Generalised Policy Improvement with Geometric Policy Composition

Shantanu Thakoor [*1] **Mark Rowland** [*1] **Diana Borsa** [1] **Will Dabney** [1] **Rémi Munos** [1] **André Barreto** [1]

Abstract

We introduce a method for policy improvement that interpolates between the greedy approach of value-based reinforcement learning (RL) and the full planning approach typical of model-based RL. The new method builds on the concept of a geometric horizon model (GHM, also known as a γ-model), which models the discounted state-visitation distribution of a given policy. We show that we can evaluate any non-Markov policy that switches between a set of base Markov policies with fixed probability by a careful composition of the base policy GHMs, *without any additional learning*. We can then apply generalised policy improvement (GPI) to collections of such non-Markov policies to obtain a new Markov policy that will in general outperform its precursors. We provide a thorough theoretical analysis of this approach, develop applications to transfer and standard RL, and empirically demonstrate its effectiveness over standard GPI on a challenging deep RL continuous control task. We also provide an analysis of GHM training methods, proving a novel convergence result regarding previously proposed methods and showing how to train these models stably in deep RL settings.

1. Introduction

Policy improvement is at the heart of reinforcement learning (RL). The prototypical approach to policy improvement in value-based RL is to take the Q-function of a policy and act *greedily* with respect to it. In contrast, in model-based RL, planning with a model in principle aims to derive a (near-)optimal policy directly. Choosing between these two extremes involves some trade-offs. While greedy improvement requires estimating only a Q-function, from which it is computationally trivial to derive the greedy policy, this may result in only a weak improvement over the existing policy. Planning, on the other hand, is a computationally intensive process, yet can yield extremely high-quality policies. In this paper, we introduce an approach to policy improvement that interpolates between these two extremes.

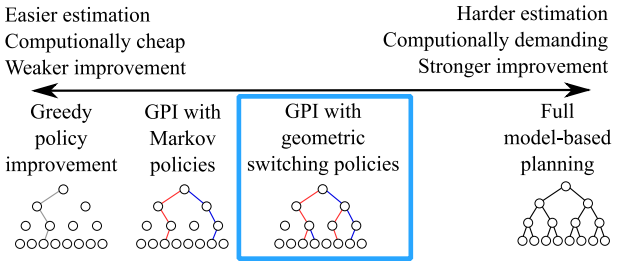


Figure 1. A spectrum of trade-offs in policy improvement. Barreto et al. (2017) propose generalised policy improvement (GPI) as a means of simultaneously improving over several policies (illustrated with blue and red trajectories), a step from greedy improvement of a single policy towards planning. The central contribution of this paper, GPI with geometric switching policies, moves a step further in this direction, allowing for improvement over non-Markov GSPs (illustrated as trajectories that switch between blue and red base policies).

Barreto et al. (2017) propose generalised policy improvement (GPI), a method that allows for improvement over a *collection* of policies $\{\pi_1, \ldots, \pi_k\}$ simultaneously, generalising the notion of greedy improvement of an individual policy. We show that GPI can be extended to a much wider class of *non-Markov* policies. These policies, which we call *geometric switching policies* (GSPs), switch between executing a base set of Markov policies $\{\pi_1, \ldots, \pi_k\}$ with fixed probability. In general, these policies do not ever need to be executed, and can instead be evaluated using information learnt about the base policies, without any further learning required, leading to a stronger improvement guarantee in GPI. This approach to policy improvement makes statistical and computational trade-offs that interpolate between greedy improvement and full model-based planning, potentially providing benefits of both worlds; Figure 1 shows where the proposed approach lies along the spectrum of methods between the conventional model-free and model-based extremes.

[*]Equal contribution. Order determined by coin flip. [1]DeepMind, London. Correspondence to: Shantanu Thakoor <thakoor@deepmind.com>, Mark Rowland <markrowland@deepmind.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Central to our approach is the notion of a geometric horizon model (GHM) (Janner et al., 2020), which models the discounted future state-visitation distribution of a given Markov policy. Janner et al. (2020) introduced GHMs mainly as a mechanism to compute the value function of a single policy. In this paper we show that GHMs of *distinct* policies can be composed to evaluate a potentially large number of GSPs with no additional learning required. We can then apply GPI over this collection of non-Markov policies to obtain a new Markov policy that will in general outperform all of its precursors (base policies *and* switching policies).

In carrying out the above, we address several technical questions which are contributions in their own right. Specifically, our central technical contributions include:

- *GSP evaluation with GHMs*, a method for evaluating geometric switching policies that only requires learning GHMs for a base class of Markov policies (Section 3).
- *Geometric generalised policy improvement* (GGPI), a method for deriving a Markov policy that improves over a collection of geometric switching policies, interpolating between greedy improvement and full model-based planning (Section 4).
- Convergence analysis of *cross-entropy temporal-difference learning*, an algorithm introduced by Janner et al. (2020) for learning GHMs (Section 6).
- New practical methods and insights for training GHMs at scale, including cross-entropy temporal-difference learning with VAE-GHMs (Section 7).
- Applications of GHM evaluation and GGPI to both transfer and standard RL settings (Section 5), with successful implementation in combination with deep learning in continuous control tasks (Section 7).

2. Background

A Markov decision process (MDP) is specified by a state space $\mathcal{X}$, action space $\mathcal{A}$, transition probabilities $P : \mathcal{X} \times \mathcal{A} \to \mathscr{P}(\mathcal{X})$, reward distributions $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathscr{P}_1(\mathbb{R})$, and corresponding expected reward function $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, defined by $r(x, a) = \mathbb{E}_{R\sim\mathcal{R}(x,a)}[R]$. For ease of presentation, we focus on the case where $\mathcal{X}$ is finite, although much of the material of the paper extends to more general state spaces. An agent interacting with the environment using a policy $\pi : \mathcal{X} \to \mathscr{P}(\mathcal{A})$ generates a trajectory of states, actions, and rewards $(X_t, A_t, R_t)_{t\geq 0}$, and the agent's return along this trajectory is defined by $\sum_{t\geq 0} \gamma^t R_t$, where $\gamma \in [0, 1)$ is the discount factor. The agent's expected return under π when beginning in state x and initially taking action a is $Q^\pi_\gamma(x, a) = \mathbb{E}^\pi_{x,a}[\sum_{t\geq 0} \gamma^t R_t]$, where $\mathbb{E}^\pi_{x,a}$ and $\mathbb{P}^\pi_{x,a}$ denote the expectation operator and probability distribution of a trajectory beginning at state-action pair (x, a) and following π thereafter. The goal of *policy evaluation* is to estimate Q^π_γ for a policy π of interest, while the goal of *policy optimisation* is to obtain a policy π^* with $Q^{\pi^*}_\gamma \geq Q^\pi_\gamma$ component-wise for all other policies $\pi \in \mathscr{P}(\mathcal{A})^\mathcal{X}$ (Sutton & Barto, 2018; Puterman, 2014; Bertsekas & Tsitsiklis, 1996; Szepesvári, 2010; Meyn, 2022). A fundamental operation in this process is *policy improvement*, described below.

2.1. Generalised policy improvement

We first recall a core method for policy improvement in RL.

Greedy policy improvement. The greedy policy improvement map $\mathcal{G} : \mathbb{R}^{\mathcal{X}\times\mathcal{A}} \rightrightarrows \mathscr{P}(\mathcal{A})^\mathcal{X}$ is a set-valued function that maps Q-functions to the corresponding set of *greedy* policies. Mathematically, we have $\pi' \in \mathcal{G}(Q)$ if and only if

$$\pi'(a|x) > 0 \implies a \in \arg\max_{a'\in\mathcal{A}} Q(x, a') .$$

We will overload notation to allow us to pass policies directly to $\mathcal{G}$, writing $\mathcal{G}(\pi)$ for $\mathcal{G}(Q^\pi)$. A classical result underpinning policy iteration is that if $\pi' \in \mathcal{G}(\pi)$, then $Q^{\pi'} \geq Q^\pi$ element-wise, with equality iff π is optimal.

Barreto et al. (2017) propose *generalised policy improvement*, which provides a means of producing a policy that simultaneously improves over a *set* of base policies.

Generalised policy improvement. The generalised policy improvement (GPI) function $\mathcal{G}$ (overloading notation) takes as input a finite set of Q-functions $\{Q_1, \ldots, Q_n\}$, and returns $\mathcal{G}(\{Q_1, \ldots, Q_n\})$, the set of greedy policies with respect to this set, defined by $\pi' \in \mathcal{G}(\{Q_1, \ldots, Q_n\})$ if and only if

$$\pi'(a|x) > 0 \implies a \in \arg\max_{a'\in\mathcal{A}} \max_{i=1}^{n} Q_i(x, a') . \quad (1)$$

Proposition 2.1 (Barreto et al. 2017)**.** If $\pi' \in \mathcal{G}(\{Q^{\pi_1}, \ldots Q^{\pi_n}\})$, then $Q^{\pi'} \geq \max_{i=1}^n Q^{\pi_i}$ element-wise, and equality implies that π' is optimal.

2.2. Discounted visitation distributions and geometric horizon models

We begin by recalling a key concept in MDPs.

Definition 2.2. The collection of *discounted future state-visitation distributions* [1] μ^π_γ for a policy π and discount factor γ are indexed by initial state-action pairs $(x_0, a_0) \in \mathcal{X} \times \mathcal{A}$, and are defined by

$$\mu^\pi_\gamma(x|x_0, a_0) = (1 - \gamma) \sum_{k=0}^{\infty} \gamma^k \mathbb{P}^\pi_{x_0,a_0}(X_{k+1} = x) ,$$

A useful interpretation of these distributions is the following.

[1] We refer specifically to *future* state-visitation distributions to emphasise that the initial state x_0 does not contribute to the distribution.

Proposition 2.3. If $T \sim \text{Geometric}(1-\gamma)$, i.e.

$$\mathbb{P}(T = k) = \gamma^{k-1}(1-\gamma) \quad \text{for } k = 1, 2, \dots,$$

and is independent of the random trajectory $(X_t, A_t, R_t)_{t\geq 0}$ generated by π beginning at state-action pair (x, a), then the random state X_T is distributed according to $\mu_\gamma^\pi(\cdot|x, a)$.

This can also be used as a means of *defining* GHMs over more general state spaces $\mathcal{X}$. Janner et al. (2020) introduce γ-models as generative models of these distributions (in this paper, we will call these objects *geometric horizon models* (GHMs)), and propose to use these models for policy evaluation. The approach is based on well-known identities such as the following (Toussaint & Storkey, 2005; 2006).

Proposition 2.4. For any policy $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$, we have

$$Q_\gamma^\pi(x, a) = r(x, a) + \frac{\gamma}{1-\gamma}\mathbb{E}_{X'\sim\mu_\gamma^\pi(\cdot|x,a)}[r^\pi(X')], \quad (2)$$

where $r^\pi(x) = \sum_{a\in\mathcal{A}} r(x, a)\pi(a|x)$.

This result then naturally suggests a Monte Carlo estimator that can be used for policy evaluation, given a generative model of the distribution $\mu_\gamma^\pi(\cdot|x, a)$, and the reward function r. Specifically, if $X'_1, \dots, X'_n \overset{\text{i.i.d.}}{\sim} \mu_\gamma^\pi(\cdot|x, a)$, then

$$r(x, a) + \frac{1}{n}\sum_{i=1}^{n} \frac{\gamma}{1-\gamma} r^\pi(X'_i) \quad (3)$$

is an unbiased estimator for $Q_\gamma^\pi(x, a)$.

Note that this expression requires access to the reward function r. This function is known in many applications—often in robotics, for example—and when this is not the case it can be learned as a supervised learning problem. Throughout the paper we will assume that r is either given or has been learned. Note that Janner et al. (2020) implicitly use a reward function that depends solely on the *destination state* x' of the transition, leading to slightly different, less general expressions than those above.

2.3. Composing geometric horizon models for evaluation of Markov policies

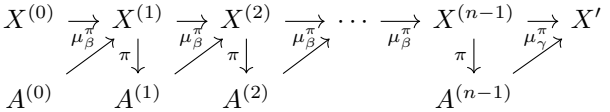
As Janner et al. (2020) note, a potential disadvantage of using the identity in Equation (2) as the basis for policy evaluation is that it requires learning the object μ_γ^π. When $\gamma \approx 1$, this distribution corresponds to predictions over long time-scales, and is therefore often more difficult to learn than more local predictions. A central observation of Janner et al. (2020) is that expressions such as those in Equation (2) can be re-expressed using a geometric horizon model corresponding to a smaller discount factor, $\beta < \gamma$, and composing this model with itself.

Proposition 2.5. (Janner et al. (2020)) For any policy $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$, $n \geq 1$, and $0 \leq \beta < \gamma$ an unbiased estimator of $Q_\gamma^\pi(x, a)$ is given by

$$r(x, a) + \frac{\gamma}{1-\gamma}\times \quad (4)$$
$$\left[\sum_{m=1}^{n-1} \frac{1-\gamma}{1-\beta}\left(\frac{\gamma-\beta}{1-\beta}\right)^{m-1} r^\pi(X^{(m)}) + \left(\frac{\gamma-\beta}{1-\beta}\right)^{n-1} r^\pi(X')\right],$$

where $X^{(m)} \sim \mu_\beta^\pi(\cdot|X^{(m-1)}, A^{(m-1)})$, $A^{(m)} \sim \pi(\cdot|X^{(m)})$, $(X^{(0)}, A^{(0)}) = (x, a)$, and $X' \sim \mu_\gamma^\pi(\cdot|X^{(n-1)}, A^{(n-1)})$.

According to Proposition 2.5, we can estimate $Q_\gamma^\pi(x, a)$ by sampling the collection of random variables $(X^{(0)}, A^{(0)}, X^{(1)}, \dots, X^{(n-1)}, A^{(n-1)}, X')$ in the proposition, summarised below:



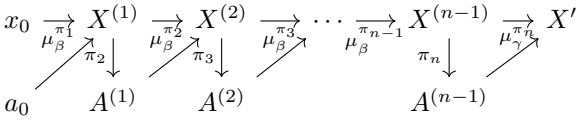
and then constructing the estimator in Equation (4), which the proposition guarantees to be unbiased for $Q_\gamma^\pi(x, a)$; independent estimators can be averaged in the usual manner to reduce variance.

The value of β impacts both the mechanics of the process above and the learning of the GHM μ_β^π itself. One extreme, $\beta = \gamma$, reverts to the single-sample estimator in Equation (3). The other extreme, $\beta = 0$, corresponds to estimating the Q-function using a single-step transition model In the first case, predictions are made over potentially long horizons, which alleviates the risk of compounding errors while estimating Q_γ^π. On the other hand, learning the GHM itself becomes more difficult—if we use bootstrapping to do so, as we will discuss shortly, errors might compound when learning μ_β^π. When $\beta = 0$ we observe the opposite trend. In practice, we expect an intermediate value of β to offer superior performance to the extremes of 0 and γ, since this will trade off errors incurred during the learning of the GHM and the estimation of the Q-function (Janner et al., 2020). The parameter n offers a trade-off between requiring more compositions of μ_β^π, and placing a higher weight on samples from the higher-discount, harder-to-train GHM μ_γ^π.

3. Composing models for non-Markov policy evaluation

Our first contribution is to extend the estimator appearing in Equation (4) by modifying the distribution of the random variables $(X^{(0)}, A^{(0)}, \dots, X^{(n-1)}, A^{(n-1)}, X')$ in Proposition 2.5, composing GHMs for *distinct* policies together. More precisely, let $(\pi_1, \dots, \pi_n)$ be a sequence of policies,

(x, a) an initial state action pair, and consider the distribution over state sequences specified by



If we form an expression analogous to Equation (4):

$$r(x) + \frac{\gamma}{1-\gamma} \times \left[\sum_{m=1}^{n-1} \frac{1-\gamma}{1-\beta} \left(\frac{\gamma-\beta}{1-\beta} \right)^{m-1} \bar{r}(X^{(m)}) + \left(\frac{\gamma-\beta}{1-\beta} \right)^{n-1} \bar{r}(X') \right] \tag{5}$$

for some suitable reward function $\bar{r}$, then following the intuition above, we should be able to interpret Expression (5) as unbiasedly estimating the value of a (non-Markov) policy that begins each trajectory by following π_1, before switching to each of $\pi_2, \ldots, \pi_{n-1}$, and eventually following π_n for the remainder of the episode. We first formalise this notion of behaviour, and then show that this intuition is correct.

Definition 3.1. Given a sequence $(\pi_1, \ldots, \pi_n)$ of (stationary Markov) policies and a switching probability $\alpha \in (0, 1]$, the corresponding *geometric switching policy* (GSP) ν is a non-Markov policy defined as follows. At the beginning of the episode, the Markov policy π_1 is followed for $T_1 \sim \text{Geometric}(\alpha)$ steps, at which point a switch is made to the Markov policy π_2. Once a switch from π_i to π_{i+1} is made, π_{i+1} is followed for $T_{i+1} \sim \text{Geometric}(\alpha)$ steps, at which point the next switching event occurs. Once π_n has been selected, it is followed for the remainder of the episode. We write $\pi_1 \overset{\alpha}{\to} \cdots \overset{\alpha}{\to} \pi_n$ to concisely refer to the GSP ν. We define $Q_\gamma^\nu : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ for a GSP ν by

$$Q_\gamma^\nu(x, a) = \mathbb{E}_{x,a}^\nu \Big[\sum_{t=0}^{\infty} \gamma^t R_t \Big];$$

precisely, the expectation on the right-hand side is over trajectories beginning at x, with actions generated by ν, with the first action overridden to be a.

We now show that the value of GSPs can be expressed as expectations of expressions such as that in Equation (5).

Theorem 3.2. Consider an MDP with reward function $r : \mathcal{X} \to \mathbb{R}$ and let $\nu = \pi_1 \overset{\alpha}{\to} \cdots \overset{\alpha}{\to} \pi_n$. With $\beta = \gamma(1-\alpha)$, the following is unbiased for $Q_\gamma^\nu(x, a)$:

$$r(x) + \frac{\gamma}{1-\gamma} \times \left[\sum_{m=1}^{n-1} \frac{1-\gamma}{1-\beta} \left(\frac{\gamma-\beta}{1-\beta} \right)^{m-1} r(X^{(m)}) + \left(\frac{\gamma-\beta}{1-\beta} \right)^{n-1} r(X') \right], \tag{6}$$

where $(X^{(0)}, A^{(0)}) = (x, a)$, $X^{(m)} \sim \mu_\beta^{\pi_m}(\cdot|X^{(m-1)}, A^{(m-1)})$, $A^{(m)} \sim \pi_{m+1}(\cdot|X^{(m)})$, $X' \sim \mu_\gamma^{\pi_n}(\cdot|X^{(n-1)}, A^{(n-1)})$.

We state the result in the case where the reward depends only on state for conciseness here; the slightly more complex formula that incorporates action dependence is given in Appendix F. The key insight is therefore that we can get an unbiased estimate of the Q-function Q_γ^ν associated with a geometric switching policy $\nu = \pi_1 \overset{\alpha}{\to} \cdots \overset{\alpha}{\to} \pi_n$ just using the models $\mu_\beta^{\pi_i}$ $(i = 1, \ldots, n-1)$ and $\mu_\gamma^{\pi_n}$ for the base policies. In particular, if we learn these models to evaluate the base policies, we can evaluate all GSPs arising from these base policies without any additional learning.

4. Generalised policy improvement with geometric switching policies

The ability to evaluate a large number of GSPs without additional learning opens up the possibility of using GPI to improve upon all these policies at once. Having established how to evaluate GSPs using GHMs for Markov base policies, the main contribution of this section is to extend GPI to allow for the inclusion of GSPs into the improvement set. Algorithmically, this is straightforward; the same definition in Equation (1) can be immediately applied to the Q-functions of geometric switching policies. Note that when applying GPI to the Q-functions of non-Markov GSPs, the returned greedy policies are still Markov; this desirable property allows us to embed the proposed approach into the usual RL loop for policy iteration, as discussed below.

What is not immediately clear is whether an improvement guarantee analogous to Proposition 2.1 still applies when using the Q-functions of geometric switching policies. It turns out, under certain conditions, we can recover such a result. To do so, we need a certain notion of 'closedness' amongst the policies to be improved upon.

Definition 4.1. A collection Π of GSPs is *suffix-closed* if whenever $n > 1$ and $\pi_1 \overset{\alpha}{\to} \cdots \overset{\alpha}{\to} \pi_n$ lies in Π, the suffix policy $\pi_2 \overset{\alpha}{\to} \cdots \overset{\alpha}{\to} \pi_n$ also lies in Π.

Theorem 4.2. Consider a suffix-closed collection of GSPs Π. Then if $\pi' \in \mathcal{G}(\Pi)$, we have

$$Q_\gamma^{\pi'}(x, a) \geq \max_{\nu \in \Pi} Q_\gamma^\nu(x, a), \quad \text{for all } (x, a) \in \mathcal{X} \times \mathcal{A}.$$

Further, if equality holds for all state-action pairs, then π' is optimal.

We refer to the procedure of computing $\pi' \in \mathcal{G}(\Pi)$ for a set of GSPs Π as *geometric generalised policy improvement* (GGPI). A rigorous proof of Theorem 4.2 is given in Appendix B, but for some intuition for the suffix-closed

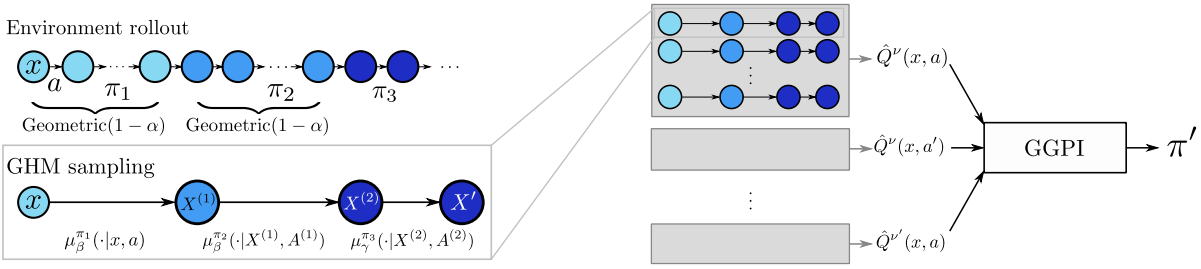


Figure 2. **Left:** A rollout of an example GSP $\nu = \pi_1 \xrightarrow{\alpha} \pi_2 \xrightarrow{\alpha} \pi_3$ in the environment, and the GHM sampling procedure that can be used to unbiasedly estimate the value of this policy via Equation (6). **Right:** The GGPI framework. Using the GHM sampling procedure, the action-values of ν and other GSPs are estimated, and fed into the GPI routine to obtain an improved policy π'.

condition, consider the two possibilities after a single step of executing $\nu = \pi_1 \xrightarrow{\alpha} \cdots \xrightarrow{\alpha} \pi_n$: either the first switch has not occurred (in which case it is as though we execute ν from scratch from the next time step), or the switch has occurred, in which case it is as though we execute the suffix policy $\nu' = \pi_2 \xrightarrow{\alpha} \cdots \xrightarrow{\alpha} \pi_n$ from the next time step. In fact, this observation yields a Bellman equation

$$Q^{\nu}_{\gamma}(x,a) = r(x,a) + \gamma \mathbb{E}_{\substack{X' \sim P(\cdot|x,a) \\ A_1 \sim \pi_1(\cdot|X') \\ A_2 \sim \pi_2(\cdot|X')}} \left[(1-\alpha) Q^{\nu}_{\gamma}(X', A_1) + \alpha Q^{\nu'}_{\gamma}(X', A_2) \right].$$

Thus, the suffix-closedness condition is a way of ensuring we can reason about both of these possibilities within the GGPI process. Perhaps surprisingly, the suffix-closedness condition in Theorem 4.2 really is necessary; some care needs to be taken when applying the ideas associated with GPI to non-Markov policies. A counterexample when the closure condition is removed is provided in Appendix D.1, along with several other examples.

In summary, GHM policy evaluation and GGPI allow us to derive Markov policies that improve over a wide range of GSPs, while only requiring learnt GHMs for the base Markov policies under consideration; see Figure 2.

5. Applications: transfer and policy iteration

We now detail two central applications of GHM evaluation and GGPI to reinforcement learning.

5.1. Transfer and zero-shot learning

In the transfer setting, we have a collection of known policies $\pi_1, \ldots, \pi_k$, and a reward function r for which we wish to find a good policy. The policies $\pi_{1:k}$ may have been obtained in a variety of ways: learnt by maximising other reward signals, exploration objectives, from imitation learning, etc. The reward function r is assumed to either be known (as is common in many robotics applications, for example), or learnt from data.

One simple approach to implementing GPI is to learn GHMs $(\mu^{\pi_i}_{\gamma})_{i=1}^{k}$, and use these in combination with the given reward function r to estimate $Q^{\pi_1}, \ldots, Q^{\pi_k}$, and perform generalised policy improvement over these Q-functions, as justified by Proposition 2.1.

With the concepts introduced above, we can improve on this by additionally learning GHMs $(\mu^{\pi_i}_{\beta})_{i=1}^{k}$, composing these to evaluate a collection of GSPs, and then using the GGPI procedure to improve over all such switching policies. A pseudocode summary of the approach is provided in Appendix D.3. Given a base set $\Pi = \{\pi_1, \ldots, \pi_k\}$ of Markov policies and a switching probability, we can define a variety of different sets of GSPs. A natural choice to consider, which we adopt in the experiments, is *the set of depth-m compositions*, $\Pi_m = \{\pi^{(1)} \xrightarrow{\alpha} \cdots \xrightarrow{\alpha} \pi^{(m)} : \pi^{(1)}, \ldots, \pi^{(m)} \in \Pi\}$, consisting of all GSPs that switch between exactly m (not necessarily distinct) base policies. We refer to GGPI on Π_m as *depth-m GGPI*. The following result shows that GGPI over Π_m guarantees an improvement, thanks to Theorem 4.2.

Proposition 5.1. Π_m is suffix-closed.

Example 5.2. Figure 3 illustrates an example experiment in the four-rooms environment (Sutton et al., 1999), with a single positive reward at the top-right-most cell, and $\gamma = 0.9$. We consider four base policies $\pi_{\mathrm{L}}, \pi_{\mathrm{D}}, \pi_{\mathrm{R}}, \pi_{\mathrm{U}}$ that always take the action left/down/right/up in each cell. GHMs are calculated for these policies with discounts γ and $\beta = 0.8$. By using GGPI over GSPs that make switches between these basic policies, the optimal policy can be recovered in almost all states of the environment, without any additional learning. Figure 3 illustrates in which states the optimal policy can be computed when using GPI over the four base policies (left), depth-2 GGPI (centre), and depth-3 GGPI (right). Depth-3 GGPI is able to compute the optimal action in the vast majority of environment states.

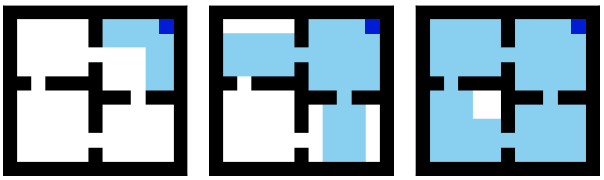


Figure 3. An illustration of the GGPI method on the four-rooms environment, with goal state indicated in dark blue. The plots illustrate which states (highlighted in light blue) each planning method is able to compute the optimal action for: GPI (left), depth-2 GGPI (centre), and depth-3 GGPI (right).

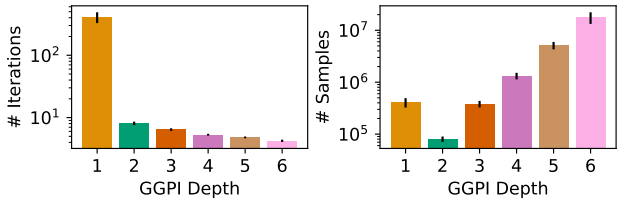


Figure 4. Comparison of GGPI with various planning depths applied to policy iteration on the standard 4-rooms domain, starting from a random policy. **Left:** Total number of iterations of policy iteration required; **Right:** Total number of GHM samples required, as a proxy to total computation performed. Error bars show bootstrapped 95% confidence intervals over 100 seeds.

5.2. Policy iteration

Policy iteration is a classical dynamic programming algorithm that computes a sequence of policies $(\pi_k)_{k\geq 0}$ through an iterative process of evaluation and greedy improvement, i.e. $\pi_k \in \mathcal{G}(Q^{\pi_{k-1}})$, which is guaranteed to reach the optimal policy in a finite number of iterations (for environments with finite state space). A natural question is whether we can use GPI to speed up this iterative process, by leveraging policies from previous iterations to compute even stronger improved policies, e.g. $\pi_k \in \mathcal{G}(\pi_{0:k-1})$. Unfortunately, when using standard GPI the answer to this question is "no"; since $Q^{\pi_{k-1}} \geq Q^{\pi_l}$ for $l < k-1$, GPI over $\pi_{0:k-1}$ reduces to standard policy improvement over π_{k-1}.

However, using GGPI may enable leveraging policies from older iterations to make larger improvement steps and converge to π^* more quickly, for example performing GGPI over all depth-m compositions over the set of previous policies $\{\pi_0, \dots \pi_{k-1}\}$. This has the advantage that any useful behaviour encoded by a prior policy that gets prematurely overwritten by subsequent iterations can still be leveraged to make larger improvement steps. Appendix D.2 contains algorithm pseudocode for applying GGPI to policy iteration, as well as an illustrative example.

Example 5.3. In Figure 4 we demonstrate the advantage of using GGPI for policy iteration in the classic four-rooms environment. The number of improvement steps decreases with the GGPI depth, indicating that the GGPI improvement step is able to compute stronger improved policies the more past knowledge it is allowed to leverage. Note however that although higher depths require fewer iterations, each improvement step is more computationally intensive for higher-depth GGPI; in this instance, depth-2 GGPI obtains the optimal trade-off between computational burden and strength of policy improvement, finding the optimal policy with the lowest total number of GHM samples. Here we solve the problem for a discount factor of $\gamma = 0.95$, switching probability $\alpha = 0.1$, compute perfect GHMs obtained using knowledge of the true environment dynamics, and evaluate each GSP ν using 1000 samples from the composed GHM μ^{ν}_{γ}.

6. Learning geometric horizon models

To use geometric horizon models for value estimation in practice, an important question is how to learn such models in the first place. An instructive starting point is to consider supervised learning with samples from μ^{π}_{β} (obtained by sampling X_T with $T \sim \text{Geometric}(1-\beta)$ by interacting with the environment using π, for example). The canonical cross-entropy loss can then be used to train a GHM μ, leading to the *cross-entropy Monte Carlo* (CEMC) loss:

$$\mathbb{E}_{X' \sim \mu^{\pi}_{\beta}(\cdot|x,a)}[-\log \mu(X'|x,a)]\,. \tag{7}$$

As with Monte Carlo learning in value-based RL, this approach is typically difficult to apply with off-policy data, incurring either bias, or potentially high variance updates from off-policy corrections (Precup et al., 2000). An alternative approach can be motivated by the observation that μ^{π}_{β} satisfies a Bellman equation involving composed models.

Definition 6.1 (Composed geometric horizon models)**.** Given two GHMs $\mu_1, \mu_2 \in \mathscr{P}(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$, and a policy $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$, the *composed model* $\mu_2 \otimes_\pi \mu_1 \in \mathscr{P}(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$ is the distribution of the random variable $X^{(2)}$, defined by

- $X^{(1)} \sim \mu_1(\cdot|x,a)$,
- $A^{(1)} \mid X^{(1)} \sim \pi(\cdot|X^{(1)})$,
- $X^{(2)} \mid (X^{(1)}, A^{(1)}) \sim \mu_2(\cdot|X^{(1)}, A^{(1)})$,

The distributions satisfy the relationship

$$(\mu_2 \otimes_\pi \mu_1)(y|x,a) = \sum_{x',a'} \mu_1(x'|x,a)\pi(a'|x')\mu_2(y|x',a')\,.$$

Proposition 6.2. Defining the Bellman operator $T^{\pi}_{\beta} : \mathscr{P}(\mathcal{X})^{\mathcal{X}\times\mathcal{A}} \to \mathscr{P}(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$ by

$$(T^{\pi}_{\beta}\mu)(x'|x,a) = (1-\beta)P(x'|x,a) + \beta(\mu \otimes_\pi P)(x'|x,a)\,,$$

then μ^{π}_{β} is the unique solution to $\mu = T^{\pi}_{\beta}\mu$.

This motivates a loss in which the Monte Carlo target in Equation (7) is replaced by the 'bootstrapped' distribution $T^{\pi}_{\beta}\mu$, leading to the *cross-entropy temporal-difference*

(CETD) loss, briefly mentioned by Janner et al. (2020):

$$\mathbb{E}_{X' \sim (T_\beta^\pi \bar{\mu})(\cdot|x,a)}[-\log \mu(X'|x,a)]\,, \tag{8}$$

where $\bar{\mu}$ denotes a stop-gradient on μ. Intuitively, $(T_\beta^\pi \mu)(\cdot|x,a)$ is the distribution obtained by sampling a next state $\tilde{x}$ from $P(\cdot|x,a)$, independently deciding whether to stop (with probability $1-\beta$) and output this state, or to sample an action $\tilde{a} \sim \pi(\cdot|\tilde{x})$ and instead return a sample from $\mu(\cdot|\tilde{x},\tilde{a})$. This also describes a method by which sample-based approximations to Equation (8) can be derived, leading to an algorithm that can be used at scale.

However, while sample-based minimisation of Equation (7) can be understood through stochastic gradient descent and convex optimisation theory, it is less clear that following sample-based gradients of the CETD loss in Equation (8) will lead to μ_β^π, due to the presence of bootstrapping. Next, we show that, under certain conditions, convergence to μ_β^π can be guaranteed, and additionally we show how the CETD loss can be applied at scale. Note that the prior approach to training GHMs at scale proposed by Janner et al. (2020) instead focused on a biased L^2 loss between log-densities; we show that CETD typically outperforms this approach in Appendix E.4, and note that it has the further advantage of not requiring access to single-step transition densities.

6.1. Convergence analysis of CETD

Consider a finite state space $\mathcal{X}$, and a tabular parametrisation of each distribution $\mu(\cdot|x,a)$ by a vector of logits $\phi(x,a) \in \mathbb{R}^{\mathcal{X}}$, so that $\mu(\cdot|x,a) = \text{softmax}(\phi(x,a))$. We show that with this parametrisation convergence to μ_β^π is obtained following CETD updates under mild conditions. To describe the precise algorithm we study, let $\phi_0 \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}\times\mathcal{X}}$ be the initial values of the logits in the parametrisation described above. We then consider generating a sequence of logits $(\phi_k)_{k\geq 0}$ and corresponding distributions $(\mu_k)_{k\geq 0}$ by iteratively applying *synchronous* CETD updates; at algorithm time k, for each state-action pair (x,a), we observe a transition (x,a,x') and perform the update:

$$\phi_{k+1}(x,a) = \phi_k(x,a) + \varepsilon_k \Big(\underbrace{(\hat{T}^\pi \mu_k)(x,a) - \mu_k(\cdot|x,a)}_{\text{stochastic cross-entropy gradient}} \Big)\,,$$

$$\text{with } (\hat{T}^\pi \mu_k)(x,a) = (1-\gamma) e_{x'} + \gamma e_{x''}\,, \tag{9}$$

where x'' is generated by sampling $a' \sim \pi(\cdot|x')$ and $x'' \sim \mu_k(\cdot|x',a')$, and where $e_y \in \mathbb{R}^{\mathcal{X}}$ is the one-hot vector at state y. Here, $(\varepsilon_k)_{k=0}^\infty$ is a sequence of step sizes.

Theorem 6.3. The CETD algorithm specified by the updates in Equation (9) produces sequences of distributions $(\mu_k)_{k\geq 0}$ such that $\mu_k(\cdot|x,a) \to \mu_\gamma^\pi(\cdot|x,a)$ for all (x,a), as long as $\sum_{k\geq 0} \varepsilon_k = \infty\,,\ \sum_{k\geq 0} \varepsilon_k^2 < \infty$.

The proof, relying on a discrete Lyapunov argument based on the Robbins-Siegmund theorem (Robbins & Siegmund, 1971), is in Appendix C with several illustrative examples.

6.2. Learning VAE-GHMs with CETD updates

We propose a novel scalable means of learning GHMs μ_β^π with VAEs (Kingma & Welling, 2014; Rezende et al., 2014) based on the CETD loss, and emphasise that these methods also apply equally well to learning GHM densities for MDPs with continuous state spaces, as is often of interest in deep RL. Specifically, we use a conditional VAE architecture $\mu_\theta(\cdot|x,a,z)$ (Sohn et al., 2015) with latent variable z, and approximate posterior $q_\psi(z|x,a,X')$. The CETD loss is then the negative log-marginal likelihood of the training state under the VAE model, leading to the following evidence lower-bound (ELBO) on the negative CETD loss:

$$\mathbb{E}_{X' \sim (T_\beta^\pi \bar{\mu}_\theta)(\cdot|x,a)} \Big[\mathbb{E}_{z \sim q_\psi(\cdot|x,a,X')} \big[\log \big(\tfrac{\mu_\theta(X'|x,a,z)}{q_\psi(z|x,a,X')} \big) \big] \Big]$$

which is then jointly optimised over θ and ψ via stochastic gradient descent with the reparametrisation trick (Kingma & Welling, 2014). Using VAEs offers several advantages, such as allowing low latent dimensionality in non-stochastic environments, and connection to the theoretically-justified CETD loss; see Section 7 for further commentary.

7. Deep reinforcement learning experiments

To understand how GSP evaluation using GHMs and GGPI perform at scale, we test them on a deep RL transfer task. Full details and further results are given in Appendix E.

Environment details. We consider a continuous control task inspired from the moving-target arena in Barreto et al. (2019), which we call *sparse-reward ant*. The agent is a quadrupedal "ant", and the environment observation is a 35-dimensional representation of the agent state, including position, velocity, and joint angles. The agent interacts with the environment via an 8-dimensional action space controlling the torque applied to its various joints. At the beginning of each episode, the agent's is initialised at rest at a location sampled from a uniform distribution over a square centred at the origin, and a target location is sampled from a smaller region surrounding the agent's initialisation (see Appendix E.1 for details). The reward is 1 for transitions that terminate in a region around the target and 0 elsewhere.

Experiment setup. Similar to Barreto et al. (2019), we first pretrain four base policies $\Pi = \{\pi_{\text{up}}, \pi_{\text{down}}, \pi_{\text{left}}, \pi_{\text{right}}\}$ that aim to move along each of the 4 directions. The policies are stochastic and implemented as a 2-layer MLP outputting the mean/variance of a Gaussian torque to be applied at each of the ant's 8 joints. These policies are pretrained using Abdolmaleki et al.'s (2018) MPO with the reward calculated based on the component of the ant's velocity in

the desired direction.

Next, we train GHMs for the base policies using the approach described in Section 6.2. The model is implemented as a standard conditional β-VAE (Higgins et al., 2016; Sohn et al., 2015) with a single latent dimension, as this is sufficient to model the probabilistic horizon in the deterministic environment. We train these GHMs from transitions using the CETD bound described in Section 6.2, for GHM β values of 0.8 and 0.9, and consider the task with discount factor $\gamma = 0.9$. This corresponds to performing GGPI for GSPs for a switching probability of $\alpha = 1/9$. Further details, observations, and recommendations are provided in Appendix E; for example, we found off-policy training of GHMs important to obtain sufficient state coverage.

Once the GHMs have been learned, the agent is evaluated on new episodes without additional learning. In each test episode, the agent must plan to optimise for a new revealed reward function r associated with the randomly-generated target region, leveraging the learnt GHMs for the set of base policies Π above. We consider two approaches: (i) GPI (Barreto et al., 2017) on Π using GHM evaluation (Janner et al., 2020), a natural baseline for this task (equivalent to depth-1 GGPI), and (ii) depth-2 GGPI (Section 5.1).

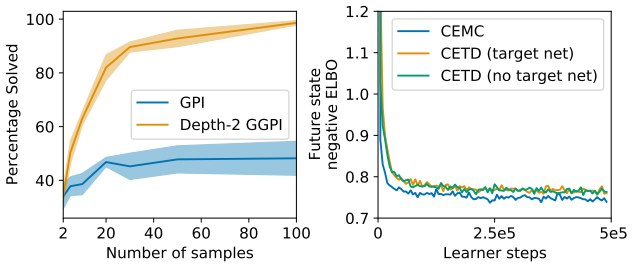


Figure 5. **Left:** Comparing GGPI at different depths m in terms of total episodes succeeded, for various sampling budgets. Agents are evaluated across 5 random seeds for GHM training and 100 test episodes with bootstrapped 90% confidence intervals. **Right:** Comparison of GHMs trained using various losses measured in terms of negative ELBO (lower is better) of samples from the true future state visitation distribution obtained by sampling states from on-policy trajectories, averaged over 5 random seeds.

Results. Figure 5 (left) shows the proportion of test episodes successfully solved by GPI and by depth-2 GGPI, varying the sample budget n_{samples} used to estimate each Q-value. We see that depth-2 GGPI outperforms GPI for $n_{\text{samples}} \geq 2$, eventually reaching a success rate close to 100%. In Table 1 we take a finer-grained look at the results when using 100 samples. We see that standard GPI manages to solve the task roughly 50% of the time, while depth-2 GGPI (where the agent is able to model changing directions) not only solves almost all the remaining goal locations but also almost always solves the task faster than standard GPI when both are capable of reaching the goal. Agent behaviour is visualised in Figure 6 and Appendix E.2.

Table 1. Results of comparing agent performance for GPI and depth-2 GGPI on the *sparse-reward ant* environment. Agents are evaluated across 5 random seeds for GHM training and 100 random environment and target initialisations.

Case	Frequency
Depth-2 GGPI succeeds, GPI fails	50.8 ± 5.7
Both succeed, depth-2 GGPI is faster	45.0 ± 7.0
Both fail	1.0 ± 0.9
Both succeed but GPI is faster	2.8 ± 1.2
GPI succeeds, depth-2 GGPI fails	0.4 ± 0.8

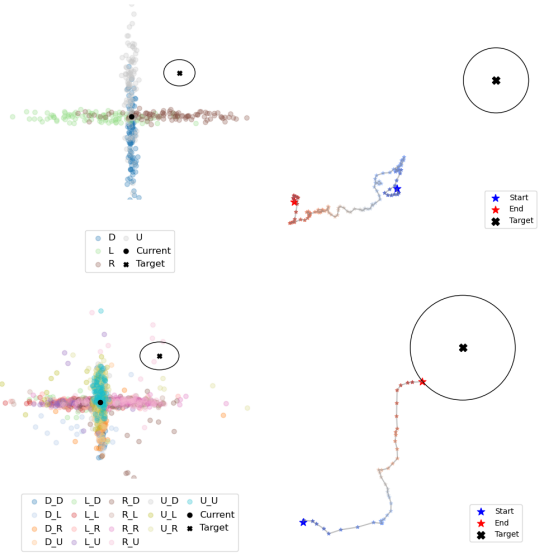


Figure 6. Visualisation of an agent performing GPI (top) and depth-2 GGPI (bottom) for the same test episode. GHM samples for use in GGPI at the first episode time step are shown on the left, while the entire episode trajectory coloured on a gradient from blue to red with time is shown on the right. We show the ant centre of mass, target, and reward signal boundary.

GHM training experiments. GHM training is an important component of the deep RL results above. We found combining VAEs with the CETD loss to work particularly well; Figure 5 (right) shows negative ELBO loss curves for the VAE-GHM of the policy π_{right} on the sparse-reward ant domain, with $\beta = 0.8$. The loss is similar to that of a VAE-GHM trained via supervised learning with the CEMC loss from Equation (7), and we found target networks unnecessary for stable training. We show in Appendix E.4 that this combination of VAEs with the CETD loss is remarkably stable compared to normalising flows using either CETD or the log-L^2 loss previously considered. Appendix E.3 compares GHMs with multi-step compositions of VAEs modelling single-step transitions, showing that the latter incur large

errors and thus are unsuitable for long-horizon planning. Finally, in Appendix E.5 we examine the performance of GGPI when using GHMs trained with varying sampling budgets, showing that even GHMs trained for only a few thousand steps whose loss has not yet converged are still useful and result in a strong improved policy.

8. Related work

This paper relates to a number of different areas in reinforcement learning; we describe the most closely related works below, with additional discussion in Appendix A.

In addition to the work of Janner et al. (2020) described above, there are several recent contributions studying the task of large-scale learning of discounted visitation distributions and related objects. Blier et al. (2021) propose several TD-based methods for learning parametric representations, including an approach based on low-rank approximations. Building on this work, Touati & Ollivier (2021) propose a compact representation of an MDP that in principle allows for the optimal policy associated with any reward function to be computed without planning, in practice relying on a low-dimensional approximation of the visitation distributions. Eysenbach et al. (2021) propose a classification-based approach based on contrastive learning; these works also note a close connection with the domain of goal-conditioned RL (Kaelbling, 1993; Schaul et al., 2015; Andrychowicz et al., 2017; Pong et al., 2018). These ideas go back to the *successor representation* (SR), introduced in the context of representation learning in finite-state MDPs by Dayan (1993), who also proposed a TD method for learning the SR; this has also been explored in combination with deep learning (Kulkarni et al., 2016b; Fujimoto et al., 2021).

Relatedly, modelling discounted visitation distributions for evaluation was proposed by Sutton (1995), who termed such objects β-models. These models were generalised by Precup et al. (1998a), who proposed multi-time models, which encompass both β-models and n-step models as special cases. More generally, there is a long-established practice of learning option models (Sutton et al., 1999; Precup et al., 1998b; Precup, 2000), and using such models in a compositional manner (Silver & Ciosek, 2012). A central difference between these option models and this work is that the use of geometric switching times (or, in the language of options, constant termination probabilities) means we do not need to model accumulated return or the taken executing each base policy, making applications to transfer possible. In this regard, the approach of this paper may be viewed as a generative approach to learning a certain class of universal option models (Yao et al., 2014), which also disentangle reward and transition structure; constant termination probabilities facilitate sample-based composition of such models.

Our application to transfer learning in RL is motivated by *successor features* and generalised policy improvement, introduced by Barreto et al. (2017; 2020). Subsequent work in this direction includes algorithmic innovations in combination with deep learning (Barreto et al., 2018; Borsa et al., 2019), reward-free learning (Grimm et al., 2019; Hansen et al., 2020), and addressing questions concerning the influence of the policy set on improvements in GPI (Zahavy et al., 2021; Alver & Precup, 2022; Lehnert & Littman, 2020; Nemecek & Parr, 2021). A notable approach that also interpolates between greedy improvement and computation of optimal policies is multi-step policy improvement (Efroni et al., 2018a;b; Tomar et al., 2020).

9. Conclusions, limitations, and future work

In this paper, we have proposed using geometric horizon models for the evaluation of non-Markov geometric switching policies, and for doing policy improvement over collections of such policies. We have shown that this pair of techniques can be applied to both transfer and policy iteration, extending existing techniques based on successor features and generalised policy improvement. We have also demonstrated that it is possible to combine these ideas with deep learning architectures to arrive at novel approaches to deep RL, and in the course have additionally provided theoretical analyses of these methods.

We foresee several key considerations in further extending the applicability of this approach. First, the method relies on constructing models over environment state; as with many other model-based methods, a key question is how to learn such models efficiently in high-dimensional settings. Additionally, the use of geometric switching times in GSPs is key to decoupling rewards from learnt models, but limits the expressivity of the non-Markov policies considered; can this restriction be lifted? In addition to these questions, there are several natural directions for future work. These include further development of theoretical convergence analyses for learning GHMs and improving over GSPs, as well as further developing combinations of these techniques with deep learning. We believe that combining GGPI with recent advances in adaptive planning techniques is a particularly promising direction for further work.

Acknowledgements

We thank the anonymous reviewers for useful comments and suggestions, and gratefully acknowledge support from our colleagues in the course of this work. Thanks in particular to Mohammad Gheshlaghi Azar, Gheorghe Comanici, Hamza Merzic, Doina Precup, Yunhao Tang, and to Théophane Weber for detailed feedback on an earlier draft.

References

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. Maximum a posteriori policy optimisation. In *Proceedings of the International Conference on Learning Representations*, 2018.

Agrawal, S. and Jia, R. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, 2017.

Alver, S. and Precup, D. Constructing a good behavior basis for transfer using generalized policy updates. In *Proceedings of the International Conference on Learning Representations*, 2022.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, 2017.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv*, 2016.

Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Fantacci, C., Godwin, J., Jones, C., Hennigan, T., Hessel, M., Kapturowski, S., Keck, T., Kemaev, I., King, M., Martens, L., Mikulik, V., Norman, T., Quan, J., Papamakarios, G., Ring, R., Ruiz, F., Sanchez, A., Schneider, R., Sezener, E., Spencer, S., Srinivasan, S., Stokowiec, W., and Viola, F. The DeepMind JAX Ecosystem, 2020.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., Van Hasselt, H., and Silver, D. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Barreto, A., Borsa, D., Quan, J., Schaul, T., Silver, D., Hessel, M., Mankowitz, D., Zidek, A., and Munos, R. Transfer in deep reinforcement learning using successor features and generalised policy improvement. In *Proceedings of the International Conference on Machine Learning*, 2018.

Barreto, A., Borsa, D., Hou, S., Comanici, G., Aygün, E., Hamel, P., Toyama, D., Hunt, J. J., Mourad, S., Silver, D., and Precup, D. The option keyboard: Combining skills in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2019.

Barreto, A., Hou, S., Borsa, D., Silver, D., and Precup, D. Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences*, 117(48):30079–30087, 2020. ISSN 0027-8424.

Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Blier, L., Tallec, C., and Ollivier, Y. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv*, 2021.

Borsa, D., Barreto, A., Quan, J., Mankowitz, D. J., van Hasselt, H., Munos, R., Silver, D., and Schaul, T. Universal successor features approximators. In *Proceedings of the International Conference on Learning Representations*, 2019.

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

Brunskill, E. and Li, L. PAC-inspired option discovery in lifelong reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2014.

Buşoniu, L. and Munos, R. Optimistic planning for Markov decision processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2012.

Buşoniu, L., Munos, R., and Babuška, R. A survey of optimistic planning in Markov decision processes. In Lewis, F. L. and Liu, D. (eds.), *Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control*, chapter 22, pp. 494–516. John Wiley & Sons, 2012.

Dabney, W., Ostrovski, G., and Barreto, A. Temporally-extended ϵ-greedy exploration. In *Proceedings of the International Conference on Learning Representations*, 2021.

Dalal, G., Hallak, A., Dalton, S., Frosio, I., Mannor, S., and Chechik, G. Improve agents without retraining: Parallel tree search with off-policy correction. In *Advances in Neural Information Processing Systems*, 2021.

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

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *Proceedings of the International Conference on Learning Representations*, 2017.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In *Advances in Neural Information Processing Systems*, 2019.

Efroni, Y., Dalal, G., Scherrer, B., and Mannor, S. Beyond the one step greedy approach in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2018a.

Efroni, Y., Dalal, G., Scherrer, B., and Mannor, S. Multiple-step greedy policies in online and approximate reinforcement learning. In *Advances in Neural Information Processing Systems*, 2018b.

Efroni, Y., Dalal, G., Scherrer, B., and Mannor, S. How to combine tree-search methods in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Eysenbach, B., Salakhutdinov, R., and Levine, S. C-learning: Learning to achieve goals via recursive classification. In *Proceedings of the International Conference on Learning Representations*, 2021.

Feldman, Z. and Domshlak, C. Monte-Carlo planning: Theoretically fast convergence meets practical efficiency. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2013.

Feldman, Z. and Domshlak, C. On MABs and separation of concerns in Monte-Carlo planning for MDPs. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 2014a.

Feldman, Z. and Domshlak, C. Simple regret optimization in online planning for Markov decision processes. *Journal of Artificial Intelligence Research*, 51:165–205, 2014b.

Frobenius, G. Über Matrizen aus nicht negativen Elementen. *Sitzungsberichte Akad. Wiss. Berlin*, 1912.

Fujimoto, S., Meger, D., and Precup, D. A deep reinforcement learning approach to marginalized importance sampling with the successor representation. In *Proceedings of the International Conference on Machine Learning*, 2021.

Grimm, C., Higgins, I., Barreto, A., Teplyashin, D., Wulfmeier, M., Hertweck, T., Hadsell, R., and Singh, S. Disentangled cumulants help successor representations transfer new tasks. *arXiv*, 2019.

Hansen, S., Dabney, W., Barreto, A., de Wiele, T. V., Warde-Farley, D., and Mnih, V. Fast task inference with variational intrinsic successor features. In *Proceedings of the International Conference on Learning Representations*, 2020.

Harb, J., Bacon, P.-L., Klissarov, M., and Precup, D. When waiting is not an option: Learning options with a deliberation cost. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

Harutyunyan, A., Dabney, W., Borsa, D., Heess, N., Munos, R., and Precup, D. The termination critic. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations*, 2016.

Hunt, J., Barreto, A., Lillicrap, T., and Heess, N. Composing entropic policies using divergence correction. In *Proceedings of the International Conference on Machine Learning*, 2019.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

Janner, M., Mordatch, I., and Levine, S. Gamma-models: Generative temporal difference learning for infinite-horizon prediction. In *Advances in Neural Information Processing Systems*, 2020.

Kaelbling, L. P. Learning to achieve goals. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1993.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations*, 2015.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*, 2014.

Kulkarni, T. D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 2016a.

Kulkarni, T. D., Saeedi, A., Gautam, S., and Gershman, S. J. Deep successor reinforcement learning. *arXiv*, 2016b.

Kushner, H. and Yin, G. G. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

Lehnert, L. and Littman, M. L. Successor features combine elements of model-free and model-based reinforcement learning. *Journal of Machine Learning Research*, 21: 196:1–196:53, 2020.

Lesner, B. and Scherrer, B. Non-stationary approximate modified policy iteration. In *Proceedings of the International Conference on Machine Learning*, 2015.

Machado, M. C., Bellemare, M. G., and Bowling, M. A Laplacian framework for option discovery in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2017.

McGovern, A. and Barto, A. G. Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the International Conference on Machine Learning*, 2001.

Menache, I., Mannor, S., and Shimkin, N. Q-cut — dynamic discovery of sub-goals in reinforcement learning. In *Proceedings of the European Conference on Machine Learning*, 2002.

Meyn, S. *Control Systems and Reinforcement Learning*. Cambridge University Press, 2022.

Munos, R. From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1): 1–129, 2014.

Nemecek, M. and Parr, R. Policy caches with successor features. In *Proceedings of the International Conference on Machine Learning*, 2021.

Norris, J. R. *Markov Chains*. Cambridge University Press, 1998.

Osband, I., Russo, D., and Van Roy, B. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, 2013.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems*, 2016.

Perron, O. Zur Theorie der Matrices. *Mathematische Annalen*, 64(2):248–263, 1907.

Pong, V., Gu, S., Dalal, M., and Levine, S. Temporal difference models: Model-free deep RL for model-based control. In *Proceedings of the International Conference on Learning Representations*, 2018.

Precup, D. *Temporal abstraction in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 2000.

Precup, D., Sutton, R. S., and Singh, S. Multi-time models for temporally abstract planning. In *Advances in Neural Information Processing Systems*, 1998a.

Precup, D., Sutton, R. S., and Singh, S. Theoretical results on reinforcement learning with temporally abstract options. In *Proceedings of the European Conference on Machine Learning*, 1998b.

Precup, D., Sutton, R. S., and Singh, S. P. Eligibility traces for off-policy policy evaluation. In *Proceedings of the International Conference on Machine Learning*, 2000.

Puterman, M. L. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *Proceedings of the International Conference on Machine Learning*, 2015.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning*, 2014.

Robbins, H. and Siegmund, D. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics*, pp. 233–257. Elsevier, 1971.

Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., Wen, Z., et al. A tutorial on Thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *Proceedings of the International Conference on Machine Learning*, 2015.

Scherrer, B. and Lesner, B. On the use of non-stationary policies for stationary infinite-horizon Markov decision processes. In *Advances in Neural Information Processing Systems*, 2012.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations*, 2016.

Seneta, E. *Non-Negative Matrices and Markov Chains*. Springer Science & Business Media, 2006.

Silver, D. and Ciosek, K. Compositional planning using optimal option models. In *Proceedings of the International Conference on Machine Learning*, 2012.

Şimşek, Ö. and Barto, A. G. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2004.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, 2015.

Strens, M. A Bayesian framework for reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2000.

Sutton, R. S. TD models: Modeling the world at a mixture of time scales. In *Proceedings of the International Conference on Machine Learning*, 1995.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

Szepesvári, Cs. *Algorithms for Reinforcement Learning*. Morgan & Claypool, 2010.

Szörényi, B., Kedenburg, G., and Munos, R. Optimistic planning in Markov decision processes using a generative model. In *Advances in Neural Information Processing Systems*, 2014.

Todorov, E., Erez, T., and Tassa, Y. MuJoCo: A physics engine for model-based control. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2012.

Tomar, M., Efroni, Y., and Ghavamzadeh, M. Multi-step greedy reinforcement learning algorithms. In *Proceedings of the International Conference on Machine Learning*, 2020.

Touati, A. and Ollivier, Y. Learning one representation to optimize all rewards. In *Advances in Neural Information Processing Systems*, 2021.

Toussaint, M. and Storkey, A. Probabilistic inference for computing optimal policies in MDPs. In *NIPS Workshop on Game Theory, Machine Learning and Reasoning under Uncertainty*, 2005.

Toussaint, M. and Storkey, A. Probabilistic inference for solving discrete and continuous state Markov decision processes. In *Proceedings of the International Conference on Machine Learning*, 2006.

Wulfmeier, M., Rao, D., Hafner, R., Lampe, T., Abdolmaleki, A., Hertweck, T., Neunert, M., Tirumala, D., Siegel, N., Heess, N., and Riedmiller, M. Data-efficient hindsight off-policy option learning. In *Proceedings of the International Conference on Machine Learning*, 2021.

Yao, H., Szepesvári, Cs., Sutton, R. S., Modayil, J., and Bhatnagar, S. Universal option models. In *Advances in Neural Information Processing Systems*, 2014.

Zahavy, T., Barreto, A., Mankowitz, D. J., Hou, S., O'Donoghue, B., Kemaev, I., and Singh, S. Discovering a set of policies for the worst case reward. In *Proceedings of the International Conference on Learning Representations*, 2021.

Generalised Policy Improvement with Geometric Policy Composition: Appendices

We briefly summarise the contents of the appendices here for convenience.

- Appendix A provides further discussion of related work, as well as additional context for geometric horizon models and their precise connection with concepts such as the successor representation.
- Appendix B provides proofs for the results in the main paper concerning evaluating and improving over geometric switching policies.
- Appendix C provides a proof of the CETD convergence result presented in the main paper, and illustrations of an implementation of the algorithm.
- Appendix D provides further examples and illustrations to complement the findings of the main paper, including counterexamples illustrating the necessity of several conditions in our results and algorithm pseudocode for application of GGPI to transfer and policy iteration.
- Appendix E provides further experimental details and results.
- Appendix F provides a generalisation of the core policy evaluation result in the main paper.

A. Additional background, related work and context

A.1. Related work

Below, we discuss connections of this work to several sub-fields of reinforcement learning.

Other generalisations of greedy policy improvement. Our proposed approach is one way of interpolating between greedy improvement and full planning. Efroni et al. (2018a;b); Tomar et al. (2020) consider multi-step improvement as a different means of achieving such a trade-off, both analysing the approach theoretically, and empirically investigating the approach in combination with deep reinforcement learning. More generally, recent developments in Monte Carlo tree search and related ideas in planning (Buşoniu & Munos, 2012; Buşoniu et al., 2012; Feldman & Domshlak, 2013; 2014b; Munos, 2014; Szörényi et al., 2014; Feldman & Domshlak, 2014a; Efroni et al., 2018a; 2019; Dalal et al., 2021) can all be viewed as sitting between greedy improvement and computation of the exact optimal policy, and have the potential to be profitably combined with GHMs and GSPs.

Option models. Modelling discounted visitation distributions was proposed by Sutton (1995), who termed them β-models. These models were generalised by Precup et al. (1998a), who proposed multi-time models, which encompass both β-models and n-step models as special cases. More generally, there is a long-established practice of learning option models (Sutton et al., 1999; Precup et al., 1998b; Precup, 2000), and using such models in a compositional manner (Silver & Ciosek, 2012). A central difference between option models and this work is that the use of geometric switching times (or in the language of options, constant termination probabilities) means we do not need to model accumulated return obtained by each base policy, or the time taken executing each base policy, making applications to transfer possible. In this regard, the approach of this paper is related to universal option models (Yao et al., 2014), which also disentangle reward and transition structure; constant termination probabilities more easily facilitate sample-based composition of such models. Although orthogonal to the direction of this work, the problem of *option discovery* is central to hierarchical RL (McGovern & Barto, 2001; Menache et al., 2002; Şimşek & Barto, 2004; Brunskill & Li, 2014; Kulkarni et al., 2016a; Machado et al., 2017; Harb et al., 2018; Harutyunyan et al., 2019; Wulfmeier et al., 2021), and is clearly relevant here too, essentially posing the question of where the base policies supplied to GGPI should come from.

The successor representation and visitation distributions. Discounted visitation distributions are closely related to the successor representation (SR), introduced by Dayan (1993), who also proposed a temporal-difference method for learning the SR. As discussed above, Janner et al. (2020) introduce several methods for learning approximate discounted visitations on continuous state spaces, among other contributions. Several other recent works also target this problem. Blier et al. (2021) propose several methods for learning parametric approximations to discounted visitation distributions, including an approach based on low-rank approximations. Building on this work, Touati & Ollivier (2021) propose a compact representation of an MDP that in principle allows for the optimal policy associated with any reward function to be computed without planning, in practice relying on a low-dimensional approximation of the visitation distributions. Eysenbach et al. (2021) propose an approach based on contrastive learning; these works also note a close connection with the domain of goal-conditioned RL (Kaelbling, 1993; Schaul et al., 2015; Andrychowicz et al., 2017; Pong et al., 2018).

Successor features and GPI. Barreto et al. (2017) introduced successor features, a generalisation of the successor represen-

tation, and GPI, in the context of transfer; later Barreto et al. (2018) discussed the practicalities involved in combining the approach with deep learning. The same conceptual machinery was then used by Barreto et al. (2019) to promote temporal abstraction in RL. Borsa et al. (2019) introduced a generalised form of successor features that has a representation of a policy as one of their inputs, thus allowing generalisation along the space of policies. Hunt et al. (2019) extended successor features to entropy-regularized RL and addressed some of the challenges involved in applying GPI to continuous action spaces. Grimm et al. (2019) and Hansen et al. (2020) propose approaches that allow the features used in successor features to be learned from data in the absence of a reward signal. Zahavy et al. (2021) and Alver & Precup (2022) studied the problem of how to construct a good set of policies to be used with GPI. Lehnert & Littman (2020) showed how successor features can be seen as a link between model-free and model-based RL. Nemecek & Parr (2021) studied a related problem: given a set of successor features and a reward function, they showed how to estimate the performance of the associated GPI policy and use this estimate to decide whether to add new successor features to the set. Recently, Barreto et al. (2020) presented a comprehensive account of GPI and successor features in which the latter are cast as a special case of a more general concept called *generalised policy evaluation* (GPE). We believe GHMs can be understood as an alternative form of GPE.

Non-Markov policies. Non-Markov/homogeneous policies are used in several other sub-fields of reinforcement learning in MDPs. Scherrer & Lesner (2012); Lesner & Scherrer (2015) consider approximate value iteration, policy iteration, and modified policy iteration algorithms, proposing the use of non-homogeneous policies that repeatedly cycle through a sequence of recent greedy Markov policies, and showing that such policies obtain improved performance bounds. In contrast, GGPI always produces a Markov policy, but one which improves upon non-Markov policies. Non-Markov policies are also commonly-encountered in exploration, for example via action repetition (Dabney et al., 2021), and Thompson sampling and its approximations and variations (Strens, 2000; Osband et al., 2013; 2016; Agrawal & Jia, 2017; Russo et al., 2018).

A.2. Successor features, the successor representation, and geometric horizon models

We provide some additional discussion regarding the relationship between the successor representation, successor features, and geometric horizon models in the case of finite state spaces $\mathcal{X}$. For ease of comparison, we phrase all three concepts in terms of variants that condition on an initial state-action pair, although the successor representation was originally introduced as a state-indexed quantity.

Dayan (1993) introduced the successor representation in reinforcement learning. In the context of discounted MDPs, the definition is as follows.

Definition A.1. For a given policy $\pi : \mathcal{X} \to \mathscr{P}(\mathcal{A})$, the corresponding *successor representation* of a state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$ is the vector

$$\lambda^\pi(x, a) = \mathbb{E}^\pi_{x,a}\Big[\sum_{k=0}^{\infty} \gamma^k e_{X_k}\Big] \in \mathbb{R}^{\mathcal{X}},$$

where $e_{x'} \in \mathbb{R}^{\mathcal{X}}$ is the one-hot vector for the coordinate x'.

We can view $\lambda^\pi(x, a)$ as an unnormalised probability distribution; scaling by a factor of $1-\gamma$ yields a probability distribution that corresponds to sampling a time $T \sim \text{Geometric}(1-\gamma)$, and then sampling $T-1$ transition steps in the environment under π, initialised at the state-action pair (x, a).

Barreto et al. (2017) introduced successor features as a generalisation of the successor representation.

Definition A.2. Consider a base feature map $\phi : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}^K$. For a given policy $\pi : \mathcal{X} \to \mathscr{P}(\mathcal{A})$, the corresponding vector of *successor features* of a state $x \in \mathcal{X}$ is the vector

$$\psi^\pi(x, a) = \mathbb{E}^\pi_{x,a}\Big[\sum_{t\geq 0} \gamma^t \phi(X_t, A_t, X_{t+1})\Big] \in \mathbb{R}^K.$$

The successor representation is subsumed as a special case of successor features when $\phi(x, a, x') \in \mathbb{R}^{\mathcal{X}}$ is taken to be the basis vector for state x. The following result relates the discounted future state-visitation distributions of Definition 2.2 with successor features.

Proposition A.3. The discounted future state-visitation distribution μ^π_γ is an instance of successor features, with the base feature map $\phi(x, a, x') = (1-\gamma)e_{x'} \in \mathbb{R}^{\mathcal{X}}$, where $e_{x'}$ is the one-hot vector for the coordinate x'.

Proof. We directly calculate the x' coordinate of $\psi^\pi(x,a)$ as:

$$\begin{aligned}\mathbb{E}^\pi_{x,a}\Big[\sum_{t\geq 0}\gamma^t \mathbb{1}\{X_{t+1}=x'\}\Big] &= \mathbb{E}^\pi_{x,a}\Big[\sum_{t\geq 0}\gamma^t(1-\gamma)\mathbb{1}\{X_{t+1}=x'\}\Big]\\ &\overset{(a)}{=} \sum_{t\geq 0}\gamma^t(1-\gamma)\mathbb{E}^\pi_{x,a}\Big[\mathbb{1}\{X_{t+1}=x'\}\Big]\\ &= (1-\gamma)\sum_{t\geq 0}\gamma^t \mathbb{P}^\pi_{x,a}(X_{t+1}=x')\\ &= \mu^\pi_\gamma(x'|x,a)\,,\end{aligned}$$

where the swapping of summation and expectation in (a) is justified by the dominated convergence theorem, since the integrand is bounded. □

Proposition A.3 sheds light on the relationship between successor features and GHMs in the case of a finite state space $\mathcal{X}$. When using the features $\phi(x,a,x') = (1-\gamma)e_{x'}$, the successor features of policy π become the γ-discounted state-visitation distribution of π—that is, $\psi^\pi(x,a) = \mu^\pi_\gamma(\cdot|x,a)$; the corresponding GHM is a generative model of this distribution.

B. Proofs relating to geometric horizon models and generalised policy improvement

B.1. Proofs of results in Section 2.2

Proposition 2.3. If $T \sim \text{Geometric}(1-\gamma)$, i.e.

$$\mathbb{P}(T=k) = \gamma^{k-1}(1-\gamma) \quad \text{for } k = 1,2,\dots\,,$$

and is independent of the random trajectory $(X_t, A_t, R_t)_{t\geq 0}$ generated by π beginning at state-action pair (x,a), then the random state X_T is distributed according to $\mu^\pi_\gamma(\cdot|x,a)$.

Proof. We have

$$\begin{aligned}\mathbb{P}^\pi_{x,a}(X_T=x') &= \mathbb{E}[\mathbb{P}^\pi_{x,a}(X_T=x' \mid T)]\\ &= \sum_{k=1}^\infty \mathbb{P}(T=k)\mathbb{P}^\pi_{x,a}(X_k=x' \mid T=k)\\ &= \sum_{k=1}^\infty (1-\gamma)\gamma^{t-1}\mathbb{P}^\pi_{x,a}(X_k=x')\\ &= \mu^\pi_\gamma(x'|x,a)\,,\end{aligned}$$

as required. □

Proposition 2.4. For any policy $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$, we have

$$Q^\pi_\gamma(x,a) = r(x,a) + \frac{\gamma}{1-\gamma}\mathbb{E}_{X'\sim\mu^\pi_\gamma(\cdot|x,a)}[r^\pi(X')]\,, \tag{2}$$

where $r^\pi(x) = \sum_{a\in\mathcal{A}} r(x,a)\pi(a|x)$.

Proof. We have

$$\begin{aligned}
Q^\pi_\gamma(x,a) &= \mathbb{E}^\pi_{x,a}\left[\sum_{t=0}^{\infty}\gamma^t R_t\right] \\
&= \mathbb{E}^\pi_{x,a}[R_0] + \mathbb{E}^\pi_{x,a}\Big[\sum_{t=1}^{\infty}\gamma^t R_t\Big] \\
&= r(x,a) + \gamma\sum_{t=1}^{\infty}\gamma^{t-1}\sum_{x'\in\mathcal{X}}\mathbb{P}^\pi_{x,a}(X_t = x')r^\pi(x') \\
&\overset{(a)}{=} r(x,a) + \gamma\sum_{x'\in\mathcal{X}}\sum_{t=0}^{\infty}\gamma^t\mathbb{P}^\pi_{x,a}(X_{t+1} = x')r^\pi(x') \\
&= r(x,a) + \gamma(1-\gamma)^{-1}\sum_{x'\in\mathcal{X}}\mu^\pi_\gamma(x'|x,a)r^\pi(x') \\
&= r(x,a) + \gamma(1-\gamma)^{-1}\mathbb{E}_{X'\sim\mu^\pi_\gamma(\cdot|x,a)}[r^\pi(X')]\,.
\end{aligned}$$

as required. The switching of the order of summation at (a) can be justified, for example, by noting that the double-sum is absolutely convergent:

$$\sum_{t=1}^{\infty}\sum_{x'\in\mathcal{X}}\left|\gamma^{t-1}\mathbb{P}^\pi_{x,a}(X_t = x')r^\pi(x')\right| \le \sum_{t=1}^{\infty}\gamma^{t-1}R^\pi_{\max} = R^\pi_{\max}(1-\gamma)^{-1} < \infty\,.$$

where $R^\pi_{\max} = \max_x |r^\pi(x)| < \infty$, as $|\mathcal{X}|$ is finite. □

B.2. Proof of result from Section 2.3

Below, we re-derive a result essentially equivalent to Theorem 2 of Janner et al. (2020), stated as Proposition 2.5 in our main paper, with a slightly different proof technique. The central idea is to develop a different way of sampling the random variable X_T appearing in Proposition 2.3, using the following results.

Lemma B.1. Let $(T_i)_{i=1}^{\infty} \overset{\text{i.i.d.}}{\sim} \text{Geometric}(1-\beta)$, and independently, $N \sim \text{Geometric}({}^{1-\gamma}/_{1-\beta})$. Then the random sum $\sum_{i=1}^{N} T_i$ has distribution $\text{Geometric}(1-\gamma)$.

Lemma B.2. Let $(T_i)_{i=1}^{n-1} \overset{\text{i.i.d.}}{\sim} \text{Geometric}(1-\beta)$, and independently, $T' \sim \text{Geometric}(1-\gamma)$, and N' a random variable taking variables in $\{1,\dots,n\}$, with probabilities

$$\mathbb{P}(N'=m) = \frac{1-\gamma}{1-\beta}\left(\frac{\gamma-\beta}{1-\beta}\right)^{m-1} \text{ for } m = 1,\dots,n-1\,, \text{ and } \mathbb{P}(N'=n) = \left(\frac{\gamma-\beta}{1-\beta}\right)^{n-1}.$$

Then the random sum

$$\sum_{i=1}^{\min(N',n-1)} T_i + \mathbb{1}\{N'=n\}T'$$

has distribution $\text{Geometric}(1-\gamma)$.

Proposition B.3. If we define a sequence of states and actions $(X^{(n)}, A^{(n)})_{n\ge 0}$ inductively by $(X^{(0)}, A^{(0)}) = (x_0, a_0)$, $X^{(n+1)} \sim \mu^\pi_\beta(\cdot|X^{(n)}, A^{(n)})$, $A^{(n+1)} \sim \pi(\cdot|X^{(n+1)})$, then $X^{(n)} \overset{\mathcal{D}}{=} X_{\sum_{i=1}^{n} T_i}$.

We also note that using different distributional identities for the random variable T leads to variants of the result given in Proposition 2.5. For example, directly using the distributional identity in Lemma B.1 can be used to establish a version of Theorem 1 of Janner et al. (2020) using exactly the same proof technique as for Proposition 2.5.

Proof of Lemma B.1. This is a classical result from elementary probability theory. We work with probability generating functions. The probability generating function of a random variable Z taking values in $\mathbb{N}$ is defined as the function $G_Z(s) = \mathbb{E}[s^Z] = \sum_{k=1}^{\infty}\mathbb{P}(Z=k)s^k$, and clearly characterises the distribution of Z.

A standard calculation shows that for $T \sim \text{Geometric}(1-\gamma)$, we have

$$G_T(s) = \frac{s(1-\gamma)}{1-s\gamma}\,, \text{ for } |s| < \gamma^{-1}\,.$$

We also have the following standard relationship for the PGF of a random sum of i.i.d. terms:

$$G_{\sum_{i=1}^N T_i}(s) = \mathbb{E}[s^{\sum_{i=1}^N T_i}] = \mathbb{E}[\mathbb{E}[s^{\sum_{i=1}^N T_i} \mid N]] = \sum_{n=1}^{\infty} \mathbb{P}(N=n)\mathbb{E}[s^{\sum_{i=1}^n T_i}] = \sum_{n=1}^{\infty} \mathbb{P}(N=n) G_{T_1}(s)^n = G_N(G_{T_1}(s))\,.$$

Since both N and T_1 have geometric distributions, we can directly calculate

$$G_N(G_{T_1}(s)) = G_N\left(\frac{s(1-\beta)}{1-s\beta}\right) = \frac{\frac{s(1-\beta)}{1-s\beta}\left(\frac{1-\gamma}{1-\beta}\right)}{1-\frac{s(1-\beta)}{1-s\beta}\left(1-\frac{1-\gamma}{1-\beta}\right)} = \frac{s(1-\gamma)}{1-s\gamma}\,,$$

for $|s| < \gamma^{-1}$, which is the probability generating function of a Geometric$(1-\gamma)$ random variable, as required. □

Proof of Lemma B.2. This follows as a straightforward corollary of Lemma B.1; under the notation of that result, we have $T \overset{\mathcal{D}}{=} \sum_{i=1}^N T_i$. We now decompose this based on whether the event $\{N \geq n\}$ occurs, and use the fact that $\mathbb{P}(N=k) = \mathbb{P}(N'=k)$ for $k = 1, \ldots, n-1$:

$$T \overset{\mathcal{D}}{=} \sum_{i=1}^{\min(N,n-1)} T_i + \mathbb{1}\{N \geq n\} \sum_{i=n}^{N} T_i \overset{\mathcal{D}}{=} \sum_{i=1}^{\min(N',n-1)} T_i + \mathbb{1}\{N' = n\} T'\,,$$

as required. The final equality in distribution holds from the memoryless property of the geometric distribution; on the event $\{N \geq n\}$, we have $N-(n-1) \sim \text{Geometric}(1-\gamma/1-\beta)$, and hence $\sum_{i=n}^N T_i \sim \text{Geometric}(1-\gamma)$ on this event. □

Proof of Proposition B.3. This follows straightforwardly by induction. The case $n=1$ follows from Proposition 2.3. Now suppose the claim holds for $n=l$. Then we have $X^{(l)} \overset{\mathcal{D}}{=} X_{\sum_{i=1}^n T_i}$. So

$$X^{(l+1)} | X^{(l)}, A^{(l)} \sim \mu_\beta^\pi(\cdot | X^{(l)}, A^{(l)})\,,$$

and so by Proposition 2.3 again, we have $X^{(l+1)}|X^{(l)} \overset{\mathcal{D}}{=} X'_{T'}$, with $T' \sim \text{Geometric}(1-\beta)$, and $(X'_t, A'_t, R'_t)_{\geq 0}$ an independent trajectory following π with initial state $X^{(l)}$. But since $X^{(l)} \overset{\mathcal{D}}{=} X_{\sum_{i=1}^n T_i}$, by the Markov property we therefore have $X^{(l+1)} \overset{\mathcal{D}}{=} X_{\sum_{i=1}^n T_i + T'} \overset{\mathcal{D}}{=} X_{\sum_{i=1}^{n+1} T_i}$ as required. □

We now restate and prove Proposition 2.5.

Proposition 2.5. (Janner et al. (2020)) For any policy $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$, $n \geq 1$, and $0 \leq \beta < \gamma$ an unbiased estimator of $Q_\gamma^\pi(x,a)$ is given by

$$r(x,a) + \frac{\gamma}{1-\gamma} \times \left[\sum_{m=1}^{n-1} \frac{1-\gamma}{1-\beta}\left(\frac{\gamma-\beta}{1-\beta}\right)^{m-1} r^\pi(X^{(m)}) + \left(\frac{\gamma-\beta}{1-\beta}\right)^{n-1} r^\pi(X')\right], \tag{4}$$

where $X^{(m)} \sim \mu_\beta^\pi(\cdot|X^{(m-1)}, A^{(m-1)})$, $A^{(m)} \sim \pi(\cdot|X^{(m)})$, $(X^{(0)}, A^{(0)}) = (x,a)$, and $X' \sim \mu_\gamma^\pi(\cdot|X^{(n-1)}, A^{(n-1)})$.

Proof. We start from the expression for $Q_\gamma^\pi(x,a)$ in Equation 2. Using the notation of Proposition 2.3, we have

$$\mathbb{E}_{X' \sim \mu_\gamma^\pi(\cdot|x,a)}[r^\pi(X')] = \mathbb{E}_{x,a}^\pi[r^\pi(X_T)]\,.$$

Now with the notation of Lemma B.2, we have

$$\begin{aligned}\mathbb{E}^{\pi}_{x,a}[r^{\pi}(X_T)] &= \mathbb{E}^{\pi}_{x,a}[r^{\pi}(X_{\sum_{i=1}^{\min(N',n-1)} T_i + \mathbb{1}\{N'=n\}T'})] \\ &= \mathbb{E}[\mathbb{E}^{\pi}_{x,a}[r^{\pi}(X_{\sum_{i=1}^{\min(N',n-1)} T_i + \mathbb{1}\{N'=n\}T'}) \mid N']] \\ &= \sum_{m=1}^{n-1} \frac{1-\gamma}{1-\beta}\left(\frac{\gamma-\beta}{1-\beta}\right)^{m-1} \mathbb{E}^{\pi}_{x,a}[r^{\pi}(X_{\sum_{i=1}^{m} T_i})] + \left(\frac{\gamma-\beta}{1-\beta}\right)^{n-1} \mathbb{E}^{\pi}_{x,a}[r^{\pi}(X_{\sum_{i=1}^{n-1} T_i + T'})]\,.\end{aligned}$$

Finally, by Proposition B.3, we have $X_{\sum_{i=1}^{m} T_i} \overset{\mathcal{D}}{=} X^{(m)}$ as defined above, and $X_{\sum_{i=1}^{n-1} T_i + T'} \overset{\mathcal{D}}{=} X'$, to obtain the desired conclusion. □

B.3. Proofs of result from Section 3

Theorem 3.2. Consider an MDP with reward function $r : \mathcal{X} \to \mathbb{R}$ and let $\nu = \pi_1 \overset{\alpha}{\to} \cdots \overset{\alpha}{\to} \pi_n$. With $\beta = \gamma(1-\alpha)$, the following is unbiased for $Q^{\nu}_{\gamma}(x,a)$:

$$\begin{aligned}&r(x) + \frac{\gamma}{1-\gamma} \times \\ &\left[\sum_{m=1}^{n-1} \frac{1-\gamma}{1-\beta}\left(\frac{\gamma-\beta}{1-\beta}\right)^{m-1} r(X^{(m)}) + \left(\frac{\gamma-\beta}{1-\beta}\right)^{n-1} r(X')\right],\end{aligned} \tag{6}$$

where $(X^{(0)}, A^{(0)}) = (x,a)$, $X^{(m)} \sim \mu_{\beta}^{\pi_m}(\cdot|X^{(m-1)}, A^{(m-1)})$, $A^{(m)} \sim \pi_{m+1}(\cdot|X^{(m)})$, $X' \sim \mu_{\gamma}^{\pi_n}(\cdot|X^{(n-1)}, A^{(n-1)})$.

Proof. Just as with Markov policies, we have the basic identity

$$Q^{\nu}_{\gamma}(x,a) = r(x) + \frac{\gamma}{1-\gamma}\mathbb{E}^{\nu}_{x,a}[r(X_T)]\,.$$

We now show that $\mathbb{E}^{\nu}_{x,a}[r(X_T)]$ has the required form by induction on n. The base case $n = 1$ follows from Proposition 2.4. For the inductive step, fix $n = l$, and suppose the required form of the expectation has been demonstrated for all smaller values of n.

Let $\nu = \pi_1 \overset{\alpha}{\to} \cdots \overset{\alpha}{\to} \pi_l$. We consider the time to switch from the first policy π_1, to the second sampled policy, π_2, denoting this time T_1, recalling that its distribution is Geometric(α). We proceed by considering whether or not the geometric horizon $T \sim$ Geometric$(1-\gamma)$ is greater than T_1:

$$\begin{aligned}&\mathbb{E}^{\nu}_{x,a}[r(X_T)] \\ =&\mathbb{E}^{\nu}_{x,a}[r(X_T)\mathbb{1}_{T\le T_1} + r(X_T)\mathbb{1}_{T>T_1}] \\ =&\mathbb{E}^{\nu}_{x,a}[r(X_T) \mid T \le T_1]\mathbb{P}(T \le T_1 \mid X_0 = x, A_0 = a) \\ &\quad + \mathbb{E}^{\nu}_{x,a}[r(X_T) \mid T > T_1]\mathbb{P}^{\nu}_{x,a}(T > T_1)\,.\end{aligned} \tag{10}$$

Since T, T_1 are independent of the trajectory $(X_t, A_t, R_t)_{t\ge 0}$, we have $\mathbb{P}^{\nu}_{x,a}(T \le T_1) = \mathbb{P}(T \le T_1)$. To compute $\mathbb{P}(T \le T_1)$, we have

$$\begin{aligned}\mathbb{P}(T \le T_1) &= \sum_{k=1}^{\infty} \mathbb{P}(T \le k)\mathbb{P}(T_1 = k) \\ &= \sum_{k=1}^{\infty} (1-\gamma^k)\alpha(1-\alpha)^{k-1} \\ &= \alpha \sum_{k=1}^{\infty} ((1-\alpha)^{k-1} - \gamma(\gamma(1-\alpha))^{k-1}) \\ &= \alpha\left(\frac{1}{\alpha} - \frac{\gamma}{1-\gamma(1-\alpha)}\right) \\ &= \frac{1-\gamma}{1-\gamma(1-\alpha)}\,.\end{aligned}$$

Now, to compute $\mathbb{E}^{\nu}_{x,a}[r(X_T) \mid T \leq T_1]$, we need the marginal distribution of T given the event $\{T \leq T_1\}$, which again is independent of the trajectory $(X_t, A_t, R_t)_{t\geq 0}$. We have

$$\begin{aligned}\mathbb{P}(T = k \mid T \leq T_1) &\propto \mathbb{P}(T = k, T \leq T_1)\\ &= \sum_{l=k}^{\infty} \mathbb{P}(T = k)\mathbb{P}(T_1 = l)\\ &= (1-\gamma)\gamma^{k-1}(1-\alpha)^k\\ &\propto (\gamma(1-\alpha))^k\,,\end{aligned}$$

which is the probability mass function of a Geometric$(1-\gamma(1-\alpha))$ distribution. Hence, conditional on $T \leq T_1$, we have that $T \sim$ Geometric$(1-\gamma(1-\alpha))$, and that the policy ν has not switched from π_1 on this event, so

$$\mathbb{E}^{\nu}_{x,a}[r(X_T) \mid T \leq T_1] = \mathbb{E}_{X'\sim\mu^{\pi_1}_{\gamma(1-\alpha)}(\cdot|x,a)}[r(X')]\,.$$

We next turn our attention to the second term on the right-hand side of Equation (10). Conditional on $\{T > T_1\}$, we compute the joint distribution of $(T - T_1, T_1)$. For any $k, l > 0$:

$$\mathbb{P}(T - T_1 = k, T_1 = l \mid T > T_1) \propto \mathbb{P}(T - T_1 = k, T_1 = l) = \mathbb{P}(T = k + l, T_1 = l) \propto \gamma^{k+l}(1-\alpha)^l = \gamma^k(\gamma(1-\alpha))^l\,,$$

which we recognise as the distribution of two independent geometric random variables with parameters $1-\gamma$ and $1-\gamma(1-\alpha)$. Hence, a sample from X_T on the event $\{T > T_1\}$ can be obtained by first sampling the state $X^{(1)} \sim \mu^{\pi_1}_{\gamma(1-\alpha)}$ at which the switch from π_1 to π_2 occurs. From this point, we require a state sampled $T - T_1 \sim$ Geometric$(1-\gamma)$ steps into the future, from initial state $X^{(1)}$, and action $A^{(1)} \sim \pi_2(\cdot|X^{(1)})$, following the suffix GSP $\nu' = \pi_2 \overset{\alpha}{\to} \cdots \overset{\alpha}{\to} \pi_l$. By induction, the corresponding expectation can be expressed as

$$\mathbb{E}^{\nu}_{x,a}[r(X_T) \mid T > T_1] = \mathbb{E}\Big[\sum_{m=1}^{l-2} \frac{1-\gamma}{1-\beta}\left(\frac{\gamma-\beta}{1-\beta}\right)^{m-1} r(\bar{X}^{(m)}) + \left(\frac{\gamma-\beta}{1-\beta}\right)^{l-2} r(\bar{X}')\Big]\,,$$

where $\bar{X}^{(0)} \sim \mu^{\pi_1}_{\beta}(\cdot|x,a)$, $\bar{X}^{(m)} \sim \mu^{\pi_{m+1}}_{\beta}(\cdot|\bar{X}^{(m\text{-}1)}, \bar{A}^{(m\text{-}1)})$, $\bar{A}^{(m)} \sim \pi_{m+2}(\cdot|\bar{X}^{(m)})$, $\bar{X}' \sim \mu^{\pi_l}_{\gamma}(\cdot|\bar{X}^{(l-2)}, \bar{A}^{(l-2)})$.

Rewriting in terms of the original sequence $(X^{(0)}, X^{(1)}, \ldots, X^{(n)}, A^{(n)}, X')$ in the theorem statement, we have

$$\mathbb{E}^{\nu}_{x,a}[r(X_T) \mid T > T_1] = \mathbb{E}\Big[\sum_{m=1}^{l-2} \frac{1-\gamma}{1-\beta}\left(\frac{\gamma-\beta}{1-\beta}\right)^{m-1} r(X^{(m+1)}) + \left(\frac{\gamma-\beta}{1-\beta}\right)^{l-2} r(X')\Big]\,.$$

Putting everything together from the decomposition in Equation (10), we therefore have

$$\begin{aligned}&\mathbb{E}^{\nu}_{x,a}[r(X_T)]\\ =&\frac{1-\gamma}{1-\gamma\beta}\mathbb{E}[r(X^{(1)})] + \frac{\gamma-\beta}{1-\beta}\mathbb{E}\Big[\sum_{m=1}^{l-2} \frac{1-\gamma}{1-\beta}\left(\frac{\gamma-\beta}{1-\beta}\right)^{m-1} r(X^{(m+1)}) + \left(\frac{\gamma-\beta}{1-\beta}\right)^{l-2} r(X')\Big]\\ =&\mathbb{E}\Big[\sum_{m=1}^{l-1} \frac{1-\gamma}{1-\beta}\left(\frac{\gamma-\beta}{1-\beta}\right)^{m-1} r(X^{(m)}) + \left(\frac{\gamma-\beta}{1-\beta}\right)^{l-1} r(X')\Big]\end{aligned}$$

as required. □

B.4. Proof of result from Section 4

Theorem 4.2. Consider a suffix-closed collection of GSPs Π. Then if $\pi' \in \mathcal{G}(\Pi)$, we have

$$Q^{\pi'}_{\gamma}(x,a) \geq \max_{\nu\in\Pi} Q^{\nu}_{\gamma}(x,a)\,, \quad \text{for all } (x,a) \in \mathcal{X}\times\mathcal{A}\,.$$

Further, if equality holds for all state-action pairs, then π' is optimal.

Proof. It is sufficient to show that for any policy $\nu \in \Pi$, we have $Q^{\pi'} \geq Q^{\nu}$. If $\nu = \pi$ is Markov, then we have

$$Q^{\pi}_{\gamma}(x,a) = r(x,a) + \gamma \sum_{x' \in \mathcal{X}} \sum_{a' \in \mathcal{A}} P(x'|x,a)\pi(a'|x')Q^{\pi}_{\gamma}(x',a')\,,$$

and hence

$$Q^{\pi}_{\gamma}(x,a) \leq r(x,a) + \gamma \sum_{x' \in \mathcal{X}} \sum_{a' \in \mathcal{A}} P(x'|x,a)\pi'(a'|x') \max_{\bar{\nu} \in \Pi} Q^{\bar{\nu}}_{\gamma}(x',a') = (T^{\pi'}(\max_{\bar{\nu} \in \Pi} Q^{\bar{\nu}}_{\gamma}))(x,a)\,.$$

Now suppose $\nu = \pi_1 \xrightarrow{\alpha} \cdots \xrightarrow{\alpha} \pi_n \in \Pi$ is a non-Markov geometric switching policy. Let $\nu' = \pi_2 \xrightarrow{\alpha} \cdots \xrightarrow{\alpha} \pi_n$ be the suffix policy of π. By suffix-closedness of Π, $\nu' \in \Pi$, and so we have the following observation:

$$\begin{aligned}
Q^{\nu}_{\gamma}(x,a) &= r(x,a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,a) \left[(1-\alpha) \sum_{a' \in \mathcal{A}} \pi_1(a'|x') Q^{\nu}_{\gamma}(x',a') + \alpha \sum_{a' \in \mathcal{A}} \pi_2(a'|x') Q^{\nu'}_{\gamma}(x',a') \right] \\
&\leq r(x,a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,a) \left[(1-\alpha) \sum_{a' \in \mathcal{A}} \pi'(a'|x') \max_{\bar{\nu} \in \Pi} Q^{\bar{\nu}}_{\gamma}(x',a') + \alpha \sum_{a' \in \mathcal{A}} \pi'(a'|x') \max_{\bar{\nu} \in \Pi} Q^{\bar{\nu}}_{\gamma}(x',a') \right] \\
&= (T^{\pi'}(\max_{\bar{\nu} \in \Pi} Q^{\bar{\nu}}_{\gamma}))(x,a)\,,
\end{aligned}$$

similarly to the Markov case. By taking a maximum over the policy considering on the left-hand side of the main chain of inequalities above, we get $\max_{\bar{\nu} \in \Pi} Q^{\bar{\nu}} \leq T^{\pi'}(\max_{\bar{\nu} \in \Pi} Q^{\bar{\nu}})$. As in the proof of improvement guarantee for standard GPI, we have that $T^{\pi'}$ is monotone, and contracts to $Q^{\pi'}$. Hence, $Q^{\nu} \leq \max_{\nu \in \Pi} Q^{\nu} \leq \lim_{n \to \infty} (T^{\pi'})^n (\max_{\nu \in \Pi} Q^{\nu}) = Q^{\pi'}$, as required. For the final statement of the result, observe that if equality holds at all state-action pairs, then we have that $\max_{\nu \in \Pi} Q^{\nu}_{\gamma}$ satisfies the Bellman optimality equation $\max_{\nu \in \Pi} Q^{\nu}_{\gamma} = T^{\pi'} \max_{\nu \in \Pi} Q^{\nu}_{\gamma} = T^{*} \max_{\nu \in \Pi} Q^{\nu}_{\gamma}$, and hence $\max_{\nu \in \Pi} Q^{\nu}_{\gamma} = Q^{\pi'} = Q^{*}$, so π' is optimal. □

B.5. Proof of result from Section 5

Proposition 5.1. Π_m is suffix-closed.

Proof. Given a policy $\nu = \pi^{(1)} \xrightarrow{\alpha} \cdots \pi^{(m)} \in \Pi_m$, its suffix policy is $\nu' = \pi^{(2)} \xrightarrow{\alpha} \cdots \pi^{(m)}$. On the face of it, this policy appears not to lie in Π_m, since it contains only $m-2$ switches. However, the key observation is that appending an additional switch from the tail Markov policy to itself does not change the geometric switching policy; that is

$$\pi^{(2)} \xrightarrow{\alpha} \cdots \pi^{(m-1)} \xrightarrow{\alpha} \pi^{(m)} = \pi^{(2)} \xrightarrow{\alpha} \cdots \pi^{(m-1)} \xrightarrow{\alpha} \pi^{(m)} \xrightarrow{\alpha} \pi^{(m)}\,.$$

The right-hand side clearly lies in Π_m, and hence the proof of suffix-closedness is complete. The improvement guarantee now follows from Theorem 4.2. □

B.6. Proof of result from Section 6

Here, we provide a proof of Proposition 6.2, and note that the (longer) proof of Theorem 6.3 is given in Appendix C.

Proposition 6.2. Defining the Bellman operator $T^{\pi}_{\beta} : \mathscr{P}(\mathcal{X})^{\mathcal{X} \times \mathcal{A}} \to \mathscr{P}(\mathcal{X})^{\mathcal{X} \times \mathcal{A}}$ by

$$(T^{\pi}_{\beta}\mu)(x'|x,a) = (1-\beta)P(x'|x,a) + \beta(\mu \otimes_{\pi} P)(x'|x,a)\,,$$

then μ^{π}_{β} is the unique solution to $\mu = T^{\pi}_{\beta}\mu$.

Proof. That μ_β^π solves $\mu = T_\beta^\pi \mu$ follows straightforwardly from the Markov property of the environment:

$$\begin{aligned}
\mu_\beta^\pi(x'|x,a) =&(1-\beta)\mathbb{E}_{x,a}^\pi\Big[\sum_{t\geq 0}\beta^t \mathbb{1}_{X_{t+1}=x'}\Big] \\
=&(1-\beta)\mathbb{E}_{x,a}^\pi\Big[\mathbb{1}_{X_{t+1}=x'}\Big] + \beta\mathbb{E}_{x,a}^\pi\Big[(1-\beta)\mathbb{E}_{X_1,A_1}^\pi\Big[\sum_{t\geq 1}\beta^t \mathbb{1}_{X_{t+1}=x'}\Big]\Big] \\
=&(1-\beta)P(x'|x,a) + \beta\mathbb{E}_{x,a}^\pi\Big[\mu(x'|X_1,A_1)\Big] \\
=&(1-\beta)P(x'|x,a) + \beta\sum_{x''\in\mathcal{X}}\sum_{a''\in\mathcal{A}} P(x''|x,a)\pi(a''|x'')\mu(x'|x',a') \\
=&(1-\beta)P(x'|x,a) + \beta(\mu\otimes_\pi P)(x'|x,a)\,.
\end{aligned}$$

We now show that T_β^π is a contraction mapping on $\mathscr{P}(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$. Let $\mu, \mu' \in \mathscr{P}(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$, from which uniqueness of the solution to $\mu = T_\beta^\pi\mu$ immediately follows. We directly calculate

$$\begin{aligned}
(T_\beta^\pi\mu - T_\beta^\pi\mu')(x'|x,a) =& \left((1-\beta)P(x'|x,a) + \beta\sum_{x''\in\mathcal{X}}\sum_{a'\in\mathcal{A}} P(x''|x,a)\pi(a'|x'')\mu(x'|x'',a')\right) - \\
&\left((1-\beta)P(x'|x,a) + \beta\sum_{x''\in\mathcal{X}}\sum_{a'\in\mathcal{A}} P(x''|x,a)\pi(a'|x'')\mu'(x'|x'',a')\right) \\
=&\ \beta\sum_{x''\in\mathcal{X}} P(x''|x,a)\pi(a'|x'')(\mu(x'|x'',a') - \mu'(x'|x'',a'))\,.
\end{aligned}$$

Hence,

$$\max_{(x,a,x')\in\mathcal{X}\times\mathcal{A}\times\mathcal{X}} |(T_\beta^\pi\mu - T_\beta^\pi\mu')(x'|x,a)| \leq \beta \max_{(x,a,x')\in\mathcal{X}\times\mathcal{A}\times\mathcal{X}} |(\mu-\mu')(x'|x,a)|\,,$$

as required. □

C. Proof of the convergence of cross-entropy temporal-difference learning

In this section we prove Theorem 6.3, which establishes the convergence of cross-entropy TD learning in the tabular, finite state-space setting, under mild conditions. The broad structure of the proof follows that of many arguments in stochastic approximation: defining a Lyapnuov function, showing convergence of this Lyapunov function to 0 as the algorithm progresses via the Robbins-Siegmund theorem (Robbins & Siegmund, 1971), and deducing convergence of the algorithm as a consequence; see for example Kushner & Yin (2003) for further background. We begin by recalling the details of the theorem.

Statement of result. The algorithm generates a sequence of logits $(\phi_k)_{k\geq 0}$, with $\phi_k \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}\times\mathcal{X}}$, and corresponding estimated geometric horizon models, denoted μ_k, and defined by

$$\mu_k(x'|x,a) = \frac{\exp(\phi_k(x'|x,a))}{\sum_{x''\in\mathcal{X}}\exp(\phi_k(x''|x,a))}\,.$$

We work with a synchronous algorithm, for which every state-action pair is updated at every algorithm time step. Thus, $\phi_0 \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}\times\mathcal{X}}$ is initialised in some manner, and for each algorithm time step $k \geq 0$, for each (x,a) we take a transition (x,a,X') generated from the MDP, independent of all other transitions used at time k and earlier, and define ϕ_{k+1} via the update

$$\phi_{k+1}(\cdot|x,a) = \phi_k(\cdot|x,a) - \varepsilon_k \nabla_{\phi_k(\cdot|x,a)} \mathrm{KL}(\mathtt{SG}[(\hat{T}^\pi\mu_k)(\cdot|x,a)]\ ||\ \mu_k(\cdot|x,a))\,, \tag{11}$$

where $\mathtt{SG}$ denotes a stop-gradient, and $(\hat{T}^\pi\mu_k)(\cdot|X_k,A_k)$ is an unbiased approximation error to the Bellman operator application $(T^\pi\mu_k)(\cdot|X_k,A_k)$, given by

$$(\hat{T}^\pi\mu_k)(\cdot|x,a) = (1-\gamma)e_{X'} + \gamma e_{X''}\,,$$

where X'' is sampled first by sampling $A' \sim \pi(\cdot|X')$, and then $X'' \sim \mu_k(\cdot|X', A')$. Evaluating the gradient above allows us to re-express the update as

$$\phi_{k+1}(\cdot|x,a) = \phi_k(\cdot|x,a) + \varepsilon_k \left((\hat{T}^\pi \mu_k)(\cdot|x,a) - \mu_k(\cdot|x,a) \right) . \tag{12}$$

Then the theorem statement is that if the Robbins-Monro conditions for the step sizes $(\varepsilon_k)_{k=0}^\infty$ hold, then we have $\mu_k \to \mu_\gamma^\pi$ with probability 1.

Proof. The proof of the result is presented below. We include schematic illustrations of some of the key ideas in the proof in Figure 7.

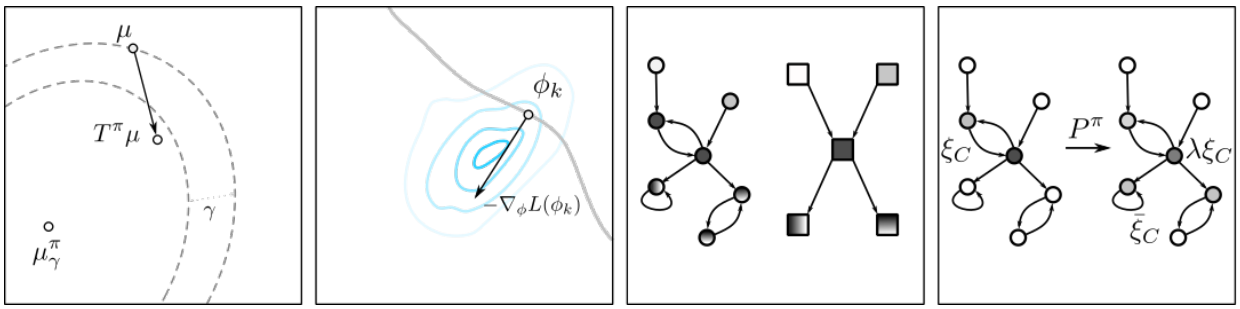


Figure 7. Schematic illustrations of core ideas in the convergence proof for cross-entropy temporal-difference learning. **Left:** Contractivity of the operator T^π in the weighted L^2 norm $\|\cdot\|_\xi$ towards μ_γ^π. **Centre-left:** For a given value of ϕ_k, the corresponding level set of the Lyapunov function L as a grey line, the conditional distribution over ϕ_{k+1} illustrated with blue contours, and the negative gradient of the Lyapunov function indicated as a black arrow. The Robbins-Siegmund argument shows that even though ϕ_{k+1} may have a higher Lyapunov value than ϕ_k, in the long term the value of the Lyapunov function must converge to 0. **Centre-right:** The decomposition of a Markov chain state space into a directed acyclic graph of communicating classes. **Right:** The distribution ξ_C supported on a given communicating class C, as constructed via the Perron-Frobenius theorem, and the result of right-multiplying by the Markov chain transition matrix P^π; on the communicating class C, the distribution is scaled by λ, while descendant communicating classes may now have non-zero probabilities, given by $\bar{\xi}_C$.

The Lyapunov function. Let ξ be a stationary state-action distribution under π, and suppose initially that it has full support; we will explain how to remove this assumption below. It is useful to introduce the function $\mu : \mathbb{R}^{\mathcal{X}\times\mathcal{A}\times\mathcal{X}} \to \mathscr{P}(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$ for the softmax function that maps logits to corresponding collections of probability distributions. We now define the Lyapunov function

$$L(\phi) = \sum_{x,a} \xi(x,a) \mathrm{KL}(\mu_\gamma^\pi(\cdot|x,a) \,||\, \mu(\phi)(\cdot|x,a)) .$$

The full support condition ensures that $L(\phi) = 0$ implies that $\mu(\phi) = \mu_\gamma^\pi$. Our goal is to show that $L(\phi_k) \to 0$ almost surely, hence $\sum_{x,a} \xi(x,a) \mathrm{KL}(\mu_\gamma^\pi(\cdot|x,a) \,||\, \mu(\phi_k)(\cdot|x,a)) \to 0$, and so $\mu_k \to \mu_\gamma^\pi$, as required.

A supermartingale argument. We start by considering a second-order Taylor expansion (with Lagrange remainder) of $L(\phi_{k+1})$ around ϕ_k (here, and in the remainder of the proof, it is useful to interpret a probability distribution in $\mathscr{P}(\mathcal{X})$ as a vector in $\mathbb{R}^\mathcal{X}$ — specifically, an element of the simplex $\Delta(\mathcal{X})$, which we will do without further remark):

$$L(\phi_{k+1}) = L(\phi_k + \varepsilon_k(\hat{T}^\pi \mu_k - \mu_k)) = L(\phi_k) + \varepsilon_k \langle \nabla_\phi L(\phi_k), \hat{T}^\pi \mu_k - \mu_k \rangle + \varepsilon_k^2 \nabla_\phi^2 L(\tilde{\phi_k})[\hat{T}^\pi \mu_k - \mu_k, \hat{T}^\pi \mu_k - \mu_k] ,$$

for some $\tilde{\phi}_k$ on the line segment $[\phi_k, \phi_{k+1}]$. Defining $\mathscr{F}_k$ to be the sigma-algebra generated by all random variables up to, but not including, those defining the update from ϕ_k to ϕ_{k+1}, we have

$$\mathbb{E}[L(\phi_{k+1}) \mid \mathscr{F}_k] = L(\phi_k) + \varepsilon_k \mathbb{E}[\langle \nabla_\phi L(\phi_k), \hat{T}^\pi \mu_k - \mu_k \rangle \mid \mathscr{F}_k] + \varepsilon_k^2 \mathbb{E}[\nabla_\phi^2 L(\tilde{\phi_k})[\hat{T}^\pi \mu_k - \mu_k, \hat{T}^\pi \mu_k - \mu_k] \mid \mathscr{F}_k] .$$

From the form of the gradient $\nabla_\phi L(\phi)$, the Hessian $\nabla_\phi^2 L(\phi)$ is readily seen to be bounded, and the inputs above $\hat{T}^\pi \mu_k - \mu_k$ are also bounded, meaning there is a constant $K > 0$ such that

$$\mathbb{E}[L(\phi_{k+1}) \mid \mathscr{F}_k] \leq L(\phi_k) + \varepsilon_k \mathbb{E}[\langle \nabla_\phi L(\phi_k), \hat{T}^\pi \mu_k - \mu_k \rangle \mid \mathscr{F}_k] + \varepsilon_k^2 K .$$

To deal with the first-order term, we note that a straightforward calculation gives

$$[\nabla_\phi L(\phi)](x'|x,a) = \xi(x,a)(\mu(\phi)(x'|x,a) - \mu^\pi_\gamma(x'|x,a))\,.$$

We hence have

$$\mathbb{E}[\langle \nabla_\phi L(\phi_k), \hat{T}^\pi \mu_k - \mu_k\rangle \mid \mathscr{F}_k] = \langle \mu_k - \mu^\pi_\gamma, T^\pi \mu_k - \mu_k\rangle_\xi\,.$$

Now we use a contractivity argument to bound this derivative. We first argue that T^π as defined above is a γ-contraction under the norm $\|\cdot\|_\xi$ defined by $\|\mu\|_\xi^2 = \sum_{x,a,x'} \xi(x,a)\mu(x'|x,a)^2$. To see this, note

$$\begin{aligned}
\|T^\pi\mu - T^\pi\mu'\|_\xi^2 =& \|\gamma P^\pi \mu - \gamma P^\pi \mu'\|_\xi^2 \\
=& \gamma^2 \|P^\pi\mu - P^\pi\mu\|_\xi^2 \\
=& \gamma^2 \sum_{x,a,x'} \xi(x,a) \left(\sum_{x'',a''} P(x''|x,a)\pi(a''|x'')(\mu(x'|x'',a'') - \mu'(x'|x'',a'')) \right)^2 \\
\overset{(a)}{\leq}& \gamma^2 \sum_{x,a,x'} \xi(x,a) \sum_{x'',a''} P(x''|x,a)\pi(a''|x'')((\mu(x'|x'',a'') - \mu'(x'|x'',a')))^2 \\
\overset{(b)}{=}& \gamma^2 \sum_{x,a,x'} \xi(x,a)(\mu(x'|x,a) - \mu'(x'|x,a))^2 \\
=& \gamma^2 \|\mu - \mu'\|_\xi^2\,,
\end{aligned}$$

as required, with (a) following from Jensen's inequality, and (b) from ξ being stationary.

Using this contraction result, we have:

$$\begin{aligned}
&\|\mu^\pi_\gamma - T^\pi\mu_k\|_\xi^2 \leq \gamma^2 \|\mu^\pi_\gamma - \mu_k\|_\xi^2 \\
\implies &\|\mu^\pi_\gamma - \mu_k + \mu_k - T^\pi\mu_k\|_\xi^2 \leq \gamma^2 \|\mu^\pi - \mu_k\|_\xi^2 \\
\implies &\|\mu^\pi_\gamma - \mu_k\|_\xi^2 + \|\mu_k - T^\pi\mu_k\|_\xi^2 + 2\langle \mu^\pi_\gamma - \mu_k, \mu_k - T^\pi\mu_k\rangle_\xi \leq \gamma^2 \|\mu^\pi_\gamma - \mu_k\|_\xi^2 \\
\implies &\langle \mu_k - \mu^\pi_\gamma, T^\pi\mu_k - \mu_k\rangle_\xi \leq \frac{1}{2}\left((\gamma^2 - 1)\|\mu^\pi_\gamma - \mu_k\|_\xi^2 - \|\mu_k - T^\pi\mu_k\|_\xi^2\right) \leq -\frac{1-\gamma^2}{2}\|\mu^\pi_\gamma - \mu_k\|_\xi^2\,.
\end{aligned}$$

Returning to the Lyapunov function, we therefore have

$$\mathbb{E}[L(\phi_{k+1}) \mid \mathscr{F}_k] \leq L(\phi_k) - \varepsilon_k \frac{1-\gamma^2}{2}\|\mu^\pi_\gamma - \mu_k\|_\xi^2 + \varepsilon_k^2 K\,.$$

We now follow the ideas of the Robbins-Siegmund theorem (Robbins & Siegmund, 1971). Based on the above inequality, $(L(\phi_k))_{k\geq 0}$ is almost a positive supermartingale, save for the additive $\varepsilon_k^2 K$ terms in the upper bounds on the conditional expectation. However, defining $\tilde{L}_k = L(\phi_k) - \sum_{l=0}^{k-1} \varepsilon_l^2 K + \sum_{l=0}^{k-1} \varepsilon_l \frac{1-\gamma^2}{2}\|\mu^\pi_\gamma - \mu_l\|_\xi^2$, we have

$$\begin{aligned}
\mathbb{E}[\tilde{L}_{k+1} \mid \mathscr{F}_k] &\leq \mathbb{E}\Big[L(\phi_{k+1}) - \sum_{l=0}^{k} \varepsilon_l^2 K + \sum_{l=0}^{k} \varepsilon_l \frac{1-\gamma^2}{2}\|\mu^\pi_\gamma - \mu_l\|_\xi^2 \;\mid \mathscr{F}_k\Big] \\
&\leq L(\phi_k) - \sum_{l=0}^{k-1} \varepsilon_l^2 K + \sum_{l=0}^{k-1} \varepsilon_l \frac{1-\gamma^2}{2}\|\mu^\pi_\gamma - \mu_l\|_\xi^2 \\
&\leq \tilde{L}_k\,.
\end{aligned}$$

Hence $(\tilde{L}_k)_{k\geq 0}$ is a supermartingale. However, the subtraction of the $\varepsilon_k^2 K$ terms means that it is not a non-negative supermartingale, so we cannot immediately apply the supermartingale convergence theorem. The approach of Robbins & Siegmund (1971) is to define a sequence of stopping times $\tau_\ell = \inf\{k \geq 0 : \tilde{L}_k \leq -\ell\}$, for $\ell \in \mathbb{N}$. By the optional stopping theorem, $(\tilde{L}_{k\wedge\tau_\ell})_{k=0}^\infty$ is a supermartingale bounded below, and hence by the supermartingale convergence theorem,

converges almost surely. By the second Robbins-Monro step size condition, $\tau_\ell = \infty$ eventually almost surely, and hence $\tilde{L}_k$ converges almost surely, leading to almost-sure convergence of $L(\phi_k)$ too, as well as $\sum_{l=0}^{k} \varepsilon_l \frac{1-\gamma^2}{2} \|\mu_\gamma^\pi - \mu_l\|_\xi^2$. Due to the first Robbins-Monro step size condition $\sum_{k=0}^{\infty} \varepsilon_k = \infty$, we must have $\|\mu_\gamma^\pi - \mu_k\|_\xi^2 \to 0$, which completes the proof of the theorem in the case where ξ has full support.

A chaining argument for invariant distributions without full support. The previous argument relied on the existence of an invariant distribution ξ for the Markov chain over state-action pairs generated by the interaction of the policy π with the MDP in question. We now explain how to generalise this proof technique to remove this restriction on ξ.

First, by appending an artificial self-transitioning terminal state if required, there always exists an invariant distribution ξ for the Markov chain concerned, even in episodic settings where trajectories terminate in finite time. The argument above may be applied as-is to obtain the same conclusion $\sum_{l=0}^{k} \varepsilon_l \frac{1-\gamma^2}{2} \|\mu_\gamma^\pi - \mu_l\|_\xi^2 < \infty$, and hence $\|\mu_\gamma^\pi - \mu_k\|_\xi^2 \to 0$. The difference now is that this only shows convergence of μ_k to μ_γ^π along the state-action pairs with support under ξ.

We begin by recalling some notions from the theory of discrete-time Markov chains on finite sets; see Norris (1998) for further background. We also clarify that in Markov chain theory, the term *state space* is typically used to refer to the set of states which a Markov chain can take on. For our Markov chain, this *state space* is $\mathcal{X} \times \mathcal{A}$, *not* the usual state space of the MDP. To avoid confusion, we will use the term Markov chain state space (or MCSS) to distinguish the state space of the Markov chain from the set $\mathcal{X}$, and the term Markov chain state to refer to an element of the MCSS.

We can partition the MCSS $\mathcal{X} \times \mathcal{A}$ into *communicating classes*. A communicating class $C \subseteq \mathcal{X} \times \mathcal{A}$ is a set of Markov chain states such that for all $(x, a), (y, b) \in C$, there exists $t > 0$ such that $\mathbb{P}((X_t, A_t) = (x, a) \mid (X_0, A_0) = (y, b)) > 0$ and $\mathbb{P}((X_t, A_t) = (y, b) \mid (X_0, A_0) = (x, a)) > 0$, and further for any $(x, a) \in C$, no Markov chain state outside C has this property. The set of communicating classes of the Markov chain can be given a directed acyclic graph structure, by adding an edge from one class C to a distinct class C' if there exist $(x, a) \in C$, $(y, b) \in C'$ with $\mathbb{P}((X_1, A_1) = (y, b) \mid (X_0, A_0) = (x, a)) > 0$. Let us refer to this directed acyclic graph as G. Without loss of generality to what follows, we may assume G is connected (the argument may be applied to each connected component of G separately if G is not connected).

The goal is to recurse backwards through the directed acyclic graph G, establishing first for the Markov chain states (x, a) in communicating classes in the leaves of the graph that $\mu_k(\cdot|x, a) \to \mu_\gamma^\pi(\cdot|x, a)$, and then inductively moving back through the graph. Note that the leaves of G are precisely the *recurrent* communicating classes of the Markov chain: those classes C for which there exists an invariant distribution ξ_C for the Markov chain supported precisely on C. The argument above establishes that $\mu_k(\cdot|x, a) \to \mu_\gamma^\pi(\cdot|x, a)$ for all $(x, a) \in C$, and in fact the stronger conclusion that $\sum_{l=0}^{\infty} \varepsilon_l \|\mu_\gamma^\pi - \mu_l\|_{\xi_C}^2 < \infty$.

Now, for the inductive step of the argument, let C be a non-recurrent communicating class of the Markov chain, and suppose that for every descendant C' of C in the directed acyclic graph G, we have established that for some distribution $\xi_{C'}$ supported on C', we have $\sum_{l=0}^{\infty} \varepsilon_l \|\mu_\gamma^\pi - \mu_l\|_{\xi_{C'}}^2 \to 0$. We now aim to construct a distribution ξ_C supported on C, and to demonstrate that $\sum_{l=0}^{\infty} \varepsilon_l \|\mu_\gamma^\pi - \mu_l\|_{\xi_C}^2 \to 0$, so that by induction the theorem is proven.

To do this, we appeal to the Perron-Frobenius theorem (Perron, 1907; Frobenius, 1912); see Seneta (2006) for a recent account. Specifically, we consider the transition matrix of the Markov chain in question, and consider the sub-matrix obtained by deleting all rows and columns corresponding to Markov chain states outside C. The resulting matrix is strictly sub-stochastic (all elements are non-negative, rows sums are less than or equal to 1, with at least one row having row sum strictly less than 1), and hence by the Perron-Frobenius theorem, there exists a left-eigenvector $v \in \mathbb{R}^C$ for this matrix with eigenvalue $0 \leq \lambda < 1$, and all elements positive; we may further scale v so that the elements sum to 1. We now set ξ_C to be the distribution over the Markov chain state space that is equal to v on C, and 0 elsewhere. We now show that T^π still behaves 'almost' like a contraction under $\|\cdot\|_{\xi_C}$, which will allow us to re-use the supermartingale argument above. First note that from the structure of the communicating classes, we have that $\xi_C P^\pi$ is equal to $\lambda \xi_C$ on C, some other non-negative vector $\bar{\xi}_C$ on $\bar{C}$ the union of descendant communicating classes from C, and 0 elsewhere. Now, note that for

$\mu, \mu' \in \Delta(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$, we have

$$\begin{aligned}
&\|T^\pi\mu - T^\pi\mu'\|^2_{\xi_C}\\
=&\gamma^2\|P^\pi\mu - P^\pi\mu\|^2_{\xi_C}\\
=&\gamma^2\sum_{x,a,x'}\xi_C(x,a)\left(\sum_{x'',a''}P(x''|x,a)\pi(a''|x'')(\mu(x'|x'',a'')-\mu'(x'|x'',a''))\right)^2\\
\leq&\gamma^2\sum_{x,a,x'}\xi_C(x,a)\sum_{x'',a''}P(x''|x,a)\pi(a''|x'')((\mu(x'|x'',a'')-\mu'(x'|x'',a')))^2\\
=&\gamma^2\left(\sum_{(x,a)\in C}\lambda\xi_C(x,a)\sum_{x'}(\mu(x'|x,a)-\mu'(x'|x,a))^2+\sum_{(x,a)\in\bar{C}}\bar{\xi}_C(x,a)\sum_{x'}(\mu(x'|x,a)-\mu'(x'|x,a))^2\right)\\
=&\gamma^2\lambda\|\mu-\mu'\|^2_{\xi_C}+\gamma^2\|\mu-\mu'\|^2_{\bar{\xi}_C}\,.
\end{aligned}$$

The intuition here is that if $\mu \approx \mu'$ on $\bar{C}$, then we have a contraction-like bound for T^π as measured by ξ_C. From this, we obtain the bound

$$\langle\mu_k-\mu^\pi_\gamma, T^\pi\mu_k-\mu_k\rangle_{\xi_C} \leq -\frac{1-\gamma^2\lambda}{2}\|\mu^\pi_\gamma-\mu_k\|^2_{\xi_C}+\frac{\gamma^2}{2}\|\mu_k-\mu^\pi_\gamma\|^2_{\bar{\xi}_C}\,.$$

Defining an alternative Lyapunov function by

$$L_{\xi_C}(\phi)=\sum_{x,a}\xi_C(x,a)\mathrm{KL}(\mu^\pi_\gamma(\cdot|x,a)\;||\;\mu(\phi)(\cdot|x,a))\,,$$

a similar calculation to the above gives

$$\mathbb{E}[L_{\xi_C}(\phi_{k+1})\mid\mathscr{F}_k]\leq L_{\xi_C}(\phi_k)-\varepsilon_k\frac{1-\gamma^2\lambda}{2}\|\mu^\pi_\gamma-\mu_k\|^2_{\xi_C}+\varepsilon_k\frac{\gamma^2}{2}\|\mu_k-\mu^\pi_\gamma\|^2_{\bar{\xi}_C}+\varepsilon_k^2K\,.$$

The inductive hypothesis leads to $\sum_{l=0}^\infty\varepsilon_l\gamma^2/2\|\mu_l-\mu^\pi_\gamma\|^2_{\bar{\xi}_C}<\infty$, and so defining the modified sequence

$$\tilde{L}^{\xi_C}_k=L_{\xi_C}(\phi_k)-\sum_{l=0}^{k-1}\left(\varepsilon_l^2K+\varepsilon_l\frac{\gamma^2}{2}\|\mu_l-\mu^\pi_\gamma\|^2_{\bar{\xi}_C}\right)+\sum_{l=0}^{k-1}\varepsilon_l\frac{1-\gamma^2\lambda}{2}\|\mu^\pi_\gamma-\mu_l\|^2_{\xi_C}\,,$$

the same Robbins-Siegmund argument yields that $(\tilde{L}^{\xi_C}_k)_{k\geq0}$ is a convergent supermartingale, and hence $\sum_{l=0}^\infty\varepsilon_l\frac{1-\gamma^2\lambda}{2}\|\mu^\pi_\gamma-\mu_l\|^2_{\xi_C}<\infty$, as required to complete the induction, and hence the proof. □

C.1. Examples of cross-entropy TD learning

Figure 8 shows an example visualisation of the synchronous CETD algorithm in the case of a randomly-generated three-state, one-action MDP. The transition matrix and initial distributions μ_0 used to generated these plots are

$$P=\begin{pmatrix}0.297492728 & 0.702444212 & 0.000063060\\ 0.584810131 & 0.257810252 & 0.157379617\\ 0.181511854 & 0.373368720 & 0.445119427\end{pmatrix},\quad \phi_0=\begin{pmatrix}-2.3634686 & 1.13534535 & -1.01701414\\ 0.63736181 & -0.85990661 & 1.77260763\\ -1.11036305 & 0.18121427 & 0.56434487\end{pmatrix},$$

where as the MDP has a single action, we specify P as a state-by-state transition matrix, and similarly ϕ_0 is presented a state-by-state matrix, with each row corresponding to the logits of a single estimated future state-visitation distribution. Further, we take $\gamma = 0.9$, and the learning rate schedule used was $\varepsilon_k = 0.75(k+1)^{-0.6}$. In all plots presented in this section, we subsample the trajectories generated by a factor of 10 to make trajectories easier to visually inspect.

We also provide a further illustration of CETD below, in the case where the target μ^π_γ lies on the boundary of $\Delta(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$, by modifying the transition matrix P above to have a transient first state. Specifically, we set

$$P=\begin{pmatrix}0.765830909 & 0.234148071 & 0.000021020\\ 0 & 0.620945430 & 0.379054570\\ 0 & 0.456168756 & 0.543831244\end{pmatrix},$$

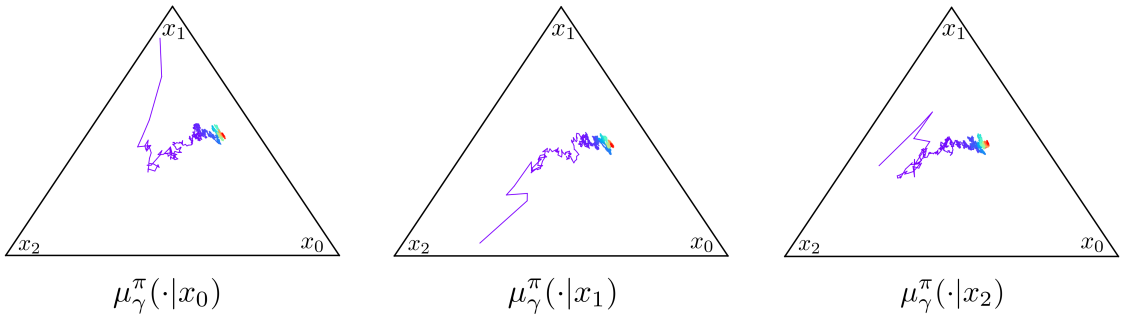


Figure 8. Illustration of CETD dynamics in a three-state MDP. The red dots indicate the fixed point of the operator T^π (the true visitation distributions). The coloured lines indicate the path taken by the CETD algorithm.

and use the same initialisation for ϕ_0 as described above. The results are shown in Figure 9. One interesting observation is that when the collection of true state visitation distributions lies on the boundary of the product of simplices, the convergence of the algorithm appears to be particularly slow. An intuition as to why this might be the case is that the sample-based CETD update is limited to decreasing logit values only by the magnitude of the current probability corresponding to the logit. Because of this, fitting zero (or near-zero) probabilities requires many gradient updates. This hints at the utility of further work to develop a finer-grained understanding of the asymptotic performance of this algorithm (such as asymptotic covariance and/or convergence rate), as well as approaches for variance reduction that may improve the convergence rate, either practically or empirically.

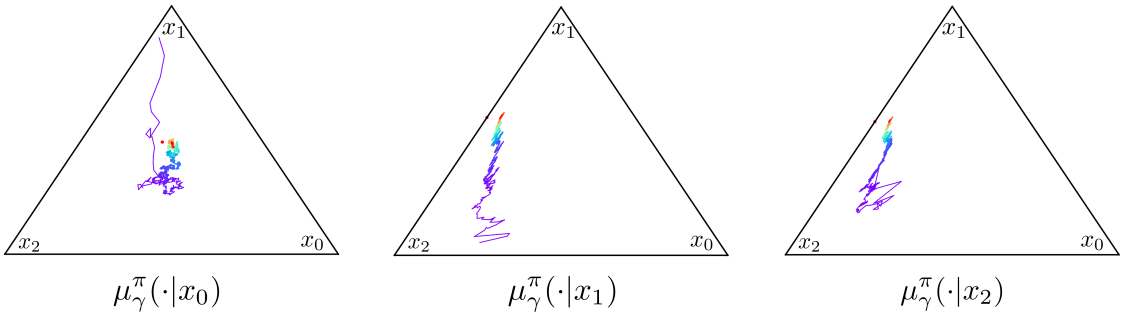


Figure 9. CETD dynamics for an MDP with μ_γ^π lying on the boundary of $\Delta(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$.

D. Further details and examples relating to core algorithms

D.1. GGPI counterexamples

We address several questions about the necessity of conditions for results appearing in the main paper through a set of counterexamples. Specifically, these questions and their resolutions are:

- Is the Q-function of a non-Markov policy always equal to that of a Markov policy? No: See Example D.1.
- Can we do GPI with the Q-functions of any collection of non-Markov policies? No: See Example D.2.
- If we restrict to GSPs, do we need the closure condition? Yes: See Example D.3.

Example D.1. Consider the two-state, two-action MDP in Figure 10, and consider a non-Markov policy π of the following form. When initialised in the left state, the agent seeks to take the action sequence bb, leading it to transition immediately to the right state, and then to terminate in the right state, having attained 0 reward. When initialised in the right state, the agent seeks to take the action sequence aab, attaining a reward of 2. In order for $Q^\pi(\mathrm{L}, b)$ to be equal to $Q^{\pi'}(\mathrm{L}, b)$ for some Markov policy π', it must be the case that π' takes action b in state R with probability 1. However, this is incompatible with the requirement $Q^\pi(\mathrm{R}, a) = Q^{\pi'}(\mathrm{R}, a)$, since we would require $\pi'(a|\mathrm{R}) > 0$.

Example D.2. As a very basic example of why non-Markov policies cannot in general be used within GPI, consider the one-state MDP in Figure 11. Consider a non-Markov policy π specified as follows:

- Initially, the policy randomises uniformly between actions a and b.
- If action a is selected at the first time-step, then the full sequence of actions is deterministically specified as $abbbb\ldots$.
- If action b is selected at the first time-step, then the full sequence of actions is deterministically specified as $baaaa\ldots$.

We therefore have $Q^\pi(a) = 1$, and $Q^\pi(b) = \gamma/(1-\gamma)$. So if $\gamma > \frac{1}{2}$, the greedy Markov policy obtained prefers action b, which is clearly worse for performance than the non-Markov policy π, meaning the GPI guarantee does not hold in this case.

Example D.3. Consider the MDP with a depth-3 binary tree transition structure displayed in Figure 12. Consider two policies π_{L} and π_{R} which always take the 'left' and 'right' actions in the tree. We consider the GSP ν that follows π_{L} for two steps and then switches to π_{R} (this is a GSP with two switches, and probability of switching equal to 1 in both cases). The value of this policy at the root node (in the undiscounted case) is +1 (obtained at the red leaf node), since the sequence of actions taken from the root node x_0 is LLR.

We now consider the Markov policy obtained by acting greedily with respect to Q^ν. To begin with, consider the Q-values $Q^\nu(x_0, \mathrm{L})$ and $Q^\nu(x_0, \mathrm{R})$. These are +1 and +2 respectively (obtained from the red and green leaf nodes), since these correspond to sequences of actions LLR and RLR respectively. So the greedy policy with respect to this Q-function takes the 'right' action at the base state x_0. Next, at state x_1, we have $Q^\nu(x_1, \mathrm{L}) = -1$ and $Q^\nu(x_1, \mathrm{R}) = 0$ (obtained at the blue and grey nodes), since these correspond to the action sequences LL and RL from x_1, meaning that the greedy policy takes the 'right' action in state x_2. This is enough to deduce that the greedy policy, executed from x_0, attains a return of 0, in contrast to the return of $+1$ obtained by the initial GSP ν; the greedy policy performs worse than the initial policy.

To see how the closure condition deals with this, note that the condition would require that we include the value functions for (i) the GSP that executes π_{L} for one step and then switches to π_{R} and (ii) π_{R} itself, in the GPI procedure. Performing GGPI over this collection of three policies then leads to an improved policy which when executed from x_0, obtains a return of +2, improving over all initial policies considered.

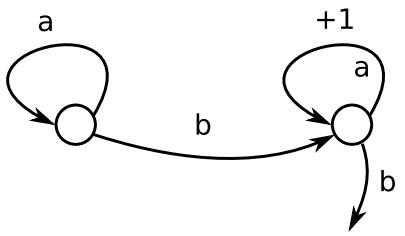


Figure 10. Example MDP showing that the Q-function of a non-Markov policy cannot necessarily be written as the Q-function of a Markov policy.

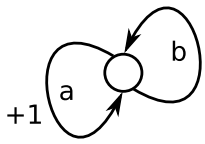


Figure 11. Example MDP illustrating that acting greedily with respect to a non-Markov policy may lead to detrimental performance.

D.2. Further examples of GGPI for policy iteration

We first give a full description in Algorithm 1 of the combination of GGPI with policy iteration that we consider in the paper. A possible optimisation is for each step to consider only those GSPs that end in the most recent policy π_k, as this dominates any GSP comprised of the same initial policies and ending in any other prior policy.

The example that follows then gives a granular sense of when GGPI delivers benefits over standard policy iteration in the tabular setting, to complement the four-rooms experiments in the main paper.

Example D.4. Figure 13 illustrates a chain environment with a large reward at one end of the chain, and 'distractor' rewards along the chain that may cause a myopic agent to prefer a sub-optimal action. An initial policy π_0 that is able to make

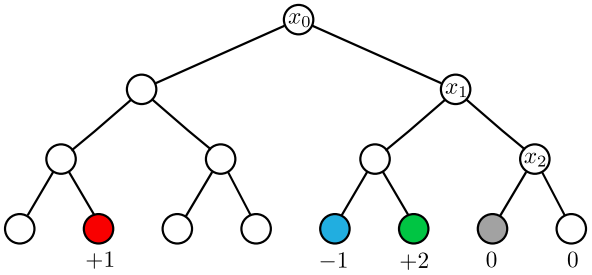


Figure 12. Example MDP illustrating that acting greedily with respect to a GSP may lead to detrimental performance.

Algorithm 1 GGPI for sample-based policy iteration

Require: Number of iterations n_{iter}, sample budget n_{samples}
  $i \leftarrow 0$
  $\pi' \leftarrow$ arbitrary policy
  $\Pi \leftarrow \emptyset$ {Set of policies seen so far}
  $\mathcal{M} \leftarrow \emptyset$ {Set of GHMs associated with policies in Π}
  repeat
    $\pi \leftarrow \pi'$
    $\Pi \leftarrow \Pi \cup \{\pi\}$
    Learn GHMs μ^π_γ and μ^π_β, with $\beta = \gamma(1-\alpha)$.
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mu^\pi_\gamma, \mu^\pi_\beta\}$
    Define a set $\mathcal{V}$ of GSPs using base policies in Π and switching probability α.
    for each state $x \in \mathcal{X}$ **do**
      For each $\nu \in \mathcal{V}$, estimate $Q^\nu_\gamma(x, \cdot)$ by composing n_{samples} GHM samples via Equation (6).
      $\pi'(x) \leftarrow \mathcal{G}\left(\{Q^\nu_\gamma(x,\cdot) | \nu \in \mathcal{V}\}\right)$
    end for
    $i \leftarrow i + 1$
  until $\pi' = \pi$ or $i \geq n_{\text{iter}}$

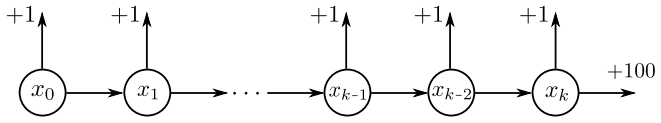
some progress towards the optimal side of the chain will have this progress 'wiped out' by myopic greedy improvements in standard policy iteration. In contrast, with GGPI, this initial progress can be used to deliver a stronger improvement over the first greedy policy π_1, leading to an optimal policy in fewer iterations. The figure illustrates an extreme setting where standard policy iteration would need $k+1$ improvement steps to reach the optimal policy, while GGPI reaches it in two single improvement steps. Note also that the set $\{\pi_0 \overset{\alpha}{\to} \pi_1, \pi_1\}$ is suffix-closed in the sense of Definition 4.1, so improvement is guaranteed by Theorem 4.2.

D.3. Full algorithmic description of GGPI for transfer

In Algorithm 2, we give algorithmic pseudocode to describe the use of GSP evaluation with GHMs and GGPI for transfer. There are several steps to the process: a set of Markov policies is given, which may be obtained through learning about prior reward signals, exploration objectives, imitation learning, etc. The agent learns GHMs for these models in a reward-free manner. A novel reward function is then revealed, and GGPI can be used to derive an improved policy in a zero-shot manner.

D.4. Geometric horizon models with $\beta = 0$ and $\beta > 0$

As noted in the main text, there is a close connection between GHMs with $\beta = 0$ (equivalently taking the switching probability as $\alpha = 1$) and one-step forward models traditionally used in planning. Concretely, we can say that $\mu^\pi_{\beta=0}(\cdot \mid x, a)$, for any policy π, is exactly identical to one-step forward models commonly used in planning. In Figure 14 (top row), we show examples of samples generated in this setting while varying the number of model compositions. For any $n \geq 1$, this corresponds to planning with a one-step model for $n-1$ steps and 'bootstrapping' afterwards with a value estimate of $V^\pi_\gamma(X')$ using μ^π_γ. When we set $\beta > 0$, the practice of planning with this model is much the same but each sample from the model may move one or more trajectory steps into the future, and now the policies $(\pi_i)_{i=1}^{n-1}$ determines the nature of this



Policy	x_0	x_1	$\cdots$	x_{k-2}	x_{k-1}	x_k
π_0	$\rightarrow$	$\rightarrow$	$\cdots$	$\rightarrow$	$\rightarrow$	$\uparrow$
$\pi_1 = \mathcal{G}(\pi_0)$	$\uparrow$	$\uparrow$	$\cdots$	$\uparrow$	$\uparrow$	$\rightarrow$
$\mathcal{G}(\pi_1)$	$\uparrow$	$\uparrow$	$\cdots$	$\uparrow$	$\rightarrow$	$\rightarrow$
$\mathcal{G}(\{\pi_0, \pi_1\})$	$\uparrow$	$\uparrow$	$\cdots$	$\uparrow$	$\rightarrow$	$\rightarrow$
$\mathcal{G}(\{\pi_0 \overset{\alpha}{\rightarrow} \pi_1, \pi_1\})$	$\rightarrow$	$\rightarrow$	$\cdots$	$\rightarrow$	$\rightarrow$	$\rightarrow$

Figure 13. Example chain environment, initial policy π_0, and policies generated by a variety of policy improvement steps. Light-blue shaded cells indicate optimal actions/policies. In this case, since π_0 encodes optimal behaviour in some states, it is useful to include switching policies beginning with π_0 in the policy improvement step, and $\mathcal{G}(\{\pi_0 \overset{\alpha}{\rightarrow} \pi_1, \pi_1\})$ is indeed optimal in this case.

Algorithm 2 GGPI for sample-based transfer.

Require: Markov policies $\pi_1, \ldots, \pi_k$, switching probability $\alpha \in (0, 1]$, sampling budget n_{samples}, discount γ.
  Learn GHMs $\mu_\gamma^{\pi_i}$ and $\mu_\beta^{\pi_i}$, with $\beta = \gamma(1-\alpha)$.
  Novel reward function r is revealed
  Select a suffix-closed set Π of GSPs
  for each state x encountered **do**
    For each $\nu \in \Pi$, estimate $Q_\gamma^\nu(x, \cdot)$ by composing n_{samples} GHM samples via Equation (6).
    Act greedily according to the output of $\mathcal{G}$ applied to these estimated Q-functions.
  end for

evolution. In Figure 14 (bottom row), the corresponding illustration for $\beta > 0$ is given. This again can be thought of as planning for $n-1$ steps and bootstrapping afterwards with a value estimate of $V_\gamma^\pi(X')$, however now each step of unrolling the model is temporally extended and thus potentially moves through many intermediate trajectory states (while following some Markov policy π_i). This allows for much deeper planning with fewer unroll steps than in the $\beta = 0$ (one-step model) case, potentially reducing problems with error accumulation common in planning with learned models. Finally, note that in the case of $n = 1$, the value of β has no effect, and we always sample directly from $\mu_\gamma^\pi(\cdot \mid x_0, a_0)$.

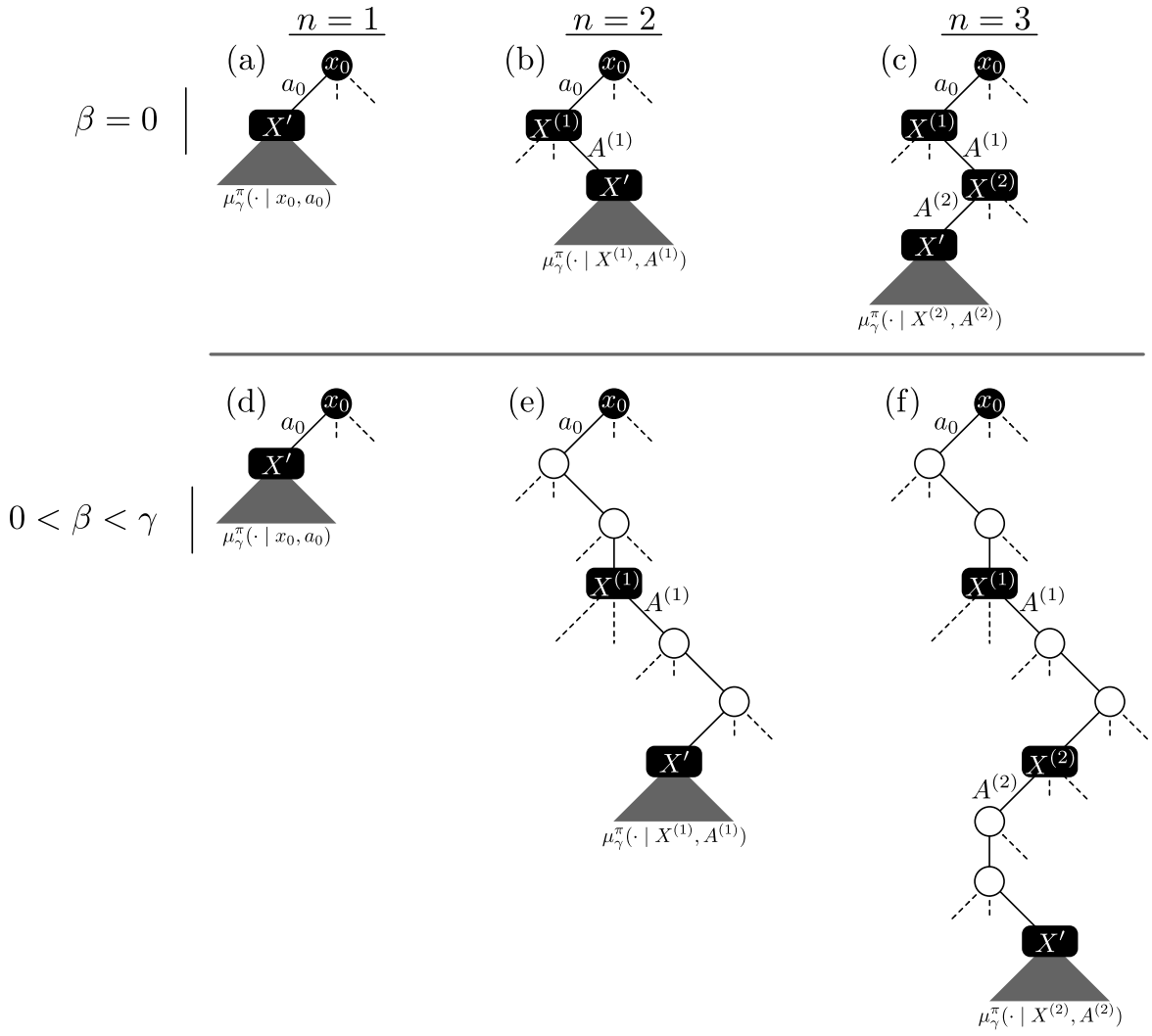


Figure 14. Illustrating samples for GHMs with (a)-(c) $\beta = 0$ and (d)-(f) $\beta > 0$, with $n \in \{1, 2, 3\}$ respectively. Starting state-action given by (x_0, a_0) shown as the root node and first branch. States that are sampled or directly observed are shown in black and actions explicitly conditioned on are denoted with solid line branches. Meanwhile, states shown in white are not sampled, but would have been produced while following the policy connecting two observed states. Action branches not followed are shown with dashed lines.

E. Further experiments and training details

All experiments were undertaken with Python, using NumPy (Harris et al., 2020) and Matplotlib (Hunter, 2007) for visualisation. Experiments involving deep neural networks were undertaken with Jax (Bradbury et al., 2018), specifically making use of the DeepMind Jax ecosystem (Babuschkin et al., 2020).

E.1. Sparse-reward ant experiment details

Environment and task. The sparse-reward ant domain is implemented in the MuJoCo framework (Todorov et al., 2012) for realistic physics simulations. The agent is a quadrapedal bot resembling an ant, first introduced by Schulman et al. (2016), with 8 controllable joints. The observation is a 35-dimensional representation of the true agent state, including information about its centre of mass position, velocity, joint angles and angular velocities, heading, etc. The environment has an 8-dimensional action space $[-1, 1]^8$ representing the torque to be applied at each joint. The ant is capable of moving about in an infinite unconstrained two-dimensional arena. Each episode starts with the agent randomly initialised at rest in a 20x20 square centered at the origin. A target is chosen randomly at a distance of between 2 to 4 units, at an angle that lies in the middle 30 degrees of each quadrant. Note that a policy trained to move consistently in a single direction typically progresses at around 0.2-0.4 units of distance per time step. The episode lasts for 150 time steps, or ends early if the ant reaches the target.

Policy training. We pretrain policies to move consistently in the 4 axis-aligned directions in the arena. In order to train the policies to effectively turn directions when switching between these policies, we jointly pretrain them by randomly switching between them every Uniform(40) steps. Thus the pretraining conditions mimic how the policies will be used during GGPI planning. We found training policies jointly on data generated in this manner to be important in learning policies that composed well together. In contrast, training the base policies entirely from on-policy data typically led to poor compositions in preliminary experiments.

The policies are implemented as stochastic Gaussian policies, with a 4-layer MLP with 256 hidden units including layer normalisation (Ba et al., 2016) and tanh non-linearities. The network outputs the mean/variance of the torque to be applied at each of the 8 joints independently. The policies are pretrained using the component of the velocity in the desired direction as a reward signal, using MPO (Abdolmaleki et al., 2018). The critic network used for MPO is a similar 5-layer MLP with 256 hidden units at each layer, with Layer Norm and tanh non-linearities. The policies are pretrained for 1 million update steps, using the Adam optimiser (Kingma & Ba, 2015) with a learning rate of 0.0003.

GHM training. We implement the GHMs as conditional β-VAE models with a single latent dimension and a $\beta_{\text{VAE}} = 20$. The encoder, prior, and decoder distributions are all assumed to be Gaussian, and implemented as a 3-layer MLP with 128 hidden units in each layer. They each take the concatenated representation of the current agent state and action (x, a) as auxiliary input to be conditioned on. We slightly modify the modelling task to predict the change in the agent state rather than the future state directly, i.e. we model $X_{t+\text{Geom}(1-\beta)} - X_t$ and add this to the state X_t to form a prediction, rather than directly modelling $X_{t+\text{Geom}(1-\beta)}$ itself. We found this to improve performance in terms of negative ELBO slightly. For each pretrained policy, we train 2 separate GHMs, one with geometric horizon parameter $\beta = 0.8$ and one with $\beta = 0.9$. The GHMs are trained for 500,000 update steps with the CETD loss, using the Adam optimiser and a learning rate of 0.0003.

We performed a light hyperparameter search for: the learning rate between $\{3 \cdot 10^{-5}, 10^{-4}, 3 \cdot 10^{-4}, 10^{-3}\}$; the β_{VAE}-parameter in the β-VAE loss between $\{1, 20, 50, 100\}$; and the VAE latent dimension between $\{1, 8\}$. Performance in terms of negative ELBO was fairly robust across this range of hyperparameters.

Both the policy and GHM training use a distributed actor-learner setup communicating via a uniform replay buffer of size 10^6. Each learner step uses a batch size of 256 and averages the loss over trajectories of length 20. These settings are conducted without a target specified, and episodes last for between 100 and 140 steps uniformly at random.

GGPI. When performing GGPI to improve on this set of policies, we evaluate with a discount factor of $\gamma = 0.9$. Thus, we can consider geometric switching policies that switch with a probability of $\alpha = 1/9$, and estimate the GHM of such a policy using the GHMs of its constituent base policies with $\beta = 0.8$ and $\beta = 0.9$. We estimate the action-value function by sampling from the composed GHM, evaluating the sample under the known non-linear reward function r, and averaging over multiple samples per geometric switching policy. For fairness, when comparing GGPI with $n = 1$ and $n = 2$, we sample 4 times the sampling budget when evaluating $n = 1$ — thus, both $n = 2$ and $n = 1$ GGPI are considering the same number of *total* samples, with $n = 1$ actually seeing more samples per policy.

Since this environment has a continuous action space, we cannot evaluate $Q^\nu(x, a)$ for all actions; thus, instead we estimate $V^\nu(x) = \mathbb{E}_{A\sim\pi_1(\cdot|x)}[Q^\nu(x, A)]$, i.e. consider only those actions that we obtain by sampling from the head-policy π_1 of the GSP ν, and use this to choose the best GSP $\nu \in \Pi$ and act according to it per time step.

E.2. Additional agent visualisations

Figures 15 and 16 show more detailed visualisations of the GPI and depth-2 GGPI agents.

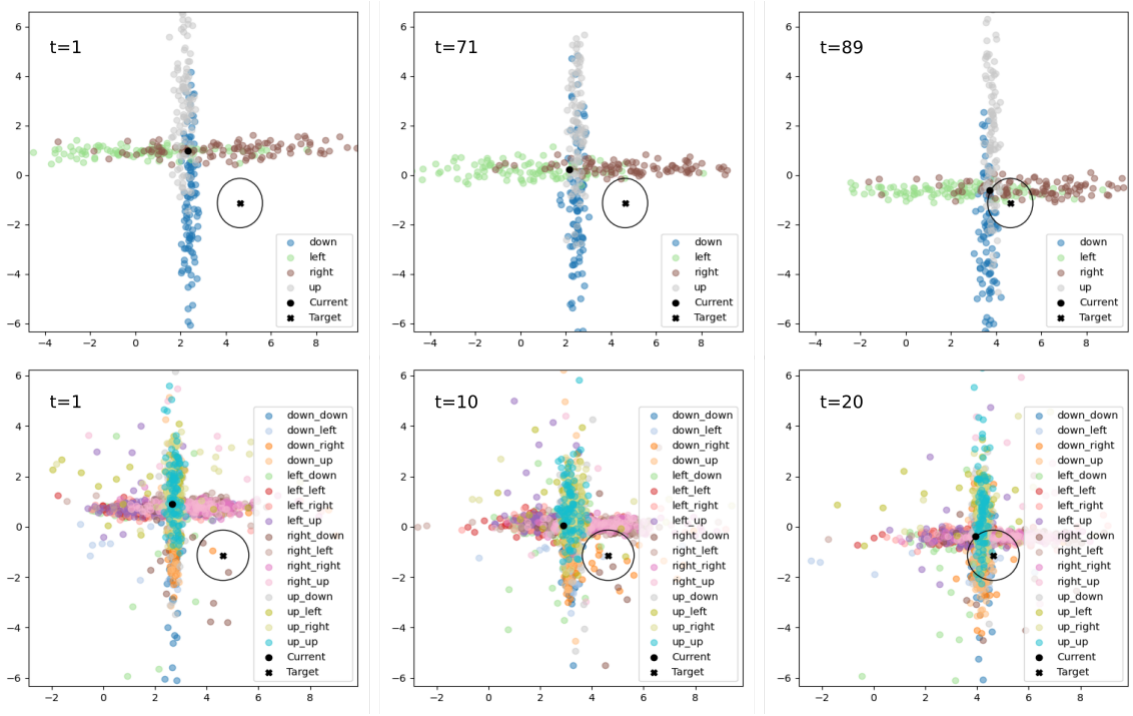


Figure 15. Visualisation of an agent performing GPI (top) and depth-2 GGPI (bottom) for the same episode initialisation. While GGPI is immediately able to plan to reach the target and reaches it within 20 steps, standard GPI is unable to do so and spends 70 steps moving randomly before happening to align with the target through pure chance, and then reaching the target. We show the agent and target locations, the boundary outside of which the reward signal is zero, and visualisation of the different agent plans.

E.3. Comparing GHMs to compositions of 1-step models

We compare the use of GHMs against a 1-step model unrolled for multiple steps. Concretely, we compare our our VAE-based GHM(β) against a one-step model unrolled for a Geometric($1 - \beta$) number of steps. Note that for perfectly trained models, these two distributions would be identical; however, the GHM models show compounding error at train time due to the use of bootstrapping, while the one-step model shows compounding error at evaluation time due to the multi-step composition. We implement the one-step model with identical architecture as the GHM model, equivalent to a GHM($\beta = 0$) model.

Figure 17 shows a comparison of these models for $\beta = 0.9$, versus the true geometrically discounted future state distribution obtained through sampling trajectories via simulating the policy in the environment. The scatterplot on the left shows samples from these models compared with the true distribution, showing a high degree of overlap for the GHM with the true distribution and large errors for the one-step model. The plot on the left measures distance of a sample from these models versus the nearest simulated future state. Though the 1-step model has lower error for very near-term predictions, its error quickly compounds and increases steadily as the prediction horizon increases. Meanwhile, the GHM models make low-error predictions even far into the future.

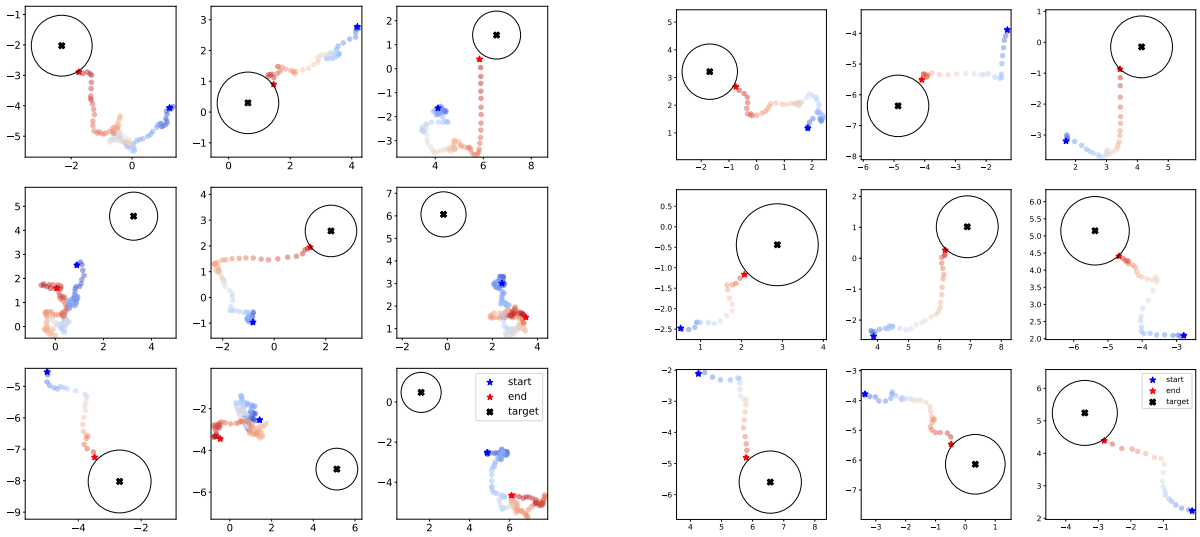


Figure 16. Comparison of standard GPI (left) and $n = 2$ GGPI (right) to navigate towards a goal given sparse rewards. We plot the x-y coordinates of the agent centre of mass, and colour on a gradient from blue to red through the duration of the episode. The target is displayed along with the boundary outside of which the reward signal is zero.

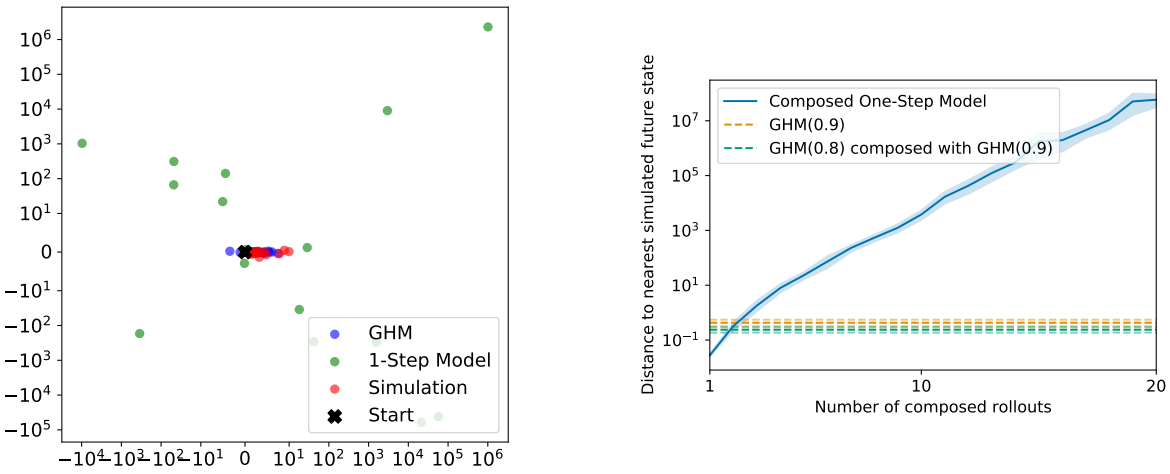


Figure 17. Comparison of a true discounted future state visitation distribution, a learnt GHM(β), and a 1-step model composed for a Geometric($1 - \beta$) number of steps, for $\beta = 0.9$. The plot on the right is averaged over 100 random seeds.

E.4. GHM training using normalising flows

In this section we provide further details of GHM training experiments that inform our choice of VAE-GHMs and the CETD loss in the main paper. One of our main points of comparison is the L^2 loss on log-densities, introduced by Janner et al. (2020).

Definition E.1 (Log-L^2 temporal-difference (LL2TD) loss (Janner et al., 2020))**.** Given an observed transition (x, a, x'), the log-L^2 bootstrap loss is defined by

$$(\log(\mu(x''|x,a)) - \log((1-\beta)P(x''|x,a) + \beta\bar{\mu}(x''|x',a')))^2\,,$$

where $a' \sim \pi(\cdot|x')$, $x'' \sim \mu(\cdot|x',a')$, $\bar{\mu}$ denotes a stop-gradient on μ.

Janner et al. (2020) focus on the LL2TD bootstrap loss in their experiments. However, as they note, this loss generally leads to an incorrect minimiser, due to the presence of bias (specifically, the averaging of x', a', x'' *outside* rather than inside the logarithm).

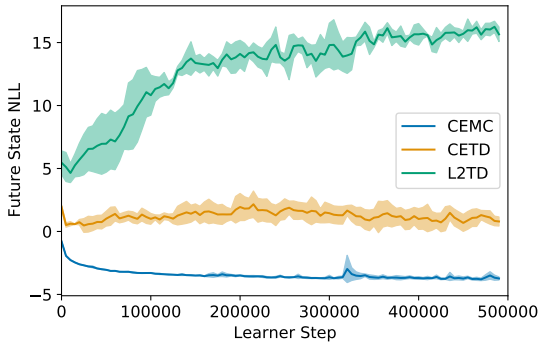


Figure 18. Comparison of CEMC, CETD, and LL2TD losses for training GHMs implemented as normalising flows (left) and β-VAEs (right). Performance (lower is better) is measured in terms of estimated negative log-likelihood of samples from the true geometric horizon distribution obtained by sampling states from on-policy trajectories. Results are shown averaged over 5 random seeds.

We provide an empirical comparison of the different methods for GHM training introduced in Section 6. We primarily consider the LL2TD loss proposed by Janner et al. (2020), against the CETD loss analysed previously. We also briefly consider training the GHM model μ by directly sampling a future state from on-policy trajectories at a time sampled according to a Geometric$(1-\beta)$ distribution into the future, following the CEMC loss introduced in the main paper. Note that the CEMC loss is straightforward to implement and does not require any bootstrapping, but cannot be learned from off-policy samples and thus is less desirable than the other methods.

In addition to the comparison using the much simpler VAE models in Section 7, here we compare the losses using normalising flow models (Rezende & Mohamed, 2015) of a similar architecture as suggested by Janner et al. (2020). As these models admit exact density computation, we also can compare against the LL2TD loss. Figure 18 shows the performance of the different methods for $\beta = 0.8$ in terms of the negative log-likelihood of a sample from the true geometric horizon distribution of the π_{right} pretrained policy on the sparse-reward ant environment. Note that the CEMC loss is explicitly optimising for this metric and achieves very strong performance, while the CETD and LL2TD losses perform much worse. Further, the LL2TD loss is much less stable than CETD and actually diverges late in training.

When training GHMs using normalising flows, we use a similar architecture to that proposed by Janner et al. (2020). We use a normalising flow consisting of 2 coupling layers, each including a batch norm flow (Dinh et al., 2017), a 1x1 invertible convolution (Kingma & Dhariwal, 2018), and a conditional neural spline (Durkan et al., 2019). The neural spline includes a rational quadratic spline with range between -5 to 5, 8 knots, and whose parameters are outputted by an MLP with a single hidden layer of size 256. When training using the LL2TD loss, we use a target network with a target update period of 200 learner steps to generate the bootstrap targets as suggested by Janner et al. (2020).

E.5. Evaluating GGPI performance for varying GHM training budgets

In our main experiments, we train GHMs using VAEs for 500000 learner steps, which is sufficient to plateau the training ELBO. We now examine the sensitivity of our proposed method to the training budget afforded to GHM training.

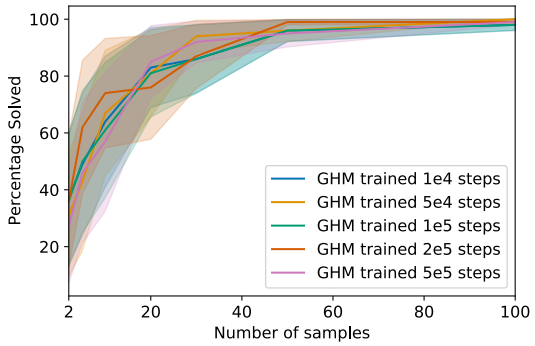


Figure 19. Performance of depth-2 GGPI on the sparse-reward ant task with snapshots of GHM models taken at various stages through training.

Figure 19 shows that GHMs trained with as few as 10^4 learner steps (taking only 2 minutes wall-clock time on our distributed training setup described in Appendix E.1) are still successful in planning. Additional preliminary experiments with even fewer learner steps did not result in GHMs useful for planning.

F. An extension of the main evaluation result

For simplicity, in the main paper we presented Theorem 3.2 for action-independent rewards. There is a simple adaptation to this result that applies to general reward functions $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$.

Theorem F.1. Consider an MDP with expected reward function $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, and let $\nu = \pi_1 \overset{\alpha}{\to} \cdots \overset{\alpha}{\to} \pi_n$. Writing $\beta = \gamma(1-\alpha)$, we have

$$r(x,a) + \frac{\gamma}{1-\gamma} \times \left[\sum_{m=1}^{n-1} \frac{1-\gamma}{1-\beta} \left(\frac{\gamma-\beta}{1-\beta} \right)^{m-1} \left((1-\alpha) r^{\pi_m}(X^{(m)}) + \alpha r^{\pi_{m+1}}(X^{(m)}) \right) + \left(\frac{\gamma-\beta}{1-\beta} \right)^{n-1} r^{\pi_n}(X') \right],$$

where $(X^{(0)}, A^{(0)}) = (x,a)$, $X^{(m)} \sim \mu_\beta^{\pi_m}(\cdot|X^{(m-1)}, A^{(m-1)})$, $A^{(m)} \sim \pi_{m+1}(\cdot|X^{(m)})$, $X' \sim \mu_\gamma^{\pi^{(n)}}(\cdot|X^{(n)}, A^{(n)})$, is an unbiased estimator for $Q_\gamma^\nu(x,a)$.

Proof. The result can be proven as a straightforward corollary of Theorem 3.2; the distribution of $X^{(m)}$ matches that of X_T conditional on the geometric time T falling between the times of the $(m-1)^{\text{th}}$ and m^{th} switches, while the GSP is executing policy π_m. Thus, to know what the distribution of actions should be when evaluating the reward function at this state, we need to know whether the switch happens at the current time step or not. From the memoryless property of the geometric distribution concerned, this probability is precisely α. So with a weighting of $1-\alpha$, the reward is evaluated for the policy π_m, and with a weighting of α, the reward is evaluated according to the distribution π_{m+1} over policies that will be switched to at this time step. □