbf{a}_{0:t}$ are latent variables to be inferred, and the product is from $t = 0$ to $t = t - 1$ to exclude the conditional distribution over the (observed) state $\mathbf{S}_{t+1} = \mathbf{g}$ from the variational distribution.

Returning to the variational problem in Equation (A.5), we can now write

$$\begin{aligned} D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0, \mathbf{S}_{t^\star} = \mathbf{g})) &= \int_{\mathcal{A}^t} \int_{\mathcal{S}^t} q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, \mathbf{s}_0) \log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, \mathbf{s}_0)}{p(\tilde{\boldsymbol{\tau}}_{0:t}|\mathbf{s}_0, \mathbf{S}_{t^\star} = \mathbf{g})} d\mathbf{s}_{1:t} d\mathbf{a}_{0:t} \\ &= -\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) + \log p(\mathbf{g}|\mathbf{s}_0), \end{aligned} \tag{A.7}$$

where

$$\begin{aligned} \bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) \stackrel{\text{def}}{=} \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, \mathbf{s}_0)}\Bigg[& \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) + \log p(\mathbf{a}_t \,|\, \mathbf{s}_t) - \log \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \\ & + \sum_{t'=0}^{t-1} \log p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}) + \log p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) - \log \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}) - \log q(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \Bigg] \end{aligned} \tag{A.8}$$

and

$$\log p(\mathbf{g}|\mathbf{s}_0) = \log \int_{\mathcal{A}^t} \int_{\mathcal{S}^t} p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0, \mathbf{S}_{t^\star} = \mathbf{g}) d\mathbf{s}_{1:t} d\mathbf{a}_{0:t} \tag{A.9}$$

is a log-marginal likelihood. Following Haarnoja et al. (2018a), letting $q(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t) = p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)$ so that

$$q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, \mathbf{s}_0) \stackrel{\text{def}}{=} \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}), \tag{A.10}$$

we can simplify $\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g})$ to

$$\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) = \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)} \left[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) + \sum_{t'=0}^{t} D_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right]. \tag{A.11}$$

Since $\log p(\mathbf{g}|\mathbf{s}_0)$ is constant in π, solving the variational optimization problem in Equation (A.5) is equivalent to maximizing the variational objective with respect to $\pi \in \Pi$, where Π is a family of policy distributions. □

Corollary 1 (Fixed-Time Outcome-Driven Reward Function)**.** *The objective in Equation (5) corresponds to KL-regularized reinforcement learning with a time-varying reward function given by*

$$r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') \stackrel{\text{def}}{=} \mathbb{I}\{t' = t\} \log p_d(\mathbf{g} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}).$$

Proof. Let

$$r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') \stackrel{\text{def}}{=} \mathbb{I}\{t' = t\} \log p_d(\mathbf{g} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'}) \tag{A.12}$$

and note that the objective

$$\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) = \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)} \left[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) + \sum_{t=0}^{t} D_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t) \right] \tag{A.13}$$

can equivalently written as

$$\bar{\mathcal{F}}(\pi, \mathbf{s}_0, \mathbf{g}) = \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)} \left[\sum_{t=0}^{t} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') + \sum_{t'=0}^{t} D_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right] \tag{A.14}$$

$$= \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, \mathbf{s}_0)} \left[\sum_{t=0}^{t} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}, t') + D_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'})) \right], \tag{A.15}$$

which, as shown in Haarnoja et al. (2018a), can be written in the form of Equation (1). □

Proposition 2 (Unknown-time Outcome-Driven Variational Objective)**.** *Let $q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\tilde{\boldsymbol{\tau}}_{0:t}, t \,|\, \mathbf{s}_0) = q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0) q_T(t)$, let $q_T(t)$ be a variational distribution defined on $t \in \mathbb{N}_0$, and let $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)$ be as defined in Equation (4). Then, given any initial state $\mathbf{s}_0$ and outcome $\mathbf{g}$, we have that*

$$D_{\text{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0, \mathbf{S}_{T^\star} = \mathbf{g})) = \log p(\mathbf{g}|\mathbf{s}_0) - \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}), \tag{A.16}$$

where

$$\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0) t \, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \left[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - D_{\text{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0) \,||\, p_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0)) \right] \tag{A.17}$$

and $\log p(\mathbf{g} \,|\, \mathbf{s}_0)$ is constant in π and q_T.

Proof. In general, solving the variational problem

$$\min_{q \in \mathcal{Q}} D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0, \mathbf{S}_{T^\star} = \mathbf{g})) \tag{A.18}$$

from Section 3.2 in closed form is challenging, but as in the fixed-time setting, we can take advantage of the fact that, by choosing a variational family parameterized by

$$q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0) \stackrel{\text{def}}{=} \pi(\mathbf{a}_t \,|\, \mathbf{s}_t) \prod_{t'=0}^{t-1} p_d(\mathbf{s}_{t'+1} \,|\, \mathbf{s}_{t'}, \mathbf{a}_{t'})\, \pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'}), \tag{A.19}$$

with $\pi \in \Pi$, we can follow the same steps as in the proof for Proposition 1 and show that given any initial state $\mathbf{s}_0$ and outcome $\mathbf{g}$,

$$D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0, \mathbf{S}_{T^\star} = \mathbf{g})) = \log p(\mathbf{g} \,|\, \mathbf{s}_0) - \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}), \tag{A.20}$$

where

$$\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \Big[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0)) \Big], \tag{A.21}$$

where $q(\tilde{\boldsymbol{\tau}}_{0:t}, t|\mathbf{s}_0) \stackrel{\text{def}}{=} q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0) q_T(t)$, and hence, solving the variational problem in Equation (8) is equivalent to maximizing $\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})$ with respect to π and q_T. □

A.2. Derivation of Recursive Variational Objective

Proposition 4 (Factorized Unknown-Time Outcome-Driven Variational Objective)**.** *Let* $q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\tilde{\boldsymbol{\tau}}_{0:T}, T \,|\, \mathbf{s}_0) = q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0) q_T(t)$*, let* $q_T(t) = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0)$ *be a variational distribution defined on* $t \in \mathbb{N}_0$*, and let* $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)$ *be as defined in Equation (4). Then, given any initial state* $\mathbf{s}_0$ *and outcome* $\mathbf{g}$*, Equation (A.17) can be rewritten as*

$$\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) = \mathbb{E}_{q(\tilde{\boldsymbol{\tau}} \,|\, \mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) - D_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s}_t)) \Big) \right] \tag{A.22}$$

Proof. Consider the variational objective $\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})$ in Equation (A.17):

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\
&= \sum_{t=0}^{\infty} q_T(t)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \Big[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:T},T}(\cdot \,|\, \mathbf{s}_0)) \Big] && \text{(A.23)} \\
&= \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \left[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0) q_T(\cdot \,|\, \mathbf{s}_0)(t)}{p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0) p_T(\cdot \,|\, \mathbf{s}_0)} d\tilde{\boldsymbol{\tau}}_{0:t} \right] && \text{(A.24)} \\
&= \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \left[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)}{p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \right] - \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0) \log \frac{q_T(t \,|\, \mathbf{s}_0)}{q_T(t \,|\, \mathbf{s}_0)}. && \text{(A.25)}
\end{aligned}$$

Noting that $\sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0) \log \frac{q_T(t \,|\, \mathbf{s}_0)}{q_T(t \,|\, \mathbf{s}_0)} = D_{\mathrm{KL}}(q_T(\cdot \,|\, \mathbf{s}_0) \,\|\, p_T(\cdot \,|\, \mathbf{s}_0))$, we can write

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) && \text{(A.26)} \\
&= \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \left[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - \log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)}{p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \right] - D_{\mathrm{KL}}(q_T(\cdot \,|\, \mathbf{s}_0) \,\|\, p_T(\cdot \,|\, \mathbf{s}_0)) && \text{(A.27)} \\
&= \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \Big[\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) \Big] \\
&\qquad - \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \left[\log \frac{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)}{p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \right] - D_{\mathrm{KL}}(q_T(\cdot \,|\, \mathbf{s}_0) \,\|\, p_T(\cdot \,|\, \mathbf{s}_0)). && \text{(A.28)}
\end{aligned}$$

Further noting that for an infinite-horizon trajectory distribution $q(\tilde{\boldsymbol{\tau}}_{t'} \,|\, \mathbf{s}_{t'}) \stackrel{\text{def}}{=} \prod_{t=t'}^{\infty} p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)$, trajectory realization $\tilde{\boldsymbol{\tau}}_{t+1} \stackrel{\text{def}}{=} \{\boldsymbol{\tau}_{t'}\}_{t'=t+1}^{\infty}$, and any joint probability density $f(\mathbf{s}_t, \mathbf{a}_t)$,

$$\sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0)\, \mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \Big[f(\mathbf{s}_t, \mathbf{a}_t) \Big] = \sum_{t=0}^{\infty} \left(\mathbb{E}_{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} \Big[q_T(t \,|\, \mathbf{s}_0) f(\mathbf{s}_t, \mathbf{a}_t) \Big] \cdot \underbrace{\left(\int q(\tilde{\boldsymbol{\tau}}_{t+1} \,|\, \mathbf{s}_0) d\tilde{\boldsymbol{\tau}}_{t+1} \right)}_{=1} \right) \tag{A.29}$$

$$= \sum_{t=0}^{\infty} \left(\left(\int_{\mathcal{S}^t \times \mathcal{A}^t} q(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, \mathbf{s}_0) q_T(t \,|\, \mathbf{s}_0) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\boldsymbol{\tau}}_{0:t} \right) \cdot \underbrace{\left(\int q(\tilde{\boldsymbol{\tau}}_{t+1} \,|\, \mathbf{s}_0) d\tilde{\boldsymbol{\tau}}_{t+1} \right)}_{=1} \right) \tag{A.30}$$

$$= \sum_{t=0}^{\infty} \left(\int q(\tilde{\boldsymbol{\tau}}_{t+1} \,|\, \mathbf{s}_0) \left(\int_{\mathcal{S}^t \times \mathcal{A}^t} q(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, \mathbf{s}_0) q_T(t \,|\, \mathbf{s}_0) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\boldsymbol{\tau}}_{0:t} \right) d\tilde{\boldsymbol{\tau}}_{t+1} \right), \tag{A.31}$$

$$= \sum_{t=0}^{\infty} \int q(\tilde{\boldsymbol{\tau}}_0 \,|\, \mathbf{s}_0) q_T(t \,|\, \mathbf{s}_0) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\boldsymbol{\tau}}_0 \tag{A.32}$$

$$= \int q(\tilde{\boldsymbol{\tau}}_0 \,|\, \mathbf{s}_0) \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0) f(\mathbf{s}_t, \mathbf{a}_t) d\tilde{\boldsymbol{\tau}}_0, \tag{A.33}$$

we can express Equation (A.28) in terms of the infinite-horizon state–action trajectory $q(\tilde{\boldsymbol{\tau}}_0 \,|\, \mathbf{s}_0) \stackrel{\text{def}}{=} \prod_{t=0}^{\infty} p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)$ as

$$\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \tag{A.34}$$

$$= \int q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0) \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0) \log p(\mathbf{g}|\mathbf{s}_t, \mathbf{a}_t) d\tilde{\boldsymbol{\tau}} - \sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0) D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0)) - D_{\mathrm{KL}}(q_T(\cdot \,|\, \mathbf{s}_0) \parallel p_T(\cdot \,|\, \mathbf{s}_0)) \tag{A.35}$$

$$= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} q_T(t \,|\, \mathbf{s}_0) \Big(\log p(\mathbf{g}|\mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0)) \Big) \right] - D_{\mathrm{KL}}(q_T(\cdot \,|\, \mathbf{s}_0) \parallel p_T(\cdot \,|\, \mathbf{s}_0)). \tag{A.36}$$

Using Lemma 5 and the definition of $q_T(t \,|\, \mathbf{s}_0)$ in Equation (10), we can rewrite this objective as

$$\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})$$

$$= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \Big(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big) q_{\Delta_{t'}}(\Delta_{t'} = 1) \Big(\log p(\mathbf{g}|\mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0)) \Big) \right] - \sum_{t=0}^{\infty} \Big(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \Big) D_{\mathrm{KL}}(q_{\Delta_{t+1}} \,||\, p_{\Delta_{t+1}}) \tag{A.37}$$

$$= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q(\Delta_{t'} = 0) \right) \cdot \Big(q(\Delta_{t+1} = 1) \big(\log p(\mathbf{g}|\mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0) \parallel p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0)) \big) - D_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \Big) \right], \tag{A.38}$$

with

$$D_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}) = q_{\Delta_{t+1}}(\Delta_{t+1}=0)\log\frac{q_{\Delta_{t+1}}(\Delta_{t+1}=0)}{p_{\Delta_{t+1}}(\Delta_{t+1}=0)} + (1-q_{\Delta_{t+1}}(\Delta_{t+1}=0))\log\frac{1-q_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)}. \tag{A.39}$$

Next, to re-express $D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0))$ as a sum over Kullback-Leibler divergences between distributions over single action random variables, we note that

$$\begin{aligned}
D_{\mathrm{KL}}(q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\cdot \,|\, t, \mathbf{s}_0)) &= \int_{\mathcal{S}^t\times\mathcal{A}^t} q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)\log\frac{q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)}{p_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)} d\tilde{\boldsymbol{\tau}}_{0:t} && \text{(A.40)}\\
&= \int_{\mathcal{S}^t\times\mathcal{A}^t} q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)\log\frac{\prod_{t'=1}^{t}\pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})}{\prod_{t'=1}^{t}p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})} d\tilde{\boldsymbol{\tau}}_{0:t} && \text{(A.41)}\\
&= \int_{\mathcal{S}^t\times\mathcal{A}^t} q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \,|\, t, \mathbf{s}_0)\sum_{t'=0}^{t}\log\frac{\pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})}{p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})} d\tilde{\boldsymbol{\tau}}_{0:t} && \text{(A.42)}\\
&= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}|\mathbf{s}_0)}\left[\sum_{t'=0}^{t}\int_{\mathcal{A}}\pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})\log\frac{\pi(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})}{p(\mathbf{a}_{t'} \,|\, \mathbf{s}_{t'})} d\mathbf{a}_{t'}\right] && \text{(A.43)}\\
&= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}|\mathbf{s}_0)}\left[\sum_{t'=0}^{t} D_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'}))\right], && \text{(A.44)}
\end{aligned}$$

where we have used the same marginalization trick as above to express the expression in terms of an infinite-horizon trajectory distribution, which allows us to express Equation (A.38) as

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})\\
&= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)}\Bigg[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t} q(\Delta_{t'}=0)\right)\\
&\qquad \cdot\left(q(\Delta_{t+1}=1)\left(\log p(\mathbf{g}|\mathbf{s}_t,\mathbf{a}_t) - \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}|\mathbf{s}_0)}\left[\sum_{t'=0}^{t} D_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'}))\right]\right) - D_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}})\right)\Bigg].
\end{aligned} \tag{A.45}$$

Rearranging and dropping redundant expectation operators, we can now express the objective as

$$\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g})\\
&= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)}\Bigg[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t} q(\Delta_{t'}=0)\right)\\
&\qquad \cdot\left(q(\Delta_{t+1}=1)\left(\log p(\mathbf{g}|\mathbf{s}_t,\mathbf{a}_t) - \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}|\mathbf{s}_0)}\left[\sum_{t'=0}^{t} D_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'}))\right]\right) - D_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}})\right)\Bigg].
\end{aligned} \tag{A.46}$$

$$\begin{aligned}
&= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)}\left[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t} q(\Delta_{t'}=0)q(\Delta_{t+1}=1)\right)\log p(\mathbf{g}|\mathbf{s}_t,\mathbf{a}_t) - D_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}})\right]\\
&\qquad - \sum_{t=0}^{\infty}\underbrace{\left(\prod_{t'=1}^{t} q(\Delta_{t'}=0)q(\Delta_{t+1}=1)\right)}_{=q_T(t \,|\, \mathbf{s}_0)}\mathbb{E}_{q(\tilde{\boldsymbol{\tau}}|\mathbf{s}_0)}\left[\sum_{t'=0}^{t} D_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t'}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t'}))\right],
\end{aligned} \tag{A.47}$$

whereupon we note that the negative term can be expressed as

$$\sum_{t=0}^{\infty} q_T(t \mid \mathbf{s}_0)\, \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}|\mathbf{s}_0)} \left[\sum_{t'=0}^{t} D_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_{t'}) \parallel p(\cdot \mid \mathbf{s}_{t'})) \right] = \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}|\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \sum_{t'=0}^{t} q_T(t \mid \mathbf{s}_0) D_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_{t'}) \parallel p(\cdot \mid \mathbf{s}_{t'})) \right] \tag{A.48}$$

$$= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}|\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} q(T \geq t) D_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_t) \parallel p(\cdot \mid \mathbf{s}_t)) \right] \tag{A.49}$$

$$= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}|\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \underbrace{\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right)}_{\text{(by Lemma 2)}} D_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_t) \parallel p(\cdot \mid \mathbf{s}_t)) \right], \tag{A.50}$$

where the second line follows from expanding the sums and regrouping terms. By substituting the expression in Equation (A.50) into Equation (A.47), we obtain an objective expressed entirely in terms of distributions over single-index random variables:

$$\begin{aligned} &\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\ &= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)} \Bigg[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \\ &\qquad \cdot \left(q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}) \right) - D_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_t) \parallel p(\cdot \mid \mathbf{s}_t)) \Bigg] \end{aligned} \tag{A.51}$$

$$= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)} \left[\sum_{t=0}^{\infty} \left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \left(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) - D_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_t) \parallel p(\cdot \mid \mathbf{s}_t)) \right) \right], \tag{A.52}$$

where we defined

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \stackrel{\text{def}}{=} q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\Delta_{t+1}} \parallel p_{\Delta_{t+1}}), \tag{A.53}$$

which concludes the proof. □

Theorem 1 (Outcome-Driven Variational Inference). *Let $q_T(t)$ and $q_{\tilde{\boldsymbol{\tau}}_{0:t}}(\tilde{\boldsymbol{\tau}}_{0:t} \mid t, \mathbf{s}_0)$ be as defined in Equation (4) and Equation (10), and define*

$$V^\pi(\mathbf{s}_t, \mathbf{g}; q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t \mid \mathbf{s}_t)}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T)] - D_{\mathrm{KL}}(\pi(\cdot \mid \mathbf{s}_t) \,\|\, p(\cdot \mid \mathbf{s}_t)), \tag{A.54}$$

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \stackrel{\text{def}}{=} r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) + q(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)}\left[V^\pi(\mathbf{s}_{t+1}, \mathbf{g}; \pi, q_T)\right], \tag{A.55}$$

$$r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) \stackrel{\text{def}}{=} q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}). \tag{A.56}$$

Then given any initial state $\mathbf{s}_0$ and outcome $\mathbf{g}$,

$$D_{\mathrm{KL}}(q_{q_{\tilde{\boldsymbol{\tau}}_{0:t}},T}(\cdot \mid \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t},T}(\cdot \mid \mathbf{s}_0, \mathbf{S}_{T^\star} = \mathbf{g})) = -\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) + C = -V^\pi(\mathbf{s}_0, \mathbf{g}; q_T) + C,$$

where $C \stackrel{\text{def}}{=} \log p(\mathbf{g} \mid \mathbf{s}_0)$ is independent of π and q_T, and hence maximizing $V^\pi(\mathbf{s}_0, \mathbf{g}; \pi, q_T)$ is equivalent to minimizing Equation (8). In other words,

$$\operatorname*{arg\,min}_{\pi \in \Pi, q_T \in \mathcal{Q}_T} \{D_{\mathrm{KL}}(q_{q_{\tilde{\boldsymbol{\tau}}_{0:t}},T}(\cdot \mid \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t},T}(\cdot \mid \mathbf{s}_0, \mathbf{S}_{T^\star} = \mathbf{g}))\} = \operatorname*{arg\,max}_{\pi \in \Pi, q_T \in \mathcal{Q}_T} \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) = \operatorname*{arg\,max}_{\pi \in \Pi, q_T \in \mathcal{Q}_T} V^\pi(\mathbf{s}_0, \mathbf{g}; q_T).$$

Proof. Consider the objective derived in Proposition 4 (Factorized Unknown-Time Outcome-Driven Variational Objective),

$$
\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\
&\quad= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)}\Bigg[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\right) \\
&\qquad\cdot \underbrace{\left(q_{\Delta_{t+1}}(\Delta_{t+1}=1)\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t) - D_{\text{KL}}(q_{\Delta_{t+1}}\,\|\,p_{\Delta_{t+1}})\right)}_{\stackrel{\text{def}}{=} r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta)} - D_{\text{KL}}(\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)\,\|\,p(\mathbf{a}_t|\mathbf{s}_t))\Bigg],
\end{aligned}
\tag{A.57}
$$

and recall that, by Proposition 2 (Unknown-time Outcome-Driven Variational Objective),

$$
D_{\text{KL}}(q_{q\tilde{\boldsymbol{\tau}}_{0:t},T}(\cdot\,|\,\mathbf{s}_0)\,\|\,p_{\tilde{\boldsymbol{\tau}}_{0:t},T}(\cdot\,|\,\mathbf{s}_0,\mathbf{S}_{T^\star}=\mathbf{g})) = -\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g}) + \log p(\mathbf{g}|\mathbf{s}_0). \tag{A.58}
$$

Therefore, to prove the result that

$$
D_{\text{KL}}(q_{q\tilde{\boldsymbol{\tau}}_{0:t},T}(\cdot\,|\,\mathbf{s}_0)\,\|\,p_{\tilde{\boldsymbol{\tau}}_{0:t},T}(\cdot\,|\,\mathbf{s}_0,\mathbf{S}_{T^\star}=\mathbf{g})) = -V^\pi(\mathbf{s}_0,\mathbf{g};q_T) + \log p(\mathbf{g}|\mathbf{s}_0),
$$

we just need to show that $\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g}) = V^\pi(\mathbf{s}_0,\mathbf{g};q_T)$ for $V^\pi(\mathbf{s}_0,\mathbf{g};q_T)$ as defined in the theorem. To do so, we start from the objective $\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g})$ and and unroll it for $t=0$:

$$
\begin{aligned}
&\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) \\
&\quad= \mathbb{E}_{q(\tilde{\boldsymbol{\tau}}_0|\mathbf{s}_0)}\left[\sum_{t=0}^{\infty}\left(\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\right) r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) - D_{\text{KL}}(\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)\,\|\,p(\mathbf{a}_t|\mathbf{s}_t))\right] \qquad (\text{A.59})\\
&\quad= \mathbb{E}_{\pi(\mathbf{a}_0\,|\,\mathbf{s}_0)}\Bigg[r(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_\Delta) + \mathbb{E}_{q(\boldsymbol{\tau}_1\,|\,\mathbf{s}_0,\mathbf{a}_0)}\left[\sum_{t=1}^{\infty}\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\Big(r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t))\Big)\right]\Bigg] \\
&\qquad\quad - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_0)\,\|\,p(\cdot\,|\,\mathbf{s}_0)).
\end{aligned}
\tag{A.60}
$$

With this expression at hand, we now define

$$
Q^\pi_{\text{sum}}(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_T) \stackrel{\text{def}}{=} r(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_\Delta) + \mathbb{E}_{q(\boldsymbol{\tau}|\mathbf{s}_0,\mathbf{a}_0)}\left[\sum_{t=1}^{\infty}\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\Big(r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t))\Big)\right],
\tag{A.61}
$$

and note that $\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g}) = \mathbb{E}_{\pi(\mathbf{a}_0\,|\,\mathbf{s}_0)}[Q^\pi_{\text{sum}}(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_T)] - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_0)\,\|\,p(\cdot\,|\,\mathbf{s}_0)) = V^\pi(\mathbf{s}_0,\mathbf{g};q_T)$, as per the definition of $V^\pi(\mathbf{s}_0,\mathbf{g};q_T)$. To prove the theorem from this intermediate result, we now have to show that $Q^\pi_{\text{sum}}(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_T)$ as defined in Equation (A.61) can in fact be expressed recursively as $Q^\pi_{\text{sum}}(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_T) = Q^\pi(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_T)$ with

$$
Q^\pi(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_T) = r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) + q(\Delta_{t+1}=0)\,\mathbb{E}_{p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1},\mathbf{g};\pi,q_T)]\,. \tag{A.62}
$$

To see that this is the case, first, unroll $Q^\pi(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_T)$ for $t=1$,

$$
\begin{aligned}
&Q^\pi_{\text{sum}}(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_T) \\
&\quad= r(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_\Delta) + \mathbb{E}_{q(\boldsymbol{\tau}_1|\mathbf{s}_0,\mathbf{a}_0)}\left[\sum_{t=1}^{\infty}\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\Big(r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t))\Big)\right] \qquad (\text{A.63})\\
&\quad= r(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_\Delta) + \mathbb{E}_{p_d(\mathbf{s}_1|\mathbf{a}_0,\mathbf{a}_0)}\left[\mathbb{E}_{q(\boldsymbol{\tau}_1|\mathbf{s}_0,\mathbf{a}_0)}\left[\sum_{t=1}^{\infty}\prod_{t'=1}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\Big(r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t))\Big)\right]\right] \qquad (\text{A.64})\\
&\quad= r(\mathbf{s}_0,\mathbf{a}_0,\mathbf{g};q_\Delta) + \mathbb{E}_{p_d(\mathbf{s}_1|\mathbf{a}_0,\mathbf{a}_0)}\Bigg[\mathbb{E}_{\pi(\mathbf{a}_1\,|\,\mathbf{s}_1)}\Bigg[q_{\Delta_1}(\Delta_1=0)\left(r(\mathbf{s}_1,\mathbf{a}_1,\mathbf{g};q_\Delta) - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_1)\,\|\,p(\cdot\,|\,\mathbf{s}_1))\right) \\
&\qquad\quad + \mathbb{E}_{q(\boldsymbol{\tau}_2|\mathbf{s}_1,\mathbf{a}_1)}\left[\sum_{t=2}^{\infty}\prod_{t'=2}^{t} q_{\Delta_{t'}}(\Delta_{t'}=0)\Big(r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t))\Big)\right]\Bigg]\Bigg],
\end{aligned}
\tag{A.65}
$$

and note that we can rearrange this expression to obtain the recursive relationship

$$\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T) \\
&= r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + q_{\Delta_1}(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{0+1} \,|\, \mathbf{s}_0, \mathbf{a}_0)} \Bigg[- D_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_1) \,\|\, p(\cdot \,|\, \mathbf{s}_1)) \\
&\quad + \mathbb{E}_{\pi(\mathbf{a}_1 \,|\, \mathbf{s}_1)} \Bigg[\underbrace{r(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; q_\Delta) + \mathbb{E}\left[\sum_{t=2}^{\infty} \left(\prod_{t'=2}^{t} q_{\Delta_{t'}}(\Delta_{t'} = 0) \right) \Big(r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta) - D_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_t) \,\|\, p(\cdot \,|\, \mathbf{s})) \Big) \right]}_{= Q^{\pi}_{\text{sum}}(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; q_T)} \Bigg] \Bigg],
\end{aligned} \tag{A.66}$$

where the innermost expectation is taken with respect to $q(\boldsymbol{\tau}_2 | \mathbf{s}_1, \mathbf{a}_1)$. With this result, we see that

$$\begin{aligned}
&Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T) \\
&= r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + q_{\Delta_1}(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{0+1} \,|\, \mathbf{s}_0, \mathbf{a}_0)} \Big[\mathbb{E}_{\pi(\mathbf{a}_1 | \mathbf{s}_1)} \left[Q^{\pi}_{\text{sum}}(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; q_T) \right] - D_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_1) \,\|\, p(\cdot \,|\, \mathbf{s}_1)) \Big] && \text{(A.67)} \\
&= r(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_\Delta) + q_{\Delta_1}(\Delta_1 = 0)\, \mathbb{E}_{p_d(\mathbf{s}_1 | \mathbf{s}, \mathbf{a})} \Big[V^{\pi}(\mathbf{s}_1, \mathbf{g}; q_T) \Big], && \text{(A.68)}
\end{aligned}$$

for $V(\mathbf{s}_{t+1}, \mathbf{g}; q_T)$ as defined above, as desired. In other words, we have that

$$\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) = \mathbb{E}_{\pi(\mathbf{a}_0 \,|\, \mathbf{s}_0)}[Q^{\pi}_{\text{sum}}(\mathbf{s}_0, \mathbf{a}_0, \mathbf{g}; q_T)] - D_{\text{KL}}(\pi(\cdot \,|\, \mathbf{s}_0) \,\|\, p(\cdot \,|\, \mathbf{s}_0)) = V^{\pi}(\mathbf{s}_0, \mathbf{g}; q_T). \tag{A.69}$$

Combining this result with Proposition 2 (Unknown-time Outcome-Driven Variational Objective) and Proposition 4 (Factorized Unknown-Time Outcome-Driven Variational Objective), we finally conclude that

$$D_{\text{KL}}(q_{q\tilde{\boldsymbol{\tau}}_{0:t}, T}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t}, T}(\cdot \,|\, \mathbf{s}_0, \mathbf{S}_{T^\star} = \mathbf{g})) = -\mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) + C = -V^{\pi}(\mathbf{s}_0, \mathbf{g}; q_T) + C, \tag{A.70}$$

where $C \stackrel{\text{def}}{=} \log p(\mathbf{g} \,|\, \mathbf{s}_0)$ is independent of π and q_T. Hence, maximizing $V^{\pi}(\mathbf{s}_0, \mathbf{g}; \pi, q_T)$ is equivalent to minimizing the objective in Equation (8). In other words,

$$\underset{\pi \in \Pi, q_T \in \mathcal{Q}_T}{\arg\min} \left\{ D_{\text{KL}}(q_{q\tilde{\boldsymbol{\tau}}_{0:t}, T}(\cdot \,|\, \mathbf{s}_0) \,\|\, p_{\tilde{\boldsymbol{\tau}}_{0:t}, T}(\cdot \,|\, \mathbf{s}_0, \mathbf{S}_{T^\star} = \mathbf{g})) \right\} = \underset{\pi \in \Pi, q_T \in \mathcal{Q}_T}{\arg\max}\ \mathcal{F}(\pi, q_T, \mathbf{s}_0, \mathbf{g}) = \underset{\pi \in \Pi, q_T \in \mathcal{Q}_T}{\arg\max}\ V^{\pi}(\mathbf{s}_0, \mathbf{g}; q_T). \tag{A.71}$$

This concludes the proof. □

Corollary 3 (Simplified Outcome-Driven Variational Inference)**.** *Let $q_T = p_T$, assume that p_T is a Geometric distribution with parameter $\gamma \in (0, 1)$, and let $p(\mathbf{a}_t | \mathbf{s}_t)$ be a uniform distribution. Then the inference problem in Equation (8) of finding a goal-directed variational trajectory distribution simplifies to maximizing the following recursively defined variational objective with respect to π:*

$$\bar{V}^{\pi}(\mathbf{s}_0, \mathbf{g}; \gamma) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_1 \,|\, \mathbf{s}_1)} \left[Q(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; \gamma) \right] - \mathcal{H}(\pi(\cdot \,|\, \mathbf{s}_0)), \tag{A.72}$$

where

$$\bar{Q}^{\pi}(\mathbf{s}_1, \mathbf{a}_1, \mathbf{g}; \gamma) \stackrel{\text{def}}{=} (1 - \gamma) \log p_d(\mathbf{g} \,|\, \mathbf{s}_1, \mathbf{a}_1) + \gamma\, \mathbb{E}_{p_d(\mathbf{s}_{0+1} \,|\, \mathbf{s}_0, \mathbf{a}_0)} \left[V(\mathbf{s}_2, \mathbf{g}; \gamma) \right] \tag{A.73}$$

and $\mathcal{H}(\cdot)$ is the entropy functional.

Proof. The result follows immediately when replacing q_Δ in Theorem 1 by p_Δ and noting that $D_{\text{KL}}(p_\Delta \,\|\, p_\Delta) = 0$. □

A.3. Derivation of Optimal Variational Posterior over T

Proposition 3 (Optimal Variational Distribution over T)**.** *The optimal variational distribution $q_T^\star$ with respect to Equation (11) is defined recursively in terms of $q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0)\forall t \in \mathbb{N}_0$ by*

$$
\begin{aligned}
&q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0;\pi,Q^\pi) \\
&\quad = \sigma\Big(\mathbb{E}_{\pi(\mathbf{a}_{t+1}\,|\,\mathbf{s}_{t+1})p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}[Q^\pi(\mathbf{s}_{t+1},\mathbf{a}_{t+1},\mathbf{g};q_T)] - \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t)] + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1}=0)\right)\Big)
\end{aligned}
$$

where $\sigma(\cdot)$ is the sigmoid function, that is, $\sigma(x)=\frac{1}{e^{-x}+1}$ and $\sigma^{-1}(x)=\log\frac{x}{1-x}$.

Proof. Consider $\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g})$:

$$
\mathcal{F}(\pi,q_T,\mathbf{s}_t,\mathbf{g}) = \mathbb{E}_{\pi(\mathbf{a}_t|\mathbf{s}_t)}[Q^\pi(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_T)] = \mathbb{E}_{\pi(\mathbf{a}_t|\mathbf{s}_t)}[r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1}=0)\,\mathbb{E}\left[V(\mathbf{s}_{t+1},\mathbf{g};q_T)\right]]. \tag{A.74}
$$

Since the variational objective $\mathcal{F}(\pi,q_T,\mathbf{s}_t,\mathbf{g})$ can be expressed recursively as

$$
V^\pi(\mathbf{s}_t,\mathbf{g};q_T) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}[Q(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_T)] - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t)),
$$

with

$$
\begin{aligned}
Q^\pi(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_T) &= r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1}=0)\,\mathbb{E}_{p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}\left[V^\pi(\mathbf{s}_{t+1},\mathbf{g};q_T)\right], \\
r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) &= q_{\Delta_{t+1}}(\Delta_{t+1}=1)\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t) - D_{\text{KL}}(q_{\Delta_{t+1}}\,\|\,p_{\Delta_{t+1}}),
\end{aligned}
$$

and since $D_{\text{KL}}(q_{\Delta_{t+1}}\,\|\,p_{\Delta_{t+1}})$ is strictly convex in $q_{\Delta_{t+1}}(\Delta_{t+1}=0)$, we can find the globally optimal Bernoulli distribution parameters $q_{\Delta_{t+1}}(\Delta_{t+1}=0)$ for all $t\in\mathbb{N}_0$ recursively. That is, it is sufficient to solve the problem

$$
q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0) \stackrel{\text{def}}{=} \operatorname*{arg\,max}_{q_{\Delta_{t+1}}(\Delta_{t+1}=0)} \{\mathcal{F}(\pi,q_T,\mathbf{s}_0,\mathbf{g})\} = \operatorname*{arg\,max}_{q_{\Delta_{t+1}}(\Delta_{t+1}=0)} \left\{\mathcal{F}(\pi,q_{\Delta_1},\dots,q_{\Delta_{t+1}},\dots,\mathbf{s}_0,\mathbf{g})\right\} \tag{A.75}
$$

for a fixed $t+1$. To do so, we take the derivative of $\mathcal{F}(\pi,q_{\Delta_1},\dots,q_{\Delta_{t+1}},\dots,\mathbf{s}_0,\mathbf{g})$, which—defined recursively—is given by

$$
\begin{aligned}
&\mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}[Q(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_T] - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t)) && \text{(A.76)} \\
&= \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}\left[r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{g};q_\Delta) + q_{\Delta_{t+1}}(\Delta_{t+1}=0)\,\mathbb{E}_{p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1},\mathbf{g};q_T)]\right] - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t)) && \text{(A.77)} \\
&= \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}\Bigg[q_{\Delta_{t+1}}(\Delta_{t+1}=1)\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t) - D_{\text{KL}}(q_{\Delta_{t+1}}\,\|\,p_{\Delta_{t+1}}) && \text{(A.78)} \\
&\qquad + q_{\Delta_{t+1}}(\Delta_{t+1}=0)\,\mathbb{E}_{p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1},\mathbf{g};q_T)]\Bigg] - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t)) && \text{(A.79)} \\
&= \mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}\Bigg[(1-q_{\Delta_{t+1}}(\Delta_{t+1}=0))\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t) - D_{\text{KL}}(q_{\Delta_{t+1}}\,\|\,p_{\Delta_{t+1}}) && \text{(A.80)} \\
&\qquad + q_{\Delta_{t+1}}(\Delta_{t+1}=0)\,\mathbb{E}_{p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}[V^\pi(\mathbf{s}_{t+1},\mathbf{g};q_T)]\Bigg] - D_{\text{KL}}(\pi(\cdot\,|\,\mathbf{s}_t)\,\|\,p(\cdot\,|\,\mathbf{s}_t)), && \text{(A.81)}
\end{aligned}
$$

with respect to $q_{\Delta_{t+1}}(\Delta_{t+1}=0)$ and set it to zero, which yields

$$
\begin{aligned}
0 = &-\mathbb{E}_{\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}\left[\log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t) + \mathbb{E}_{\pi(\mathbf{a}_{t+1}\,|\,\mathbf{s}_{t+1})p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)}[Q^\pi(\mathbf{s}_{t+1},\mathbf{a}_{t+1},\mathbf{g};q_T)]\right] \\
&+ \log\frac{1-q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)} - \log\frac{q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0)}{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}.
\end{aligned} \tag{A.82}
$$

Rearranging, we get

$$
\begin{aligned}
&\frac{q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-q^\star_{\Delta_{t+1}}(\Delta_{t+1}=0)} \\
&\quad = \exp\left(\mathbb{E}_{\pi(\mathbf{a}_{t+1}\,|\,\mathbf{s}_{t+1})p_d(\mathbf{s}_{t+1}\,|\,\mathbf{s}_t,\mathbf{a}_t)\pi(\mathbf{a}_t\,|\,\mathbf{s}_t)}[Q^\pi(\mathbf{s}_{t+1},\mathbf{a}_{t+1},\mathbf{g};q_T) - \log p_d(\mathbf{g}\,|\,\mathbf{s}_t,\mathbf{a}_t)] + \log\frac{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)}\right),
\end{aligned} \tag{A.83}
$$

where the Q-function depends on $q(\Delta_{t'})$ with $t' > t$, but not on $q^{\star}_{\Delta_{t+1}}(\Delta_{t+1} = 0)$. Solving for $q^{\star}_{\Delta_{t+1}}(\Delta_{t+1} = 0)$, we obtain

$$
\begin{aligned}
&q^{\star}_{\Delta_{t+1}}(\Delta_{t+1} = 0) && \text{(A.84)}\\
&= \frac{\exp(\mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\pi(\mathbf{a}_t \mid \mathbf{s}_t)}[Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T) - \log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t)] + \log \frac{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)})}{1 + \exp(\mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\pi(\mathbf{a}_t \mid \mathbf{s}_t)}[Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T) - \log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t)] + \log \frac{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)})} && \text{(A.85)}\\
&= \sigma\Big(\mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)}[Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T)] - \mathbb{E}_{\pi(\mathbf{a}_t \mid \mathbf{s}_t)}[\log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t)] + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1} = 0)\right) \Big), && \text{(A.86)}
\end{aligned}
$$

where $\sigma(\cdot)$ is the sigmoid function with $\sigma(x) = \frac{1}{e^{-x}+1}$ and $\sigma^{-1}(x) = \log \frac{x}{1-x}$. This concludes the proof. □

Remark 1. *As can be seen from Proposition 3 (Optimal Variational Distribution over T), the optimal approximation to the posterior over T trades off short-term rewards via $\mathbb{E}_{\pi(\mathbf{a}_t \mid \mathbf{s}_t)}[r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_{\Delta})]$, long-term rewards via $\mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)}[Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T)]$, and the prior log-odds of not achieving the outcome at a given point in time conditioned on the outcome not having been achieved yet, $\frac{p_{\Delta_{t+1}}(\Delta_{t+1}=0)}{1-p_{\Delta_{t+1}}(\Delta_{t+1}=0)}$.*

Corollary 2 (Equivalence to Soft Actor-Critic (Haarnoja et al., 2018a))**.** *Let $q_T = p_T$, assume that p_T is a Geometric distribution with parameter $\gamma \in (0, 1)$, and let $p(\mathbf{a}_t \mid \mathbf{s}_t)$ be a uniform distribution. If the model of the state transition distribution is a Gibbs distribution,*

$$
\tilde{p}_d(\mathbf{g}|\mathbf{s}_t, \mathbf{a}_t; \gamma) \stackrel{\text{def}}{=} \frac{\exp\left(-\beta E(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g})\right)}{Z(\mathbf{s}_t, \mathbf{a}_t; \beta)}, \tag{A.87}
$$

with $\beta \stackrel{\text{def}}{=} \log(1-\gamma)$, $Z(\mathbf{s}_t, \mathbf{a}_t; \beta) \stackrel{\text{def}}{=} \int_{\mathcal{S}} \exp\left(-\beta E(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}')\right) d\mathbf{g}' < \infty$ and $E : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$, then for a fixed $\mathbf{g}$ and a reward function

$$
\tilde{r}(\mathbf{s}_t, \mathbf{a}_t) \stackrel{\text{def}}{=} E(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}) + \log Z(\mathbf{s}_t, \mathbf{a}_t; \beta), \tag{A.88}
$$

the inference problem in Equation (8) of finding a goal-directed variational trajectory distribution simplifies to the infinite-horizon, discounted Soft Actor-Critic objective (Haarnoja et al., 2018a), with

$$
Q^{\pi}(\mathbf{s}, \mathbf{a}) \stackrel{\text{def}}{=} \tilde{r}(\mathbf{s}, \mathbf{a}) + \gamma\, \mathbb{E}\left[V^{\pi}(\mathbf{s}_1)\right], \tag{A.89}
$$

where the expectation is w.r.t. the true state transition distribution $p_d(\mathbf{s}_1|\mathbf{s}, \mathbf{a})$, and

$$
V^{\pi}(\mathbf{s}_1) \stackrel{\text{def}}{=} \mathbb{E}_{\pi(\mathbf{a}_1 \mid \mathbf{s}_1)}\left[Q^{\pi}(\mathbf{s}_1, \mathbf{a}_1)\right] + \mathcal{H}(\pi(\mathbf{a}_1 \mid \mathbf{s}_1)), \tag{A.90}
$$

where $\mathcal{H}(\cdot)$ is the entropy functional.

Proof. The result follows immediately from the definition of Q^{π} in Theorem 1. □

Remark 2. *Corollary 2 shows that the infinite-horizon, discounted Soft Actor-Critic algorithm can be derived entirely from first principles. In contrast, Haarnoja et al. (2018a) do not derive the discounted infinite-horizon objective from first principles, but instead include a discount factor post-hoc. Corollary 2 provides a probabilistic justification for this post-hoc objective.*

A.4. Lemmas

Lemma 1. *Let $q(T = t) \stackrel{\text{def}}{=} q(T = t|T \geq t) \prod_{i=1}^{t} q(T \neq i-1|T \geq i-1)$ be a discrete probability distribution with support $\mathbb{N}_0$. Then for any $t \in \mathbb{N}_0$, we have that*

$$
q(T \geq t) = \sum_{i=t}^{\infty} q(T = i|T \geq i) \prod_{j=1}^{i} q(T \neq j-1|T \geq j-1) = \prod_{i=1}^{t} q(T \neq i-1|T \geq i-1). \tag{A.91}
$$

Proof. We proof the statement by induction on t.

Base case: For $t = 0$, $q(T \geq 0) = 1$ by definition of the empty product.

Inductive case: Note that $q(T \leq t) = \prod_{i=1}^{t} q(T = i-1 | T \geq i-1)$. Show that

$$q(T \geq t) = \prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1) \Longrightarrow q(T \geq t+1) = \prod_{i=1}^{t+1} q(T \neq i-1 | T \geq i-1). \tag{A.92}$$

Consider $q(T \geq t+1) = \sum_{i=t+1}^{\infty} q(T = i | T \geq i) \prod_{j=1}^{i} q(T \neq j-1 | T \geq j-1)$. To proof the inductive hypothesis, we need to show that the following equality is true:

$$\sum_{i=t+1}^{\infty} q(T = i | T \geq i) \prod_{j=1}^{i} q(T \neq j-1 | T \geq j-1) = \prod_{i=1}^{t+1} q(T \neq i-1 | T \geq i-1) \tag{A.93}$$

$$\begin{aligned} \Longleftrightarrow & \sum_{i=t}^{\infty} q(T = i | T \geq i) \prod_{j=1}^{i} q(T \neq j-1 | T \geq j-1) - q(T = t | T \geq t) \prod_{j=1}^{t} q(T \neq j-1 | T \geq j-1) \\ & = q(T \neq t | T \geq t) \prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1). \end{aligned} \tag{A.94}$$

By the inductive hypothesis,

$$q(T \geq t) = \sum_{i=t}^{\infty} q(T = i | T \geq i) \prod_{j=1}^{i} q(T \neq j-1 | T \geq j-1) = \prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1), \tag{A.95}$$

and so

$$\text{Equation (A.94)} \Longleftrightarrow \prod_{j=1}^{t} q(T \neq j | T \geq j) - q(T \neq t+1 | T \geq t+1) \prod_{j=1}^{t} q(T = j | T \geq j) \tag{A.96}$$

$$= q(T \neq t | T \geq t) \prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1). \tag{A.97}$$

Factoring out $\prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1)$, we get

$$\Longleftrightarrow \prod_{j=1}^{t} q(T \neq j-1 | T \geq j-1) \underbrace{(1 - q(T = t | T \geq t))}_{= q(T \neq t | T \geq t)} = q(T \neq t | T \geq t) \prod_{j=1}^{t} q(T = j-1 | T \geq j-1) \tag{A.98}$$

$$\Longleftrightarrow q(T \neq t | T \geq t) \prod_{j=1}^{t} q(T \neq j-1 | T \geq j-1) = q(T \neq t | T \geq t) \prod_{j=1}^{t} q(T \neq j-1 | T \geq j-1), \tag{A.99}$$

which proves the inductive hypothesis. ◻

Lemma 2. *Let $q_T(t)$ and $p_T(t)$ be discrete probability distributions with support $\mathbb{N}_0$, let Δ_t be a Bernoulli random variable, with success defined as $T = t+1$ given that $T \geq t$, and let q_{Δ_t} be a discrete probability distribution over Δ_t for $t \in \mathbb{N} \backslash \{0\}$, so that*

$$\begin{aligned} q_{\Delta_{t+1}}(\Delta_{t+1} = 0) &\stackrel{\text{def}}{=} q(T \neq t \,|\, T \geq t) \\ q_{\Delta_{t+1}}(\Delta_{t+1} = 1) &\stackrel{\text{def}}{=} q(T = t \,|\, T \geq t). \end{aligned} \tag{A.100}$$

Then we can write $q(T = t) = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \prod_{i=1}^{t} q_{\Delta_i}(\Delta_i = 0)$ for any $t \in \mathbb{N}_0$ and have that

$$q(T \geq t) = \sum_{i=t}^{\infty} q_{\Delta_{i+1}}(\Delta_{i+1} = 1) \prod_{j=1}^{i} q_{\Delta_j}(\Delta_j = 0) = \prod_{i=1}^{t} q_{\Delta_i}(\Delta_i = 0). \tag{A.101}$$

Proof. By Lemma 1, we have that for any $t \in \mathbb{N}_0$

$$q(T \geq t) = \sum_{i=t}^{\infty} q(T = i | T \geq i) \prod_{j=1}^{i} q(T \neq j-1 | T \geq j-1) = \prod_{i=1}^{t} q(T \neq i-1 | T \geq i-1). \tag{A.102}$$

The result follows by replacing $q(T = i|T \geq i)$ by $q_{\Delta_{i+1}}(\Delta_{i+1} = 1)$, $q(T \neq j-1|T \geq j-1)$ by $q_{\Delta_j}(\Delta_j = 0)$, and $q(T \neq i-1|T \geq i-1)$ by $q_{\Delta_i}(\Delta_i = 0)$. □

Lemma 3. *Let $q_T(t)$ and $p_T(t)$ be discrete probability distributions with support $\mathbb{N}_0$. Then for any $k \in \mathbb{N}_0$,*

$$\mathbb{E}_{t \sim q(T \,|\, T \geq k)}\left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)}\right] = f(q, p, k) + q(T \neq k \,|\, T \geq k)\, \mathbb{E}_{t \sim q(T \,|\, T \geq k+1)}\left[\log \frac{q(T = t \,|\, T \geq k+1)}{p(T = t \,|\, T \geq k+1)}\right]. \tag{A.103}$$

Proof. Consider $\mathbb{E}_{t \sim q(T \,|\, T \geq k)}\left[\log \frac{q(T=t \,|\, T \geq k)}{p(T=t \,|\, T \geq k)}\right]$ and note that by the law of total expectation we can rewrite it as

$$\begin{aligned}
&\mathbb{E}_{t \sim q(T \,|\, T \geq k)}\left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)}\right] \\
&= q(T = k \,|\, T \geq k)\, \mathbb{E}_{t \sim q(T \,|\, T = k)}\left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)}\right] + q(T \neq k \,|\, T \geq k)\, \mathbb{E}_{t \sim q(T \,|\, T \geq k+1)}\left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)}\right] \qquad \text{(A.104)} \\
&= q(T = k \,|\, T \geq k) \log \frac{q(T = k \,|\, T \geq k)}{p(T = k \,|\, T \geq k)} + q(T \neq k \,|\, T \geq k)\, \mathbb{E}_{t \sim q(T | T \geq k+1)}\left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)}\right]. \qquad \text{(A.105)}
\end{aligned}$$

For all values of $T \geq k+1$, we have that

$$q(T = t \,|\, T \geq k) = q(T = t \,|\, T \geq k+1) q(T \neq k \,|\, T \geq k) \tag{A.106}$$

$$p(T = t \,|\, T \geq k) = p(T = t \,|\, T \geq k+1) p(T \neq k \,|\, T \geq k) \tag{A.107}$$

and so we can rewrite the expectation in Equation (A.105) as

$$\begin{aligned}
\mathbb{E}_{t \sim q(T \,|\, T \geq k+1)}\left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)}\right] &= \mathbb{E}_{t \sim q(T \,|\, T \geq k+1)}\left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)} + \log \frac{q(T \neq k \,|\, T \geq k)}{p(T \neq k \,|\, T \geq k)}\right] \qquad \text{(A.108)} \\
&= \mathbb{E}_{t \sim q(T \,|\, T \geq k+1)}\left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)}\right] + \log \frac{q(T \neq k \,|\, T \geq k)}{p(T \neq k \,|\, T \geq k)} \qquad \text{(A.109)}
\end{aligned}$$

Combining Equation (A.109) with Equation (A.105), we have

$$\begin{aligned}
&\mathbb{E}_{t \sim q(T \,|\, T \geq k)}\left[\log \frac{q(T = t \,|\, T \geq k)}{p(T = t \,|\, T \geq k)}\right] \\
&= \underbrace{q(T = k \,|\, T \geq k) \log \frac{q(T = k \,|\, T \geq k)}{p(T = k \,|\, T \geq k)} + q(T \neq k \,|\, T \geq k) \log \frac{q(T \neq k \,|\, T \geq k)}{p(T \neq k \,|\, T \geq k)}}_{\stackrel{\text{def}}{=} f(q,p,k)} \qquad \text{(A.110)} \\
&\quad + q(T \neq k \,|\, T \geq k)\, \mathbb{E}_{t \sim q(T \,|\, T \geq k+1)}\left[\log \frac{q(T = t \,|\, T \geq k+1)}{p(T = t \,|\, T \geq k+1)}\right],
\end{aligned}$$

which concludes the proof. □

Lemma 4. *Let $q_T(t)$ and $p_T(t)$ be discrete probability distributions with support $\mathbb{N}_0$. Then the KL divergence from q_T to p_T can be written as*

$$D_{\mathrm{KL}}(q_T(\cdot \,|\, \mathbf{s}_0) \,||\, p_T(\cdot \,|\, \mathbf{s}_0)) = \sum_{t=0}^{\infty} q(T \geq t) f(q_T, p_T, t) \tag{A.111}$$

where $f(q_T, p_T, t)$ is shorthand for

$$f(q_T, p_T, t) = q(T = t \,|\, T \geq t) \log \frac{q(T = t \,|\, T \geq t)}{p(T = t \,|\, T \geq t)} + q(T \neq t \,|\, T \geq t) \log \frac{q(T \neq t \,|\, T \geq t)}{p(T \neq t \,|\, T \geq t)}. \tag{A.112}$$

Proof. Note that $q(T = k)$ denotes the probability that the distribution q assigns to the event $T = k$ and $q(T \geq m)$ denotes the tail probability, that is, $q(T \geq m) = \sum_{t=m}^{\infty} q(T = t)$. We will write $q(T|T \geq m)$ to denote the conditional distribution of q given $T \geq m$, that is, $q(T = k|T \geq m) = \mathbb{1}[k \geq m]q(T = k)/q(T \geq m)$. We will use analogous notation for p.

By the definition of the KL divergence and using the fact that, since the support is lowerbounded by $T = 0$, $q(T = 0) = q(T = 0 \,|\, T \geq 0)$, we have

$$D_{\mathrm{KL}}(q_T(\cdot \,|\, \mathbf{s}_0) \parallel p_T(\cdot \,|\, \mathbf{s}_0)) = \mathbb{E}_{t \sim q(T)} \left[\log \frac{q(T = t)}{p(T = t)} \right] = \mathbb{E}_{t \sim q(T \,|\, T \geq 0)} \left[\log \frac{q(T = t \,|\, T \geq 0)}{p(T = t \,|\, T \geq 0)} \right]. \tag{A.113}$$

Using Lemma 3 with $k = 0, 1, 2, 3, \ldots$, we can expand the above expression to get

$$\begin{aligned}
D_{\mathrm{KL}}(q_T(\cdot \,|\, \mathbf{s}_0) \parallel p_T(\cdot \,|\, \mathbf{s}_0)) &= f(q_T, p_T, 0) + q(T \neq 0 \,|\, T \geq 0)\, \mathbb{E}_{t \sim q(T \,|\, T \geq 1)} \left[\log \frac{q(T = t \,|\, T \geq 1)}{p(T = t \,|\, T \geq 1)} \right] \qquad (A.114) \\
&= f(q, p, 0) + q(T \neq 0 \,|\, T \geq 1) f(q_T, p_T, 1) \\
&\quad + q(T \neq 0 \,|\, T \geq 0) q(T \neq 1 \,|\, T \geq 1)\, \mathbb{E}_{t \sim q(T \,|\, T \geq 2)} \left[\log \frac{q(T = t \,|\, T \geq 2)}{p(T = t \,|\, T \geq 2)} \right] \qquad (A.115) \\
&= \underbrace{1}_{=q(T \geq 0)} \cdot f(q, p, 0) \\
&\quad + \underbrace{q(T \neq 0 \,|\, T \geq 0)}_{=q(T \geq 1)} f(q, p, 1) \\
&\quad + \underbrace{q(T \neq 0 \,|\, T \geq 0) q(T \neq 1 \,|\, T \geq 1)}_{=q(T \geq 2)} f(q_T, p_T, 2) \\
&\quad + \underbrace{q(T \neq 0 \,|\, T \geq 0) q(T \neq 1 \,|\, T \geq 1) q(T \neq 2 \,|\, T \geq 2)}_{=q(T \geq 3)} \mathbb{E}_{t \sim q(T \,|\, T \geq 3)} \left[\log \frac{q(T = t \,|\, T \geq 3)}{p(T = t \,|\, T \geq 3)} \right] \\
&\qquad (A.116) \\
&= \sum_{t=0}^{\infty} q(T \geq t) f(q_T, p_T, t), \qquad (A.117)
\end{aligned}$$

where $f(q_T, p_T, t)$ is shorthand for

$$f(q_T, p_T, t) = q(T = t \,|\, T \geq t) \log \frac{q(T = t \,|\, T \geq t)}{p(T = t \,|\, T \geq t)} + q(T \neq t \,|\, T \geq t) \log \frac{q(T \neq t \,|\, T \geq t)}{p(T \neq t \,|\, T \geq t)}. \tag{A.118}$$

and we used the fact that, by Lemma 1,

$$q(T \geq t) = \prod_{k=1}^{t} q(T \neq k - 1 \,|\, T \geq k - 1). \tag{A.119}$$

This completes the proof. $\square$

Lemma 5. *Let $q_T(t)$ and $p_T(t)$ be discrete probability distributions with support $\mathbb{N}_0$, let Δ_t be a Bernoulli random variable, with success defined as $T = t$ given that $T \geq t$, and let q_{Δ_t} and p_{Δ_t} be discrete probability distributions over Δ_t for $t \in \mathbb{N}_0 \backslash \{0\}$, so that*

$$q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \stackrel{\text{def}}{=} q(T \neq t \,|\, T \geq t) \qquad q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \stackrel{\text{def}}{=} q(T = t \,|\, T \geq t) \tag{A.120}$$

$$p_{\Delta_{t+1}}(\Delta_{t+1} = 0) \stackrel{\text{def}}{=} p(T \neq t \,|\, T \geq t) \qquad p_{\Delta_{t+1}}(\Delta_{t+1} = 1) \stackrel{\text{def}}{=} p(T = t \,|\, T \geq t). \tag{A.121}$$

Then the KL divergence from $q_T(\cdot \,|\, \mathbf{s}_0)$ to $p_T(\cdot \,|\, \mathbf{s}_0)$ can be written as

$$D_{\mathrm{KL}}(q_T(\cdot \,|\, \mathbf{s}_0) \,||\, p_T(\cdot \,|\, \mathbf{s}_0)) = \sum_{t=0}^{\infty} \Big(\prod_{k=1}^{t} q_{\Delta_t}(\Delta_t = 0) \Big) D_{\mathrm{KL}}(q_{\Delta_{t+1}} \,||\, p_{\Delta_{t+1}}) \tag{A.122}$$

Proof. The result follows from Lemma 4, Equation (A.119), Equation (A.120), and the definition of f.

In detail, from Lemma 1, and Equation (A.120) we have that

$$q(T \geq t) = \prod_{k=1}^{t} q(T \neq k-1 \,|\, T \geq k-1) = \prod_{k=1}^{t} q_{\Delta_k}(\Delta_k = 0). \tag{A.123}$$

From the definition of $f(q_T, p_T, t)$, we have

$$f(q_T, p_T, t) = q(T = t \,|\, T \geq t) \log \frac{q(T = t \,|\, T \geq t)}{p(T = t \,|\, T \geq t)} + q(T \neq t \,|\, T \geq t) \log \frac{q(T \neq t \,|\, T \geq t)}{p(T \neq t \,|\, T \geq t)} \tag{A.124}$$

$$= q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \log \frac{q_{\Delta_{t+1}}(\Delta_{t+1} = 0)}{p_{\Delta_{t+1}}(\Delta_{t+1} = 0)} + q(\Delta_{t+1} = 1) \log \frac{q_{\Delta_{t+1}}(\Delta_{t+1} = 1)}{p_{\Delta_{t+1}}(\Delta_{t+1} = 1)} \tag{A.125}$$

$$= D_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}). \tag{A.126}$$

Combining Equation (A.123), Equation (A.126), and Equation (A.111) completes the proof. □

B. Proof of Outcome-Driven Policy Iteration Theorem

Theorem 2 (Variational Outcome-Driven Policy Iteration)**.** *Assume $|\mathcal{A}| < \infty$ and that the MDP is ergodic.*

1. *Outcome-Driven Policy Evaluation (ODPE): Given policy π and a function $Q^0 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, define $Q^{i+1} = \mathcal{T}^{\pi} Q^i$. Then the sequence Q^i converges to the lower bound in Theorem 1.*
2. *Outcome-Driven Policy Improvement (ODPI): The policy that solves*

$$\pi^{+} = \arg\max_{\pi' \in \Pi} \left\{ \mathbb{E}_{\pi'(\mathbf{a}_t \,|\, \mathbf{s}_t)} \left[Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \right] - D_{\mathrm{KL}}(\pi'(\cdot \,|\, \mathbf{s}_t) \,||\, p(\cdot \,|\, \mathbf{s}_t)) \right\} \tag{B.127}$$

and the variational distribution over T recursively defined in terms of

$$\begin{aligned} & q^{+}(\Delta_{t+1} = 0 \,|\, \mathbf{s}_0; \pi, Q^{\pi}) \\ & = \sigma\Big(\mathbb{E}_{\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} [Q^{\pi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T)] - \mathbb{E}_{\pi(\mathbf{a}_t \,|\, \mathbf{s}_t)} [\log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t)] + \sigma^{-1}\left(p_{\Delta_{t+1}}(\Delta_{t+1} = 0)\right) \Big) \end{aligned} \tag{B.128}$$

improve the variational objective. In other words, $\mathcal{F}(\pi^{+}, q_T, \mathbf{s}_0) \geq \mathcal{F}(\pi, q_T, \mathbf{s}_0)$ and $\mathcal{F}(\pi, q_T^{+}, \mathbf{s}_0) \geq \mathcal{F}(\pi, q_T, \mathbf{s}_0)$ for all $\mathbf{s}_0 \in \mathcal{S}$.

3. *Alternating between ODPE and ODPI converges to a policy $\pi^{\star}$ and a variational distribution over T, $q_T^{\star}$, such that $Q^{\pi^{\star}}(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_T^{\star}) \geq Q^{\pi}(\mathbf{s}, \mathbf{a}, \mathbf{g}; q_T)$ for all $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ and any $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$.*

Proof. Parts of this proof are adapted from the proof given in Haarnoja et al. (2018a), modified for the Bellman operator proposed in Definition 1.

1. Outcome-Driven Policy Evaluation (ODPE): Instead of absorbing the entropy term into the Q-function, we can define an entropy-augmented reward as

$$\begin{aligned} r^{\pi}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_{\Delta}) \stackrel{\text{def}}{=} {} & q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) - D_{\mathrm{KL}}(q_{\Delta_{t+1}} \,\|\, p_{\Delta_{t+1}}) \\ & + q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} [D_{\mathrm{KL}}(\pi(\cdot \,|\, \mathbf{s}_{t+1}) \,\|\, p(\cdot \,|\, \mathbf{s}_{t+1}))]. \end{aligned} \tag{B.129}$$

We can then write an update rule according to Definition 1 as

$$\tilde{Q}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \leftarrow r^{\pi}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_{\Delta}) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \, \mathbb{E}_{\pi(\mathbf{a}_{t+1} \,|\, \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)} [\tilde{Q}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T)], \tag{B.130}$$

where $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) \leq 1$. This update is similar to a Bellman update (Sutton and Barto, 1998), but with a discount factor given by $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$. In general, this discount factor $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ can be computed dynamically based on the current state and action, such as in Equation (14). As discussed in White (2017), this Bellman operator is still a contraction mapping so long as the Markov chain induced by the current policy is ergodic and there exists a state such that $q_{\Delta_{t+1}}(\Delta_{t+1} = 0) < 1$. The first condition is true by assumption. The second condition is true since $q_{\Delta_{t+1}}(\Delta_{t+1} = 0)$ is given by Equation (14), which is always strictly between 0 and 1. Therefore, we apply convergence results for policy evaluation with transition-dependent discount factors (White, 2017) to this contraction mapping, and the result immediately follows.

2. Outcome-Driven Policy Improvement (ODPI): Let $\pi_{\text{old}} \in \Pi$ and let $Q^{\pi_{\text{old}}}$ and $V^{\pi_{\text{old}}}$ be the outcome-driven state and state-action value functions from Definition 1, let q_T be some variational distribution over T, and let π_{new} be given by

$$\begin{aligned}\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t) &= \arg\max_{\pi' \in \Pi} \left\{ \mathbb{E}_{\pi'(\mathbf{a}_t \,|\, \mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \right] - D_{\text{KL}}(\pi'(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \right\} \\ &= \arg\max_{\pi' \in \Pi} \mathcal{J}_{\pi_{\text{old}}}(\pi'(\mathbf{a}_t, \mathbf{s}_t), q_T).\end{aligned} \tag{B.131}$$

Then, it must be true that $\mathcal{J}_{\pi_{\text{old}}}(\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t); q_T) \leq \mathcal{J}_{\pi_{\text{old}}}(\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t); q_T)$, since one could set $\pi_{\text{new}} = \pi_{\text{old}} \in \Pi$. Thus,

$$\begin{aligned}&\mathbb{E}_{\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \right] - D_{\text{KL}}(\pi_{\text{new}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \\ &\quad \geq \mathbb{E}_{\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \right] - D_{\text{KL}}(\pi_{\text{old}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)),\end{aligned} \tag{B.132}$$

and since

$$V^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{g}; q_T) = \mathbb{E}_{\pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \right] - D_{\text{KL}}(\pi_{\text{old}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)), \tag{B.133}$$

we get

$$\mathbb{E}_{\pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t)} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) \right] - D_{\text{KL}}(\pi_{\text{new}}(\cdot \,|\, \mathbf{s}_t) \parallel p(\cdot \,|\, \mathbf{s}_t)) \geq V^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{g}; q_T). \tag{B.134}$$

We can now write the Bellman equation as

$$\begin{aligned}&Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) && \text{(B.135)} \\ &\quad = q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p_d(\mathbf{s}_{t+1} \,|\, \mathbf{s}_t, \mathbf{a}_t)}[V^{\pi_{\text{old}}}(\mathbf{s}_{t+1}, \mathbf{g}; q_T)] && \text{(B.136)} \\ &\quad \leq q_{\Delta_{t+1}}(\Delta_{t+1} = 1) \log p_d(\mathbf{g} \,|\, \mathbf{s}_t, \mathbf{a}_t) \\ &\qquad + q_{\Delta_{t+1}}(\Delta_{t+1} = 0)\, \mathbb{E}_{p(\mathbf{s}_{t'}|\mathbf{s}_t, \mathbf{a}_t)}[\mathbb{E}_{\pi_{\text{new}}(\mathbf{a}_{t'}|\mathbf{s}_{t'})} \left[Q^{\pi_{\text{old}}}(\mathbf{s}_{t'}, \mathbf{a}_{t'}, \mathbf{g}; q_T) \right] - D_{\text{KL}}(\pi_{\text{new}}(\cdot \,|\, \mathbf{s}_{t'}) \parallel p(\cdot \,|\, \mathbf{s}_{t'}))], && \text{(B.137)} \\ &\quad \vdots \\ &\quad \leq Q^{\pi_{\text{new}}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_T) && \text{(B.138)}\end{aligned}$$

where we defined $t' \stackrel{\text{def}}{=} t + 1$, repeatedly applied the Bellman backup operator defined in Definition 1 and used the bound in Equation (B.134). Convergence follows from Outcome-Driven Policy Evaluation above.

3. Locally Optimal Variational Outcome-Driven Policy Iteration: Define π^i to be a policy at iteration i. By ODPI for a given q_T, the sequence of state-action value functions $\{Q^{\pi^i}(q_T)\}_{i=1}^{\infty}$ is monotonically increasing in i. Since the reward is finite and the negative KL divergence is upper bounded by zero, $Q^{\pi}(q_T)$ is upper bounded for $\pi \in \Pi$ and the sequence $\{\pi^i\}_{i=1}^{\infty}$ converges to some $\pi^\star$. To see that $\pi^\star$ is an optimal policy, note that it must be the case that $\mathcal{J}_{\pi^\star}(\pi^\star(\mathbf{a}_t|\mathbf{s}_t); q_T) > \mathcal{J}_{\pi^\star}(\pi(\mathbf{a}_t \,|\, \mathbf{s}_t); q_T)$ for any $\pi \in \Pi$ with $\pi \neq \pi^\star$. By the argument used in ODPI above, it must be the case that the outcome-driven state-action value of the converged policy is higher than that of any other non-converged policy in Π, that is, $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T)$ for all $\pi \in \Pi$ and any $q_T^i \in \mathcal{Q}_T$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Therefore, given q_T, $\pi^\star$ must be optimal in Π, which concludes the proof.

4. Globally Optimal Variational Outcome-Driven Policy Iteration: Let π^i be a policy and let q_T^i be variational distributions over T at iteration i. By Locally Optimal Variational Outcome-Driven Policy Iteration, for a *fixed* q_T^i with $q_T^i = q_T^j \forall i, j \in \mathbb{N}_0$, the sequence of $\{(\pi^i, q_T^i)\}_{i=1}^{\infty}$ increases the objective Equation (9) at each iteration and converges to a stationary point in π^i, where $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^i) > Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t; q_T^i)$ for all $\pi \in \Pi$ and any $q_T^i \in \mathcal{Q}_T$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$.

Since the objective in Equation (9) is concave in q_T, it must be the case that for, $q_T^{\star^i} \in \mathcal{Q}_T$, the optimal variational distribution over T at iteration i, defined recursively by

$$q^{\star^i}(\Delta_{t+1} = 0; \pi^i, Q^{\pi^i}) = \sigma\Big(\mathbb{E}_{\pi(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) p_d(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)}[Q^{\pi^i}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}, \mathbf{g}; q_T(\pi^i, Q^{\pi^i}))] \\ - \mathbb{E}_{\pi(\mathbf{a}_t \mid \mathbf{s}_t)}[\log p_d(\mathbf{g} \mid \mathbf{s}_t, \mathbf{a}_t)] + \sigma^{-1}(p_{\Delta_{t+1}}(\Delta_{t+1} = 0)\Big),$$

for $t \in \mathbb{N}_0$, $Q^\pi(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^\pi(\mathbf{s}_t, \mathbf{a}_t; q_T)$ for all $\pi \in \Pi$ and any $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Note that q_T is defined implicitly in terms of π^i and Q^{π^i}, that is, the optimal variational distribution over T at iteration i is defined as a function of the policy and Q-function at iteration i. Hence, it must then be true that for $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T)$ for all $q_T^\star(\pi^\star, Q^{\pi^\star}) \in \mathcal{Q}_T$ and for any $\pi^\star \in \Pi$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. In other words, for an optimal policy and corresponding Q-function, there exists an optimal variational distribution over T that maximizes the Q-function, given the optimal policy. Repeating locally optimal variational outcome-driven policy iteration under the new variational distribution $q_T^\star(\pi^\star, Q^{\pi^\star})$ will yield an optimal policy $\pi^{\star\star}$ and computing the corresponding optimal variational distribution, $q_T^{\star\star}(\pi^{\star\star}, Q^{\pi^{\star\star}})$ will further increase the variational objective such that for $\pi^{\star\star}) \in \Pi$ and $q_T^{\star\star}(\pi^{\star\star}, Q^{\pi^{\star\star}}) \in \mathcal{Q}_T$, we have that

$$Q^{\pi^{\star\star}}(\mathbf{s}_t, \mathbf{a}_t; q_T^{\star\star}) > Q^{\pi^{\star\star}}(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T) \tag{B.139}$$

for any $\pi^\star \in \Pi$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$. Hence, global optimal variational outcome-driven policy iteration increases the variational objective at every step. Since the objective is upper bounded (by virtue of the rewards being finite and the negative KL divergence being upper bounded by zero) and the sequence of $\{(\pi^i, q_T^i)\}_{i=1}^\infty$ increases the objective Equation (9) at each iteration, by the monotone convergence theorem, the objective value converges to a supremum and since the objective function is concave the supremum is unique. Hence, since the supremum is unique and obtained via global optimal variational outcome–driven policy iteration on $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$, the sequence of $\{(\pi^i, q_T^i)\}_{i=1}^\infty$ converges to a unique stationary point $(\pi^\star, q_T^\star) \in \Pi \times \mathcal{Q}_T$, where $Q^{\pi^\star}(\mathbf{s}_t, \mathbf{a}_t; q_T^\star) > Q^\pi(\mathbf{s}_t, \mathbf{a}_t; q_T^i)$ for all $\pi \in \Pi$ and any $q_T^i \in \mathcal{Q}_T$ and $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$.

□

Corollary 3 Optimality of Variational Outcome Driven Policy Iteration**.** *Variational Outcome-Driven Policy Iteration on $(\pi, q_T) \in \Pi \times \mathcal{Q}_T$ results in an optimal policy at least as good or better than any optimal policy attainable from policy iteration on $\pi \in \Pi$ alone.*

Remark 3. *The convergence proof of ODPE assumes a transition-dependent discount factor (White, 2017), because the variational distribution used in Equation (14) depends on the next state and action as well as on the desired outcome.*

C. Additional Experiments

Description of Figure 1 We implemented a tabular version of ODAC and applied it to the 2D environment shown in Figure 3. We discretize the environment into an 8×8 grid of states. The action correspond to moving up, down, left, or right. If probability $1 - \epsilon$, this action is taken. If the agent runs into a wall or boundary, the agent stays in its current state. With probability $\epsilon = 0.1$, the commanded action is ignored and a neighboring state grid (including the current state) is uniformly sampled as the next state. The policy and Q-function are represented with look-up tables and randomly initialized. The entropy reward is weighted by 0.01 and the time prior $p(T)$ is geometric with parameter 0.5. The dynamics model, $p_d^{(0)}$ is initialized to give a uniform probability to each states for every state and action. Each iteration, we simulate data collection by updating the dynamics model with the running average update

$$p_d^{(t+1)} = 0.99 p_d^{(t)} + 0.01 p_d$$

where p_d is the true dynamics and update the policy and Q-function according to Equation (18) and Equation (16), respectively. Figure 1 shows that, in contrast to the binary-reward setting, the learned reward provides shaping for the policy, which solves the task within 100 iterations.

Full Ablation Results We show the full ablation learning curves in Figure 6. We see that ODAC consistently performs the best, and that ODAC with a fixed model also performs well. However, on a few tasks, and in particular the Fetch Push and Sawyer Faucet tasks, we see that using a fixed q_T hurts the performance, suggesting that our derived formula in Equation (14) results in better empirical performance.

Comparison to methods with oracle goal sampling. To demonstrate the impact of sampling the desired outcome $\mathbf{g}$ during exploration, we evaluate the methods on the Fetch task when using oracle goal sampling. As shown in Figure 7, we see that the performances of UVD and ODAC are similar and that both outperform other methods.

As shown in Figure 4, ODAC performs well on both this setting and the harder setting where the desired outcome $\mathbf{g}$ was fixed during exploration, suggesting that ODAC does not rely as heavily on the uniform sampling of $\mathbf{g}$ to learn a good policy than do other methods.

Comparison to model-based planning. ODAC learns a dynamics model but does not use it for planning and instead relies on the derived Bellman updates to obtain a policy. However, a natural question is whether or not the method would benefit from using this model to perform model-based planning, as in **?**. We assess this by comparing ODAC with model-based baseline that uses a 1-step look-ahead. In particular, we follow the training procedure in **?** with $k = 1$. To ensure a fair comparison, we use the exact same dynamics model architecture as in ODAC and match the update-to-environment step ratio to be 4-to-1 for both methods. Table 2 shows the final distance to the goal (best results in bold). Using the same dynamics model, ODAC, which does not use the dynamics model to perform planning and only uses it to compute rewards, outperforms the model-based planning method. While a better model might lead to better performance for the model-based baseline, these results suggest that ODAC is not sensitive to model quality to the same degree as model-based planning methods.

Reward Visualization We visualize the reward for the Box 2D environment in Figure 8. We see that over the course of training, the reward function initially flattens out near $\mathbf{g}$, making learning easier by encouraging the policy to focus on moving just out of the top left corner of the environment. Later in training (around 16,000 steps), the policy learns to move out of the top left corner, and we see that the reward changes to have a stronger reward gradient near $\mathbf{g}$. We also note that the reward are much more negative for being far $\mathbf{g}$ at the end of training: the top left region changes from having a penalty of -1.6 to -107. Overall, these visualizations show that the reward function automatically changes during training and provides a strong reward signal for different parts of the state space depending on the behavior of the policy.

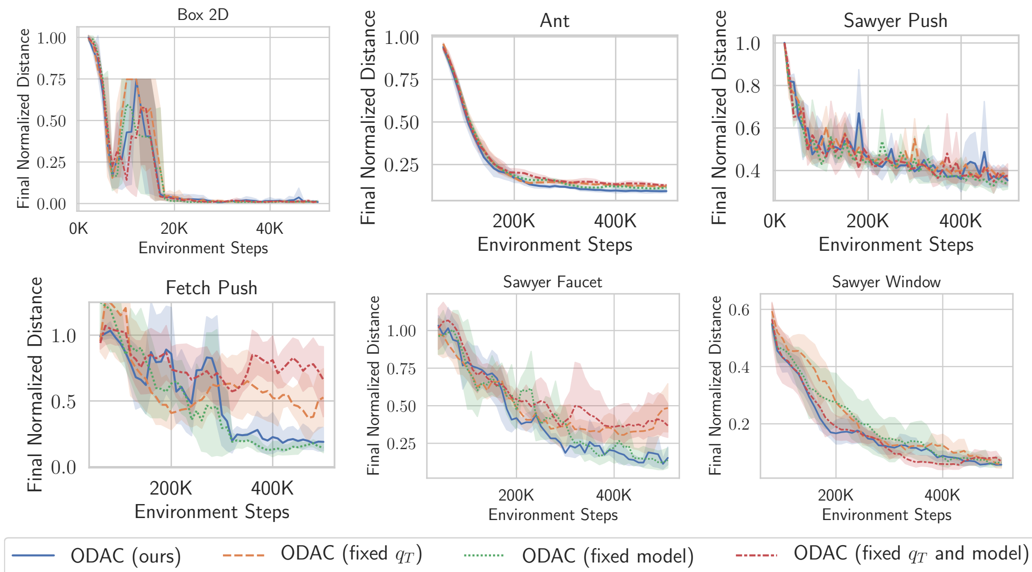


Figure 6: Ablation results across all six environments. We see that using our derived q_T equation is important for best performance across all six tasks and that ODAC is not sensitive to the quality of dynamics model.

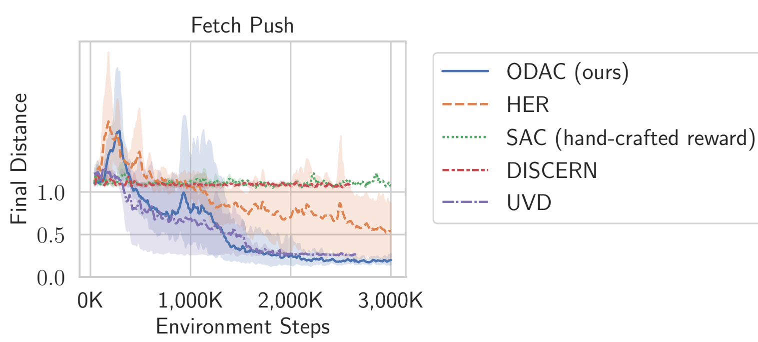
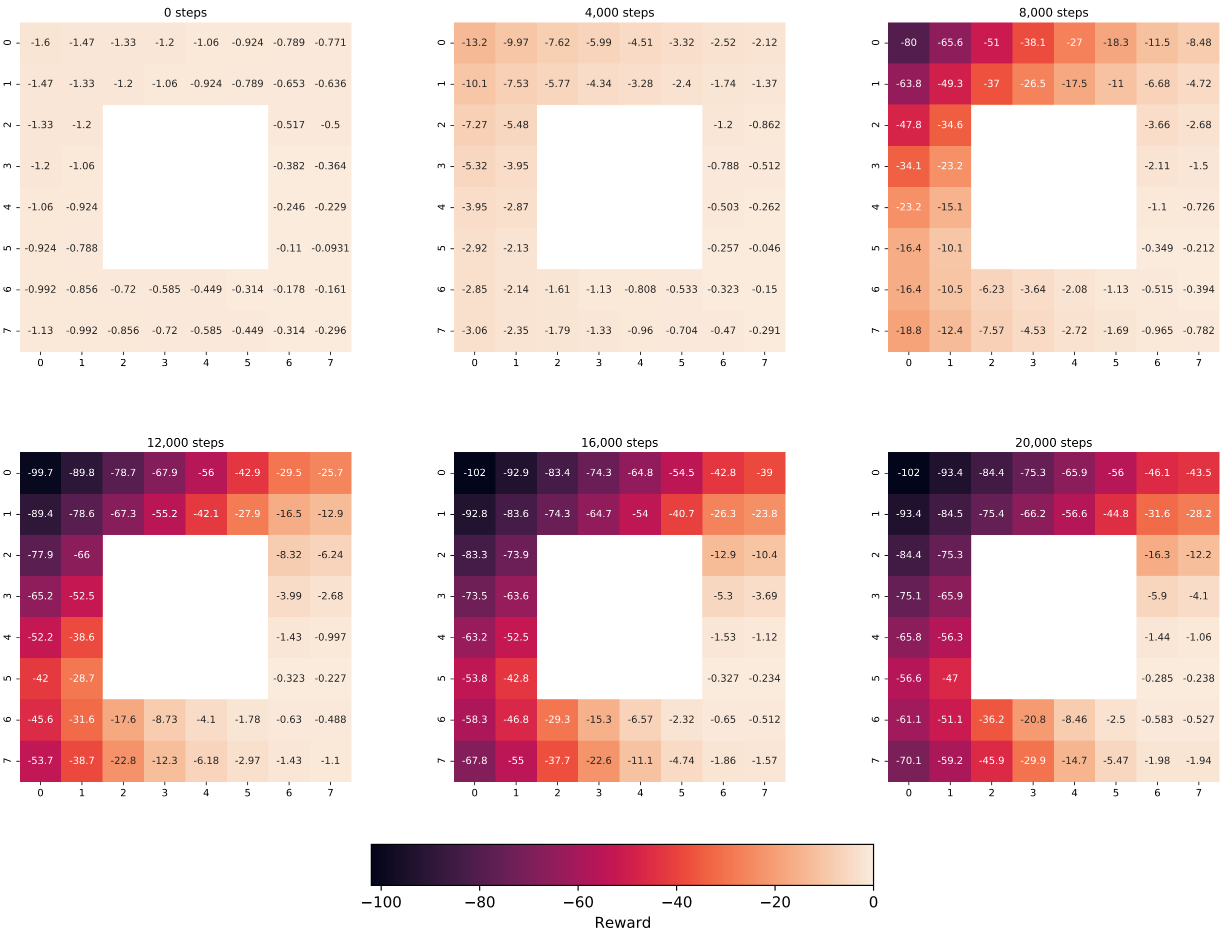


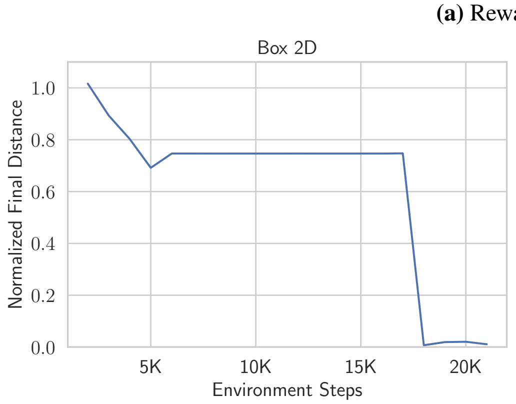
Figure 7: Comparison of different methods when the desired outcome $\mathbf{g}$ is sampled uniformly from the set of possible outcomes during exploration. In this easier setting, we see that the UVD performance is similar to ODAC. These results suggest that UVD depends more heavily on sampling outcomes from the set of desired outcomes than ODAC.

Environment	ODAC (Mean + Standard Error)	Dyna (Mean + Standard Error)
Box 2D	0.74 (0.091)	0.87 (0.058)
Ant	33 (27)	102 (0.83)
Sawyer Faucet	14 (6.3)	100 (5)
Fetch Push	12 (3.7)	96 (3.8)
Sawyer Push	58 (8.7)	96 (0.39)
Sawyer Window	4.4 (1.5)	116 (14)

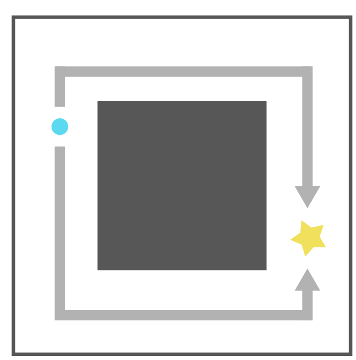
Table 2: Normalized final distances (lower is better) across four random seeds, multiplied by a factor of 100.



(a) Reward Visualization



(b) Learning Curve associated with Reward Visualized



(c) Environment Visualization

Figure 8: We visualize the rewards over the course of training on a single random seed for the Box 2D environment. To visualize the reward, we discretize the continuous state space and evaluate $r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}; q_\Delta)$ for $\mathbf{a} = \vec{0}$ at different states. As shown in Figure 8c, the desired outcome $\mathbf{g}$ is near the bottom right and the states in the center are invalid. After 4-8 thousand environment steps, the reward is more flat near $\mathbf{g}$, and only provides a reward gradient far from $\mathbf{g}$. After 20 thousand environment steps, the reward gradient is much larger again near the end, and the penalty for being in the top left corner has changed from -1.6 to -107.

D. Experimental Details

D.1. Environment

Ant. This Ant domain is based on the "Ant-V3" OpenAI Gym (Brockman et al., 2016) environment, with three modifications: the gear ratio is reduced from 150 to 120, the contact force sensors are removed from the state, and there is no termination condition and the episode only terminates after a fixed amount of time. In this environment, the state space is 23 dimensional, consistent of the XYZ coordinate of the center of the torso, the orientation of the ant (in quaternion), and the angle and angular velocity of all 8 joints. The action space is 8-dimensional and corresponds to the torque to apply to each joint. The desired outcome consists of the desired XYZ, orientation, and joint angles at a position that is 5 meters down and to the right of the initial position. This desired pose is shown in Figure 4.

Sawyer push. In this environment, the state and goal space is 4 dimensional and the action space is 2 dimension. The state and goal consists of the XY end effector (EE) and the XY position of the puck. The object is on a 40cm x 50cm table and starts 20 cm in front of the hand. The goal puck position is fixed to 15 cm forward and 30 cm to the right of the initial hand position, while the goal hand position is 5cm behind and 20 cm to the right of the initial hand position. The action is the change in position in each XY direction, with a maximum change of 3 cm per direction at each time step. The episode horizon is 100.

Box 2D. In this environment, the state is a $4x4$ with a $2x2$ box in the middle. The policy is initialized to to $(-3.5, -2)$ and the desired outcome is $(3.5, 2)$. The action is the XY velocity of the agent, with wall collisions taken into account and maximum velocity of 0.2 in each direction. To make the environment stochastic, we add Gaussian noise to actions with mean zero and standard deviation that's 10% of the maximum action magnitude.

Sawyer window and faucet. In this environment, the state and goal space is 6 dimensional and the action space is 2 dimension. The state and goal consists of the XYZ end effector (EE) and the XYZ position of the window or faucet end endpoint. The hand is initialized away from the window and faucet. The EE goal XYZ position is set to the initial window or faucet position. The action is the change in position in each XYZ direction. For the window task, the goal positions is to close the window, and for the faucet task, the goal position is to rotate the faucet 90 degrees counter-clockwise from above.

D.2. Algorithm

Pseudocode for the complete algorithm is shown in Algorithm 2.

D.3. Implementation

Table 4 lists the hyper-parameters that were shared across the experiments. Table 3 lists hyper-parameters specific to each environment. We give extra implementation details.

Dynamics model. For the Ant and Sawyer experiments, we train a neural network to output the mean and standard deviation of a Laplace distribution. This distribution is then used to model the distribution over the *difference* between the current state and the next state, which we found to be more reliable than predicting the next state. So, the overall distribution is given by a Laplace distribution with learned mean μ and fixed standard deviation σ computed via

$$p_\psi = \text{Laplace}(\mu = g_\psi(\mathbf{s}, \mathbf{a}) + f(\mathbf{s}), \sigma = 0.00001)$$

where g is the output of a network and f is a function that maps a state into a goal.

For the 2D Navigation experiment, we use a Gaussian distribution. The dynamics neural network has hidden units of size $[64, 64]$ with a ReLU hidden activations. For the Ant and Sawyer experiments, there is no output activation. For the linear-Gaussiand and 2D Navigation experiments, we have a tanh output, so that the mean and standard To bound the standard deviation outputted by the network, the standard-deviation tanh is multiplied by two with the standard deviation be between limited to between

Reward normalization. Because the different experiments have rewards of very different scale, we normalize the rewards by dividing by a running average of the maximum reward magnitude. Specifically, for every reward r in the ith batch of

Algorithm 2 Outcome-Driven Actor Critic

Require: Policy π_θ, Q-function Q_ϕ, dynamics model p_ψ, replay buffer $\mathcal{R}$, and map from state to achieved goal f.
  for $n = 0, \ldots N-1$ episodes **do**
    Sample initial state $\mathbf{s}_0$ from environment.
    Sample goal $\mathbf{g}$ from environment.
    for $t = 0, \ldots, H-1$ steps **do**
      Get action $\mathbf{a}_t \sim \pi_\theta(\mathbf{s}_t, \mathbf{g})$.
      Get next state $\mathbf{s}_{t+1} \sim p(\cdot \mid \mathbf{s}_t, \mathbf{a}_t)$.
      Store $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \mathbf{g})$ into replay buffer $\mathcal{R}$.
      Sample transition $(\mathbf{s}, \mathbf{a}, \mathbf{s}', \mathbf{g}) \sim \mathcal{R}$.
      Compute reward $r = \log p_\psi(\mathbf{g} \mid \mathbf{s}, \mathbf{a}) - D_{\mathrm{KL}}(q_\Delta(\cdot|\mathbf{s}_t, \mathbf{a}_t) \parallel p(\Delta))$.
      Compute $q(\Delta_t = 0|\mathbf{s}, \mathbf{a})$ using Equation (14).
      Update Q_ϕ using Equation (19) and data $(\mathbf{s}, \mathbf{a}, \mathbf{s}', \mathbf{g}, r)$.
      Update π_θ using Equation (20) and data $(\mathbf{s}, \mathbf{a}, \mathbf{g})$.
      Update p_ψ using Equation (21) and data $(\mathbf{s}, \mathbf{a}, \mathbf{g})$.
    end for
    for $t = 0, ..., H-1$ steps **do**
      for $i = 0, ..., k-1$ steps **do**
        Sample future state $\mathbf{s}_{h_i}$, where $t < h_i \leq H-1$.
        Store $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, f(\mathbf{s}_{h_i}))$ into $\mathcal{R}$.
      end for
    end for
  end for

data, we replace the reward with

$$\hat{r} = r / C_i$$

where we update the normalizing coefficient C_i using each batch of reward $\{r_b\}_{b=1}^B$:

$$C_{i+1} \leftarrow (1-\lambda) \times C_i + \lambda \max_{b \in [1, \ldots, B]} |r_b|$$

and C_i is initialized to 1. In our experiments, we use $\lambda = 0.001$.

Target networks. To train our Q-function, we use the technique from Fujimoto et al. (2018) in which we train two separate Q-networks with target networks and take the minimum over two to compute the bootstrap value. The target networks are updated using a slow, moving average of the parameters after every batch of data:

$$\bar{\phi}_{i+1} = (1-\tau)\bar{\phi}_i + \times \phi_i.$$

In our experiments, we used $\tau = 0.001$.

Automatic entropy tuning. We use the same technique as in Haarnoja et al. (2018b) to weight the rewards against the policy entropy term. Specifically, we pre-multiply the entropy term in

$$\hat{V}(\mathbf{s}', \mathbf{g}) \approx Q_{\bar{\phi}}(\mathbf{s}', \mathbf{a}', \mathbf{g}) - \log \pi(\mathbf{a}'|\mathbf{s}'; \mathbf{g}),$$

by a parameter α that is updated to ensure that the policy entropy is above a minimum threshold. The parameter α is updated by taking a gradient step on the following function with each batch of data:

$$\mathcal{F}_\alpha(\alpha) = -\alpha \left(\log \pi(\mathbf{a} \mid \mathbf{s}, \mathbf{g}) + \mathcal{H}_{\text{target}}\right)$$

and where $\mathcal{H}_{\text{target}}$ is the target entropy of the policy. We follow the procedure in Haarnoja et al. (2018b) to choose $\mathcal{H}_{\text{target}}$ and choose $\mathcal{H}_{\text{target}} = -D_{\text{action}}$, where D_{action} is the dimension of the action space.

Environment	horizon	Q-function and policy hidden sizes
Box 2D	100	[64, 64]
Ant	100	[400, 300]
Fetch Push	50	[64, 64]
Sawyer Push	100	[400, 300]
Sawyer Window	100	[400, 300]
Sawyer Faucet	100	[400, 300]

Table 3: Environment specific hyper-parameters.

Hyper-parameter	**Value**
# training batches per environment step	1
batch size	256
discount Factor	0.99
policy hidden activation	ReLU
Q-function hidden activation	ReLU
replay buffer size	1 million
hindsight relabeling strategy	future
hindsight relabeling probability	80%
target network update speed τ	0.001
reward scale update speed λ	0.001

Table 4: General hyper-parameters used for all experiments.

Exploration policy. Because ODAC is an off-policy algorithm, we are free to use any exploration policy. It may be beneficial to add For the Ant and Sawyer tasks, we simply sample current policy. For the 2D Navigation task, at each time step, the policy takes a random action with probability 0.3 and repeats its

Evaluation policy. For evaluation, we use the mean of the learned policy for selecting actions.