General Value Discrepancies Mitigate Partial Observability in Reinforcement Learning

Peter Koepernik[1,2,*], **Ruo Yu Tao**[3,*], **Ronald Parr**[4],
George Konidaris[3,†], **Cameron Allen**[1,†]
peter.koepernik@stcatz.ox.ac.uk, ruo_yu_tao@brown.edu

[1]**UC Berkeley**, [2]**University of Oxford**, [3]**Brown University**, [4]**Duke University**

[*]Equal contribution, [†]Equal advising

Abstract

In most realistic sequential decision-making tasks, an agent only observes partial and noisy information about the state of its environment, and must learn to summarize its history for optimal decision-making. Past work has leveraged discrepancies over different TD(λ) value function estimates to reveal and mitigate partial observability. While effective in many cases, the so-called λ-discrepancy crucially relies on the reward signal to gauge partial observability. We introduce the General Value Discrepancy (GVD), a principled extension of the λ-discrepancy that computes discrepancies over arbitrary, observable features using the frameworks of general value functions and successor features. Our key theoretical contribution is a proof that—unlike the λ-discrepancy—GVD can always detect partial observability if it exists, irrespective of the environment's reward structure. By minimizing GVD as an auxiliary objective in deep reinforcement learning, we create a dense and robust learning signal that improves agent performance in a range of challenging partially observable benchmarks.

1 Introduction

Sequential decision-making tasks in the real world typically involve partial observations, where an agent does not directly perceive the underlying environment state but instead receives incomplete and noisy information. Such an environment is most commonly modeled as a POMDP (Kaelbling et al., 1998), the partially observable generalization of a Markov Decision Process (MDP; Puterman, 2014).

A common approach to solving a POMDP is for the agent to augment the observations it receives with a suitable summary of its history—a *memory*. If it retains enough information about its past to recover a Markov state representation, then the optimal memory-conditioned policy is guaranteed to be as good as if it were allowed to condition on the full history (Puterman, 2014). A recent approach to learning such memory functions, the λ-*discrepancy* (Allen et al., 2024), exploits the fact that different ways of estimating a value function—e.g. through Monte-Carlo regression or temporal difference learning—generally only coincide in fully observable settings, whereas discrepancies arise in the presence of partial observability. Minimizing the λ-discrepancy (LD) as an auxiliary goal helps the agent learn a memory function that mitigates the partial observability.

While effective in many environments, a weakness of the LD is its crucial reliance on the reward. Its utility is limited in environments with sparse or no rewards, and even if rewards are dense it may forego potentially informative learning signals from other observables. In this paper, we extend the LD by introducing *general value discrepancies* (GVDs), which measure differences between value functions defined over arbitrary observable features and, optionally, with observation-dependent discounting.

Our contributions are as follows.

(1) We introduce and formalize the general value discrepancy. We prove that, in contrast to the λ-discrepancy, when partial observability exists, it can always be detected with general value discrepancies, regardless of the reward structure.

(2) We empirically validate and test the efficacy of general value discrepancies in tabular, closed-form experiments, as well as in a deep reinforcement learning setting. We show that, as predicted by our theoretical results, GVDs can detect and mitigate partial observability in all of the tested tabular environments, including those in which LD fails. We also present initial results showing that GVD can improve deep RL algorithms in complex, partially observable tasks.

2 Background

A *partially observable Markov decision process* (POMDP) is a tuple $(\mathcal{S}, \mathcal{A}, \Omega, T, \Phi, R, p_0, \gamma)$, where $\mathcal{S}, \mathcal{A}$, and Ω are the state, action, and observation spaces, $T\colon \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}$ is the transition function, $\Phi\colon \mathcal{S} \to \Delta\Omega$ is the observation function, $R\colon \Omega \to \mathbb{R}$ is the reward function, $p_0 \in \Delta\mathcal{S}$ is the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor. We assume that rewards $r = R(\omega)$ are contained in observations; this assumption does not restrict generality (Thrun, 1999; Sutton & Barto, 2018; Jiang et al., 2017; Liu et al., 2022), and in practice rewards can be appended to subsequent observations before being passed to the policy. We further assume that $\mathcal{S}$, $\mathcal{A}$, and Ω are finite; this assumption is for convenience, and is not essential. An agent following a policy $\pi\colon \Omega \to \Delta\mathcal{A}$ starts in $s_0 \sim p_0$, and at each timestep $t \in \mathbb{N}_0$ receives observation $\omega_t \sim \Phi(\cdot \mid s_t)$ and reward $r_t = R(\omega_t)$, chooses action $a_t \sim \pi(\cdot \mid \omega_t)$, and transitions to state $s_{t+1} \sim T(\cdot \mid s_t, a_t)$, until reaching a designated terminal state $s_H = s_{\text{terminal}}$. The timestep H is the *length* of the episode, which we set to $H = \infty$ if the terminal state is never reached. In this paper we assume that there exists some deterministic $H_{\max} \in \mathbb{N}$ such that $H \leq H_{\max}$ almost-surely. If the agent has access to the underlying state[1] then we recover the well-known definition of a *Markov decision process* (MDP).

The agent seeks to select actions that maximize the expectation of the discounted cumulative reward, called the *return*, $g_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$, where $r_i := 0$ for $i > H$. Choosing optimal actions in POMDPs requires memory, because the full history $(\omega_0, a_0, \ldots, a_{t-1}, \omega_t)$ generally contains more information about the underlying state s_t than the current observation ω_t alone. Let $\mathcal{M}$ be the memory space, and let $\mu\colon \mathcal{M} \times \Omega \times \mathcal{A} \to \Delta\mathcal{M}$ be the memory update function. The agent maintains $m_t \in \mathcal{M}$, selects actions $a_t \sim \pi(\cdot \mid \omega_t, m_t)$, and updates memory as $m_{t+1} \sim \mu(\cdot \mid m_t, \omega_t, a_{t+1})$. The *memory-augmented POMDP* is the POMDP with augmented observations $\tilde{\omega}_t = (m_t, \omega_t)$.

> **Definition 1.** We say that the memory *resolves partial observability* if $(\tilde{\omega}_0, \tilde{\omega}_1, \ldots)$ is a Markov chain *for all* policies.

In that case, an optimal policy can be written as a deterministic function of $\tilde{\omega}_t$ (Puterman, 2014), where by optimal we mean as well as we could possibly do with access to (histories of) observations alone. Moreover, if the memory does not resolve partial observability, then the observation trajectory is not a Markov chain for *all* policies except those in a set of measure zero.

An idea motivated by this fact is to define a metric that is positive if the sequence of (memory-augmented) observations encountered by an agent is non-Markovian, and using its minimization as an auxiliary objective to learn a memory that resolves partial observability. Allen et al. (2024) proposed the λ-*discrepancy*, a method based on this idea that improves deep RL performance in partially observable settings. In the next section, we explain their approach in detail, analyze its limitations, and introduce a generalization that rectifies these shortcomings.

[1] This could be formally achieved within the POMDP framework by setting $\Omega = \mathcal{S}$ and $\Phi(s' \mid s) = \mathbf{1}_{\{s=s'\}}$.

3 Value Discrepancies

The idea behind λ-discrepancy is as follows: in an MDP, the value function $V^\pi(s) = \mathbb{E}^\pi[g_t \mid s_t = s]$ is the unique fixed point of the Bellman equation $V^\pi(s) = \mathbb{E}^\pi[r_t + \gamma V^\pi(s_{t+1}) \mid s_t = s]$, so two (theoretically) equivalent methods for estimating V^π are Monte Carlo (MC) regression and temporal-difference (TD) learning. In a POMDP, the MC estimator V^π_{MC} still converges to the value function, and the TD estimator V^π_{TD} still converges to the fixed point of the Bellman equation, but the two generally no longer coincide. Their difference, $\Lambda = \|V^\pi_{\text{TD}} - V^\pi_{\text{MC}}\|_2$, or more generally the discrepancy between two TD(λ) value functions for different λ (with $\lambda = 0$ yielding V^π_{TD} and $\lambda = 1$ yielding V^π_{MC}), defines the *λ-discrepancy* (LD) introduced by Allen et al. (2024). We focus on the case $\lambda_0 = 0$, $\lambda_1 = 1$ in our theoretical analysis for simplicity, but we expect our results to generalize, and we conduct our experiments in the general setting (see Section 5).

We will now introduce a theoretical framework that allows us to (1) precisely analyze the limitations of LD, (2) define a natural generalization, and (3) prove that this generalization overcomes the limitations of LD. We start by analyzing the MC and TD value estimators in a POMDP.

Value estimators in POMDPs. The notion of a value function in a POMDP is ambiguous, since, unlike in an MDP, $\mathbb{E}^\pi[g_t \mid \omega_t = \omega]$ generally depends on t. One resolution is to let the value function depend on t, but in practice this would require training a separate value function for every timestep. A more practical approach is to consider the MC regression target $\sum_{(\omega_t, g_t)\in\mathcal{D}} \sum_{t=0}^{H} \left|\hat{V}(\omega_t) - g_t\right|^2$ given a dataset $\mathcal{D}$ of episodes collected with a fixed policy π. This loss is minimized in the infinite-data limit by

$$V^\pi_{\text{MC}}(\omega) = \sum_{s,a} W^\pi(s \mid \omega)\pi(a \mid \omega)Q^\pi(s,a), \tag{1}$$

where $W^\pi(s \mid \omega) = \frac{\mathbb{E}^\pi\left[\sum_{t=0}^{H} \mathbf{1}_{\{s_t = s, \omega_t = \omega\}}\right]}{\mathbb{E}^\pi\left[\sum_{t=0}^{H} \mathbf{1}_{\{s_t = s\}}\right]}$ is the probability that the state underlying an observation ω drawn uniformly from $\mathcal{D}$ is s, and $Q^\pi(s,a) = \mathbb{E}[g_t \mid s_t = s, a_t = a]$ (independent of t) is the state-conditioned Q-function. Eq. (1) implies the following.

Lemma 1 (MC). *V^π_{MC} is the value function of the POMDP, in that $V^\pi_{\text{MC}}(\omega)$ is the expected return of an agent acting according to π that starts in a state sampled from $W^\pi(\cdot \mid \omega)$.*

For TD learning, we would iteratively minimize $\sum_{\mathcal{D}} \sum_{t=1}^{H} |\hat{V}^{(i+1)}(\omega_t) - [r_t + \gamma \hat{V}^{(i)}(\omega_{t+1})]|^2$, which by Banach's fixed point theorem converges in the infinite-data limit to the unique fixed point of

$$V^\pi_{\text{TD}}(\omega) = \sum_{s,a,s',\omega'} W^\pi(s \mid \omega)\pi(a \mid \omega)T(s' \mid s,a)\Phi(\omega' \mid s')\Big(R(\omega) + \gamma V^\pi_{\text{TD}}(\omega')\Big). \tag{2}$$

Eq. (2) is the Bellman equation (and therefore V^π_{TD} is the value function) of an MDP with state space Ω that "forgets" the hidden state after every transition: after observing $\omega^{(t)}$, we sample $s^{(t)} \sim W^\pi(\cdot \mid \omega^{(t)})$, act, transition, obtain $\omega^{(t+1)}$, and then *resample* $s^{(t+1)} \sim W^\pi(\cdot \mid \omega^{(t+1)})$ and repeat (see Fig. 2a)[2]. Following Allen et al. (2024) we call this the *effective MDP*. Intuitively, the effective MDP is the result of trying to "fit" an MDP to a dataset of transitions (ω, a, ω') sampled from the POMDP. An illustration for a simple environment called T-Maze is in Fig. 1. To summarise:

Lemma 2 (TD). *V^π_{TD} is the value function of the effective MDP.*

Measuring Markovianity of observation trajectories. Denote by $P^\pi_{\text{MC}}(\cdot \mid \omega)$ and $P^\pi_{\text{TD}}(\cdot \mid \omega)$ the distributions over observation trajectories $(\omega = \omega^{(0)}, \omega^{(1)}, \ldots)$ in the POMDP and in the effective MDP, respectively (see Fig. 2a for an illustration, and Appendix A for details).

Lemma 3. *The observation trajectory $(\omega^{(t)})$ is Markov under π in the POMDP iff $P^\pi_{\text{MC}} = P^\pi_{\text{TD}}$.*

Proof. By definition of the effective MDP, $P^\pi_{\text{TD}}(\omega^{(t)} \mid \omega^{(t-1)}, \ldots, \omega^{(0)}) = P^\pi_{\text{MC}}(\omega^{(t)} \mid \omega^{(t-1)})$, so $P^\pi_{\text{MC}} = P^\pi_{\text{TD}}$ iff P^π_{MC} is itself already Markov. □

[2]We use super- rather than subscripts because $\omega = \omega^{(0)}$ need not necessarily be the first observation ω_0 of an episode.

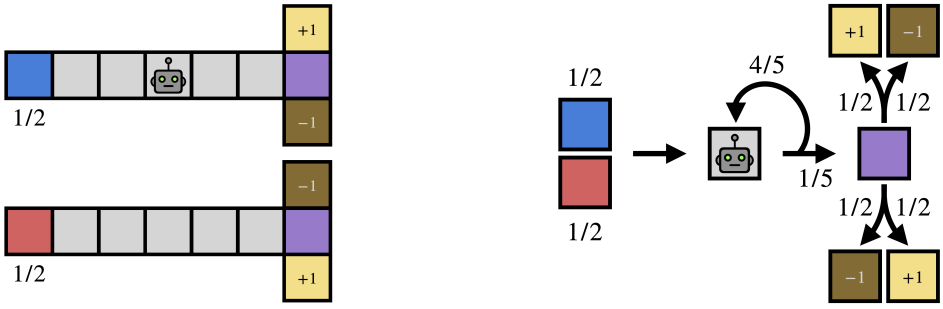
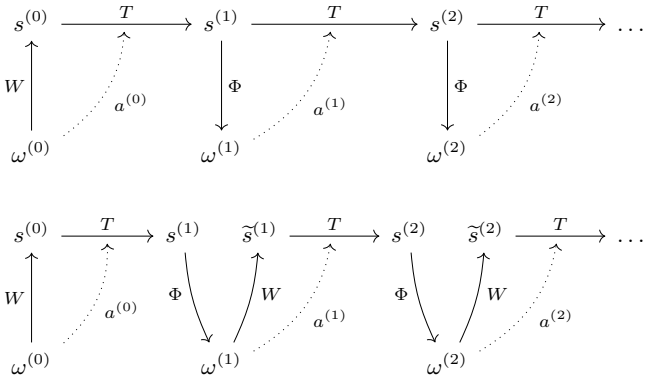
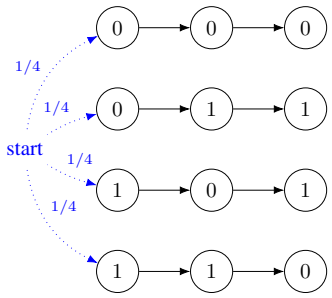


Figure 1: T-Maze (left) is an environment in which an agent receives a blue or red observation at the beginning, depending on which it has to move up or down at the end of a corridor to receive a positive reward. The colours of states in the illustration indicate the observation the agent receives. The effective MDP of T-Maze does not depend on the agent's policy and is illustrated on the right.



(a) Observation trajectories under P^{π}_{MC} and P^{π}_{TD}.



(b) Parity Check

Figure 2: (Left) Illustration of how observation trajectories $(\omega = \omega^{(0)}, \omega^{(1)}, \omega^{(2)}, \dots)$ are sampled from $P^{\pi}_{\text{MC}}(\cdot \mid \omega)$ (top) and $P^{\pi}_{\text{TD}}(\cdot \mid \omega)$ (bottom). (Right) Illustration of the POMDP in Example 1.

Hence, measuring the gap between P^{π}_{MC} and P^{π}_{TD} quantifies non-Markovianness of the observation trajectory. We cannot access P^{π}_{TD} and P^{π}_{MC} directly, but we can estimate certain expectations (like value functions). Therefore, a convenient family of measures for this task is the *maximum mean discrepancy* (MMD; Gretton et al., 2012), $\text{MMD}_{\mathcal{F}}\left(P^{\pi}_{\text{TD}}, P^{\pi}_{\text{MC}}\right) = \sup_{F \in \mathcal{F}} \left\| \mathbb{E}^{\pi}_{\text{MC}}[F \mid \cdot] - \mathbb{E}^{\pi}_{\text{TD}}[F \mid \cdot] \right\|_2$, where $\mathcal{F}$ is some class of functions on the domain $\Omega^{\star} = \{\omega^{(0:H)} \colon H \leq H_{\max}\}$ of variable-length observation sequences, $\mathbb{E}[F \mid \cdot]$ abbreviates the function $\omega \mapsto \mathbb{E}[F \mid \omega]$, and $\| \cdot \|_2$ is the L^2-norm.

The λ-discrepancy. We can recast LD from this perspective: by Lemmas 1 and 2, it can be written as

$$\Lambda^{\pi}_{\gamma} = \|V^{\pi}_{\text{MC},\gamma} - V^{\pi}_{\text{TD},\gamma}\|_2 = \left\| \mathbb{E}^{\pi}_{\text{MC}}[F_{\gamma} \mid \cdot] - \mathbb{E}^{\pi}_{\text{TD}}[F_{\gamma} \mid \cdot] \right\|_2, \qquad F_{\gamma}(\omega^{(0:H)}) = \sum_{t=0}^{H} \gamma^t R(\omega^{(t)}),$$

where we make dependencies on γ explicit. That is, LD is exactly the mean discrepancy between P^{π}_{MC} and P^{π}_{TD} for functions taken from the class $\mathcal{F} = \{F_{\gamma}\}_{\gamma \in (0,1)}$ of discounted returns. In particular, LD vanishes (for every choice of γ) exactly if $\text{MMD}_{\mathcal{F}}(P^{\pi}_{\text{TD}}, P^{\pi}_{\text{MC}}) = 0$. This observation can be used to derive the following characterization, which we prove in Appendix A.

Theorem 4 (LD). *The λ-discrepancy satisfies $\Lambda^{\pi}_{\gamma} = 0$ for every $\gamma \in (0,1)$ if and only if the marginal expected rewards at every timestep coincide in the effective MDP and the POMDP, that is*

$$\mathbb{E}^{\pi}_{\text{MC}}\left[r^{(t)} \,\middle|\, \omega^{(0)} = \omega \right] = \mathbb{E}^{\pi}_{\text{TD}}\left[r^{(t)} \,\middle|\, \omega^{(0)} = \omega \right], \quad \forall \omega \in \Omega, t \in \mathbb{N}_0. \tag{3}$$

Otherwise, $\Lambda^{\pi}_{\gamma} \neq 0$ for all $\gamma \in (0,1)$ except those in a set of measure zero.

General Value Discrepancies. We can increase the discriminative power of the MMD by expanding the underlying class $\mathcal{F}$ of functions $F\colon \Omega^\star \to \mathbb{R}$. An obvious first step is to replace the reward by an arbitrary observable feature $f\colon \Omega \to \mathbb{R}$, leading to the discounted "pseudo-return" $F_{f,\gamma}(\omega^{(0:H)}) = \sum_{t=0}^{H} \gamma^t f(\omega^{(t)})$. The resulting discrepancy

$$\Lambda^\pi_{f,\gamma} := \left\| \mathbb{E}^\pi_{\mathrm{MC}} \left[\sum_{t=0}^{H} \gamma^t f(\omega^{(t)})^t \,\middle|\, \cdot \right] - \mathbb{E}^\pi_{\mathrm{TD}} \left[\sum_{t=0}^{H} \gamma^t f(\omega^{(t)})^t \,\middle|\, \cdot \right] \right\|_2 \tag{4}$$

generalizes LD (recovered with $f = R$) and we call it the *general value discrepancy* (GVD). There is no reason to restrict to a single f; given several functions $f_1, \dots, f_n$, we can consider the combined discrepancy $\sum_{i=1}^n \Lambda^\pi_{f_i,\gamma}$. For example, the combined discrepancy of the functions of the form $f = \mathbf{1}_{\{\omega = \cdot\}}$ for $\omega \in \Omega$ equals the discrepancy between successor representations (Dayan, 1993; Kulkarni et al., 2016) of the POMDP and the effective MDP, and we refer to it as the *successor representation discrepancy* (SR-GVD). For general functions (features) $f\colon \Omega \to \mathbb{R}$, we analogously retrieve discrepancies between successor features (Barreto et al., 2017; 2019), which we call *successor feature discrepancies* (SF-GVD).

Theorem 5 (GVD, I)**.** *The generalized value discrepancy satisfies $\Lambda^\pi_{f,\gamma} = 0$ for all $f\colon \Omega \to \mathbb{R}$ (equivalently, for all f of the form $\mathbf{1}_{\{\omega=\cdot\}}$) and all $\gamma \in (0,1)$ if and only if the observation marginals coincide in the effective MDP and the POMDP, that is*

$$P^\pi_{\mathrm{MC}}\left(\omega^{(t)} = \cdot \,\middle|\, \omega\right) = P^\pi_{\mathrm{TD}}\left(\omega^{(t)} = \cdot \,\middle|\, \omega\right), \quad \forall \omega \in \Omega, t \in \mathbb{N}_0. \tag{5}$$

Otherwise, $\Lambda^\pi_{f,\gamma} \neq 0$ for all f, γ except those in a set of measure zero.

See Appendix A for the statement and proof of a slightly stronger version of Theorem 5. Note that Theorem 5 in particular implies that the SR-GVD has full discriminative power, and hence should always be used if Ω is sufficiently small; see also Section 4, where we conduct closed-form experiments in small tabular POMDPs using SR-GVD.

This form of the GVD is robust to settings with sparse or no rewards, but still fails in cases where observations in P^π_{MC} and P^π_{TD} have the same marginals, and their difference only shows in correlations. An example is the parity check environment of Allen et al. (2024), which we describe here.

> **Example 1** (Parity Check)**.** Consider a POMDP in which the agent observes a sequence of three fair coin flips c_1, c_2, c_3, where c_1 and c_2 are independent, and c_3 is the XOR of c_1 and c_2. After seeing c_2, the agent is asked to predict c_3 and receives a reward of $+1$ (resp. -1) if it predicts c_3 correctly (resp. incorrectly). Since c_3 is independent of c_2 (and of c_1, but not of both together), P^π_{TD} is a sequence of three independent coin flips. Thus $P^\pi_{\mathrm{TD}} \neq P^\pi_{\mathrm{MC}}$, but the marginals of c_1, c_2, c_3 are fair coin flips under both. By Theorem 5, $\Lambda^\pi_{f,\gamma} = 0$ for all f, γ, that is, both LD and GVD fail to detect the partial observability.

The remedy proposed by Allen et al. (2024) for the parity check environment is to randomly initialize memory to break symmetry. Our framework (1) explains precisely why this helps, and (2) reveals a potential issue: the memory m_3 is a random function of c_1 and c_2 that causes the memory-augmented observation $\tilde{\omega}_3 = (m_3, c_3)$ at timestep three to have different distributions under P^π_{MC} and P^π_{TD}, even though c_3 remains a coin flip under both. This allows GVD and LD to detect the partial observability. However, if the memory is trained to minimize discrepancy after random initialization, it might either learn to encode the XOR of c_1 and c_2 to resolve the partial observability, or collapse to a trivial memory, as both LD and GVD are zero in the absence of memory. This is one explanation for the non-trivial but poor performance of LD on parity check in our closed form experiments in Fig. 3.

Observation-dependent discounting. To further increase the discriminative power of the discrepancy and remove aliasing completely, we must further expand the class of functions $F\colon \Omega^\star \to \mathbb{R}$ while keeping $\mathbb{E}^\pi_{\mathrm{MC}}[F \mid \omega]$ and $\mathbb{E}^\pi_{\mathrm{TD}}[F \mid \omega]$ tractable to estimate. Monte Carlo regression allows estimating

$\mathbb{E}^\pi_{\text{MC}}[F \mid \omega]$ without restrictions on F, but $\mathbb{E}^\pi_{\text{TD}}[F \mid \omega]$ can only be estimated with TD, which requires F to have a recursive structure: $F(\omega, \omega', \ldots) = h(\omega, F(\omega', \ldots))$ for some function $h\colon \Omega \times \mathbb{R} \to \mathbb{R}$. Then $V^\pi_F(\omega) \coloneqq \mathbb{E}^\pi_{\text{TD}}[F \mid \omega] = \mathbb{E}^\pi_{\text{TD}}[h(\omega, F(\omega', \ldots)) \mid \omega]$, and to pull the expectation inside the second argument of h to allow bootstrapping, we further need h to be linear in its second argument. The most general such form is $h(\omega, x) = f(\omega) + \gamma(\omega)\, x$ for some $f, \gamma\colon \Omega \to \mathbb{R}$, which yields the class of functions

$$F_{f,\gamma}(\omega^{(0:H)}) = \sum_{t=0}^{H} \gamma(\omega^{(0)}) \cdots \gamma(\omega^{(t-1)}) f(\omega^{(t)}). \tag{6}$$

Then the Bellman equation for V^π_F is $V^\pi_F(\omega) = \mathbb{E}^\pi_{\text{TD}}\Big[f(\omega) + \gamma(\omega) V^\pi_F(\omega') \,\Big|\, \omega\Big]$, which can be used for learning V^π_F with TD. If γ only takes values in $(0, 1)$, then $F_{f,\gamma}$ is a discounted pseudo-return with an observation-dependent discount factor, so V^π_F is a *general value function* (GVF) in the language of Sutton et al. (2011). Write

$$\Lambda^\pi_{f,\gamma} \coloneqq \Big\| \mathbb{E}^\pi_{\text{MC}}\left[F_{f,\gamma} \mid \cdot\right] - \mathbb{E}^\pi_{\text{TD}}\left[F_{f,\gamma} \mid \cdot\right] \Big\|_2 \tag{7}$$

for functions $f\colon \Omega \to \mathbb{R}$ and $\gamma\colon \Omega \to (0, 1)$. We recover the earlier form of GVD from (4) when γ is constant. Our main result is that the GVD with observation-dependent discounting is a perfect discriminator.

Theorem 6 (GVD, II)**.** *The GVD satisfies $\Lambda^\pi_{f,\gamma} = 0$ for all $f\colon \Omega \to \mathbb{R}$ (equivalently, for all f of the form $\mathbf{1}_{\{\omega = \cdot\}}$) and all $\gamma\colon \Omega \to (0, 1)$ if and only if $P^\pi_{\text{MC}} = P^\pi_{\text{TD}}$.*

That is, GVD can always detect a difference between P^π_{MC} and P^π_{TD} if there is one, and therefore—via Lemma 3—provides a (theoretically) perfect signal for memory learning. We remark that, while Theorems 4 and 5 are relatively elementary to prove, Theorem 6 is significantly more involved and requires a mix of probabilistic and combinatorial arguments as well as abstract theorems from measure theory and topology to prove. See Appendix A.

4 Memory Learning with General Value Discrepancies

The GVD measures how non-Markovian observation trajectories are in a given POMDP, and, much like LD, we can use it as a signal to learn a memory function that makes the memory-augmented trajectories Markov (and thereby resolves partial observability, recall Definition 1).

Consider an agent in a POMDP $\mathcal{P} = (\mathcal{S}, \mathcal{A}, \Omega, T, \Phi, R, p_0, \gamma)$ with memory states $\mathcal{M}$, initialized at m_0 and updated via $m_{t+1} \sim \mu(\cdot \mid m_t, \omega_t, a_{t+1})$, which induces a memory-augmented POMDP $\mathcal{P}^\mu$ with observations $\tilde{\omega}_t = (\omega_t, m_t)$. Applying GVD to $\mathcal{P}^\mu$ gives a measure for how close the memory μ is to inducing Markov trajectories. If we can compute gradients of the GVD of $\mathcal{P}^\mu$ with respect to the memory μ, then we can *learn* a memory μ by descending those gradients.

We demonstrate this in a range of small tabular POMDPs, where the full successor representation discrepancy (SR-GVD) and its gradients are analytically tractable. This allows us to evaluate GVD in isolation, without confounding factors like approximation error or sampling variance. Experimental details are in Appendix B; results in Figure 3 show that, as predicted by our theory, GVD performs consistently across all tested environments, while LD fails in parity check (Example 1). Somewhat surprisingly, GVD also performs significantly better than LD in T-Maze (recall Fig. 1), even though LD is theoretically capable of solving this environment. An explanation might be that, in this particular environment, GVD provides a more stable signal and/or has a smoother loss landscape.

5 Learning in Sparse, Partially Observable Domains

Since GVD is defined using only observable quantities, it can be estimated from samples via function approximation with deep neural networks. Like LD, GVD can then be used as an auxiliary loss to train a recurrent policy, e.g., with PPO (Schulman et al., 2017). In most deep RL environments,

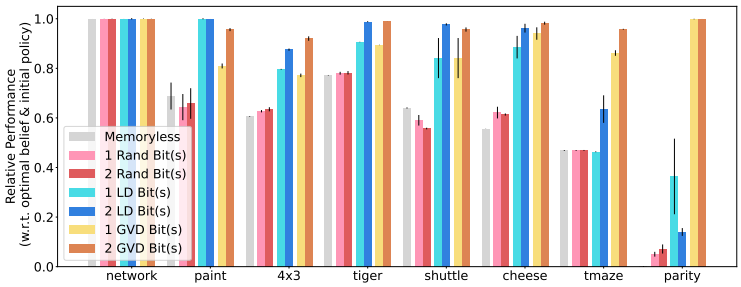


Figure 3: Performance of our method (GVD) versus baselines in closed-form tabular POMDPs.

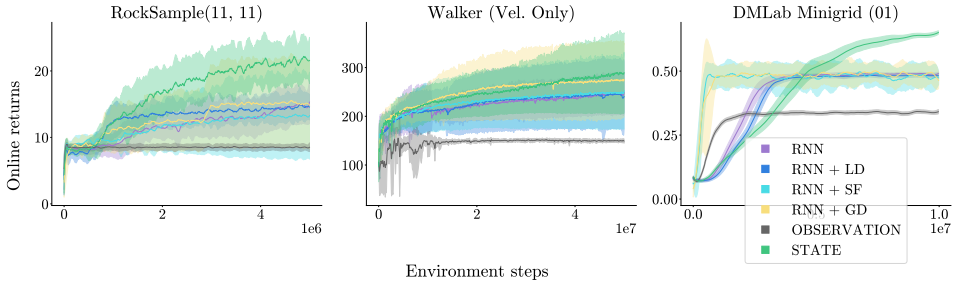


Figure 4: Performance of our method (RNN + GD) versus baselines in three deep RL environments.

successor representations are intractable, so we use the successor feature discrepancy (SF-GVD) instead. We experimented with different features, and found that random projections of observation differences (Achlioptas, 2003; Jaderberg et al., 2017) worked best (see Appendix C). We further use the version of GVD with fixed γ, as observation-dependent discounting slightly degraded performance. This is likely because these environments do not have any parity issues to overcome, so observation-dependent discounting is not required but makes the GVD signal noiser because the associated general value function is harder to learn.

We augment the recurrent PPO algorithm with two additional losses. The first is a successor feature learning loss for two TD(λ) successor feature heads, parameterized by $\theta_{F,0}$ and $\theta_{F,1}$:

$$L_{\text{SF}}(\theta) = \mathbb{E}^{\pi}\left[\sum_{i=1}^{n}\left(V^{\pi,\lambda=0}_{\theta_{F,0}}(z_t) - G^{\pi,\lambda=0}_{t,F}\right)^2_i + \left(V^{\pi,\lambda=1}_{\theta_{F,1}}(z_t) - G^{\pi,\lambda=1}_{t,F}\right)^2_i\right], \tag{8}$$

where $G^{\pi,\lambda}_{t,F}$ is the successor feature target for TD(λ) at timestep t, and z_t is the hidden state output of the RNN, $z_t = \mu_{\theta_{\text{RNN}}}(\omega_t, z_{t-1})$. θ represents all parameters, and the sum is taken over the number n of features. The second loss is the approximation of the SF-GVD based on the SF value heads:

$$L_{\text{GVD}}(\theta) = \mathbb{E}^{\pi}\left[\sum_{i=1}^{n}\left(V^{\pi,\lambda=0}_{\theta_{F,0}}(z_t) - V^{\pi,\lambda=1}_{\theta_{F,1}}(z_t)\right)^2_i\right]. \tag{9}$$

We started by testing the performance of (PPO augmented with) GVD in environments that *do* have a dense reward signal, where LD provides a strong baseline that GVD should be able to match. We use a subset of partially observable environments from the POBAX benchmark (Tao et al., 2025), specifically *RockSample*, *Velocity-Only Walker*, and *DMLab Minigrid*. Figure 4 shows the results for GVD-augmented PPO (*RNN + GVD*) in comparison with baselines standard recurrent PPO (*RNN*), LD-augmented PPO (*RNN + LD*), and PPO with successor features but no GVD (*RNN + SF*; see Equation 8), allowing us to isolate the effect of the GVD loss from that of the SF auxiliary task. We also include a memoryless policy as a floor and a state-conditioned policy as a ceiling. Full experimental details are in Appendix C. Across all environments, GVD matches or outperforms LD and SF, and even reaches ceiling-level performance in velocity-only Walker. This confirms empirically that GVD performs at least as well as LD in settings where the latter is applicable.

However, the core strength of GVD lies in the fact that, unlike LD, it can be used for memory learning in environments were rewards are very sparse or completely absent. For example, GVD could be integrated into an exploration phase in which an agent learns to navigate a partially observable environment without extrinsic reward; the memory learned with GVD during this phase could then be used to "warm-start" the learning phase of arbitrary downstream tasks. Experiments of this kind are currently in progress.

6 Related Work

Memory Learning in RL. Memory learning has been a long-studied problem in reinforcement learning. Whereas early approaches were based on classical methods for hidden state approximation such as finite state controllers (Meuleau et al., 1999), most modern techniques rely on variants of recurrent deep neural networks (RNN; Amari, 1972) trained with backpropagation through time (Mozer, 1995). One common and successful approach is to train an agent with a large RNN "end-to-end" to maximize reward (Bakker, 2001; Hausknecht & Stone, 2015; Ni et al., 2022; Lee et al., 2025). The λ-discrepancy introduced by Allen et al. (2024), which we generalize, offers an alternative approach that resolves partial observability by explicitly minimizing a measure of non-Markovianity as an auxiliary task.

Successor Features. A key ingredient in using GVD in a deep RL setting is successor features (SFs), which must be learned as an intermediate step. There is a large body of work on successor features in deep RL (Hoffman et al., 2024; Chua et al., 2024; Machado et al., 2020; Hansen et al., 2020). A known issue with learning SFs is representation collapse (Hoffman et al., 2024; Chua et al., 2024), remedies for which include adding a reconstruction penalty (Machado et al., 2020) or an entropy maximizing loss (Hansen et al., 2020). Another approach, which we use in this work, is to learn successor features directly over differences of subsequent observations (Jaderberg et al., 2017).

Self-Supervised Auxiliary Tasks. Auxiliary losses such as next-step prediction (Oh et al., 2015), inverse dynamics (Pathak et al., 2017), pixel control and reward prediction (Jaderberg et al., 2017), or contrastive predictive coding (van den Oord et al., 2018) enrich latent representations but do not directly test the Markov property. GVD is complementary: it evaluates whether the *current* representation is already Markovian and provides a targeted objective to fix deficiencies.

Summary. GVD unifies ideas from successor features, discrepancy-based memory learning, and auxiliary self-supervision. Unlike prior approaches that rely on rewards or handcrafted reconstruction signals, GVD offers a principled, reward-agnostic criterion that provably detects and mitigates partial observability while avoiding representation collapse.

7 Conclusion

We introduced the General Value Discrepancy (GVD), an extension of the λ-discrepancy designed to effectively detect and mitigate partial observability even when rewards are sparse or non-existent. We give a proof that GVD is capable of detecting partial observability under all circumstances, and demonstrate its effectiveness empirically both in a closed-form analytical setting as well as in large and complex deep RL environments.

References

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

Cameron Allen, Aaron Kirtland, Ruo Yu Tao, Sam Lobel, Daniel Scott, Nicholas Petrocelli, Omer Gottesman, Ronald Parr, Michael Littman, and George Konidaris. Mitigating partial observability in sequential decision processes via the lambda discrepancy. *Advances in Neural Information Processing Systems*, 37:62988–63028, 2024.

Shun-ichi Amari. Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers*, C-21(11):1197–1206, 1972.

Bram Bakker. Reinforcement learning with long short-term memory. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

André Barreto, Diana Borsa, John Quan, Tom Schaul, David Silver, Matteo Hessel, Daniel Mankowitz, Augustin Žídek, and Rémi Munos. Transfer in deep reinforcement learning using successor features and generalised policy improvement. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Douglas Blount and Michael A. Kouritzin. On convergence determining and separating classes of functions. *Stochastic Processes and their Applications*, 120(10):1898–1907, 2010. ISSN 0304-4149. DOI: https://doi.org/10.1016/j.spa.2010.05.018. URL https://www.sciencedirect.com/science/article/pii/S0304414910001523.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Raymond Chua, Arna Ghosh, Christos Kaplanis, Blake A. Richards, and Doina Precup. Learning successor features the simple way, 2024. URL https://arxiv.org/abs/2410.22133.

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

Stewart N. Ethier and Thomas G. Kurtz. *Markov Processes: Characterization and Convergence*. John Wiley & Sons, New York, 1986.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Steven Hansen, Will Dabney, André Barreto, David Warde-Farley, Tom Van de Wiele, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=BJeAHkrYDS.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable MDPs. In *Proceedings of the 2015 American Association for Artificial Intelligence*, 2015.

J. Hoffman, R. Ahmed, and J. Kiros. Successor features collapse in deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2410.22133.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *ICLR*. OpenReview.net, 2017. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2017.html#JaderbergMCSLSK17.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. Contextual decision processes with low bellman rank are pac-learnable. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1704–1713, Sydney, NSW, Australia, 2017. PMLR. URL https://proceedings.mlr.press/v70/jiang17a.html.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Steven G Krantz and Harold R Parks. *A Primer of Real Analytic Functions*. Birkhäuser, 1992.

Tejas D. Kulkarni, Ardavan Saeedi, Simanta Gautam, and Samuel J. Gershman. Deep successor reinforcement learning. In *NeurIPS Deep RL Workshop*, 2016.

Hojoon Lee, Takuma Seno, Jun Jet Tai, Kaushik Subramanian, Kenta Kawamoto, Peter Stone, and Peter R. Wurman. A champion-level vision-based reinforcement learning agent for competitive racing in gran turismo 7. *IEEE Robotics and Automation Letters*, 10(6):5545–52, June 2025.

Qinghua Liu, Alan Chung, Csaba Szepesvári, and Chi Jin. When is partially observable reinforcement learning not scary? In Jacob Abernethy and Alon Orlitsky (eds.), *Proceedings of the 35th Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 5175–5220, London, United Kingdom, July 2022. PMLR. URL https://proceedings.mlr.press/v178/liu22a.html.

Marlos C. Machado, Marc G. Bellemare, and Michael Bowling. Count-based exploration with the successor representation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Nicolas Meuleau, Leonid Peshkin, Kee-Eung Kim, and Leslie P. Kaelbling. Learning finite-state controllers for partially observable environments. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pp. 427–436, 1999.

Boris Samuilovich Mityagin. The zero set of a real analytic function. *Matematicheskie Zametki*, 107 (3):473–475, 2020.

Michael Mozer. A focused backpropagation algorithm for temporal pattern recognition. *Complex Systems*, 3, 1995.

Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free rl can be a strong baseline for many pomdps. *Proceedings of Machine Learning Research*, 162:16691–16723, 2022.

J. Oh, X. Guo, S. Singh, and H. Lee. Action-conditional video prediction using deep networks in atari games. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2863–2871, 2015.

D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, pp. 2778–2787, 2017.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, second edition, 2018. ISBN 9780262039246. URL http://incompleteideas.net/book/the-book-2nd.html.

Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 761–768, 2011.

Ruo Yu Tao, Kaicheng Guo, Cameron Allen, and George Konidaris. Benchmarking partial observability in reinforcement learning with a suite of memory-improvable domains. In *Reinforcement Learning Conference (RLC)*, 2025.

Sebastian Thrun. Monte carlo pomdps. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/299570476c6f0309545110c592b6a63b-Paper.pdf.

Heinrich Tietze. Über Funktionen, die auf einer abgeschlossenen Menge stetig sind. *Journal für die reine und angewandte Mathematik*, 145:9–14, 1915.

A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. In *arXiv preprint arXiv:1807.03748*, 2018.

Supplementary Materials

The following content was not necessarily subject to peer review.

Appendix A. Proof Details

This appendix includes formalizations and proofs of the material in Sections 2 and 3.

A.1 Setup

Fix a POMDP $(\mathcal{S}, \mathcal{A}, \Omega, T, \Phi, R, p_0, \gamma)$, and a policy $\pi\colon \Omega \to \Delta\mathcal{A}$. Recall that we assume $\mathcal{S}$, $\mathcal{A}$, and Ω to be finite, and that the length H of every episode is almost-surely bounded by some fixed $H_{\max} \in \mathbb{N}$. Then $W^\pi(s \mid \omega)$, the probability that the true state underlying an observation ω at a randomly sampled timestep of an episode is equal to s, is well-defined and given by

$$W^\pi(s \mid \omega) = \mathbb{P}^\pi\Big(s_t = s \,\Big|\, \omega_t = \omega, t \sim \text{Unif}(\{0, \dots, H\})\Big) = \frac{\mathbb{E}^\pi\left[\sum_{t=0}^H \mathbf{1}_{\{s_t = s, \omega_t = \omega\}}\right]}{\mathbb{E}^\pi\left[\sum_{t=0}^H \mathbf{1}_{\{s_t = s\}}\right]}.$$

Next, we give a precise definition of $P^\pi_{\text{MC}}(\cdot \mid \omega)$ and $P^\pi_{\text{TD}}(\cdot \mid \omega)$, which are probability distributions on the space

$$\Omega^\star = \left\{\omega^{(0:H)} = (\omega^{(0)}, \dots, \omega^{(H)})\colon H \le H_{\max}, \omega^{(i)} \in \Omega\right\}$$

of observation trajectories of length at most $H_{\max}$. Intuitively, $P^\pi_{\text{MC}}(\cdot)$ is the distribution over observation trajectories that we get in a dataset $\mathcal{D}$ when we repeatedly sample full episodes $(\omega_0, \omega_1, \dots, \omega_H)$ from the POMDP, and then add all of their partial truncations, $(\omega_t, \dots, \omega_H)$, for $t = 0, \dots, H$, to $\mathcal{D}$, as Monte-Carlo training targets for (general) value functions. Equivalently, $P^\pi_{\text{MC}}(\cdot \mid \omega)$ is the distribution over observation trajectories experienced by an agent that starts in a state sampled from $W^\pi(\cdot \mid \omega)$. Fig. 2a contains a good visualisation of this distribution: $s^{(0)} \sim W^\pi(\cdot \mid \omega^{(0)})$ and $a^{(0)} \sim \pi(\cdot \mid \omega^{(0)})$ are sampled conditional on $\omega^{(0)} = \omega$, and from thereon out, all transitions are sampled in the (hidden) state space, and observations are sampled given states using Φ.

In $P^\pi_{\text{TD}}(\cdot \mid \omega)$, the distribution of observation trajectories in the effective MDP, a single transition $\omega^{(0)} \to \omega^{(1)}$ is sampled just as in the true underlying POMDP—$s^{(0)} \sim W(\cdot \,|\, \omega^{(0)})$, $a^{(0)} \sim \pi(\cdot \,|\, \omega^{(0)})$, $s^{(1)} \sim T(\cdot \,|\, s^{(0)}, a^{(0)})$, and $\omega^{(1)} \sim \Phi(\cdot \,|\, s^{(1)})$—but then we *forget* about $s^{(1)}$ and repeat, that is we *resample* $\tilde{s}^{(1)} \sim W(\cdot \,|\, \omega^{(1)})$ and continue as previously. See Fig. 2a. We can think of $P^\pi_{\text{TD}}(\cdot \mid \omega)$ as the "Markovianization" of $P^\pi_{\text{MC}}(\cdot \mid \omega)$ that is being forced to forget its entire history except for the most recent observation after every transition. In that light, Lemma 3 makes intuitive sense: P^π_{MC} is equal to its Markovianization iff it is already Markov.

Separating classes Most of our theoretical results use the notion of *separating classes* of functions. We will use X to denote a generic *Polish space*—a topological space for which it is possible to choose a metric with respect to which it is complete and separable. This may seem awfully abstract and specific, but it is a natural assumption in the relevant measure theory literature, and suffice it to say that finite sets and $\mathbb{R}^k$ are Polish, and if Ω is Polish then so is $\Omega^\star$. For a probability measure P and a function f on the same domain, we write $P[f]$ as short-hand for the integral $\int f \,\mathrm{d}P$, if it exists.

> **Definition 2.** A *separating class* in a Polish space X is a set $\mathcal{F}$ of bounded measurable functions $X \to \mathbb{R}$ such that, for any two probability measures P and Q on X,
>
> $$P[f] = Q[f] \forall f \in \mathcal{F} \implies P = Q.$$
>
> Equivalently, if $\text{MMD}_{\mathcal{F}}(P, Q) = 0$ implies $P = Q$.

An excellent reference for this topic is chapter III.4 of Ethier & Kurtz (1986). In particular, the following two simple but useful examples follow directly from Theorem III.4.5 in that book.

Example 2. (i) The class of bounded and continuous functions is *always* separating.

(ii) If X is finite, then the class $\mathcal{F}_{\text{SR}} = \left\{\mathbf{1}_{\{x=\cdot\}}\right\}_{x\in X}$ of functions underlying successor representations is separating.

A.2 Proof of Theorems 4 and 5

We start by proving Theorems 4 and 5, which are relatively elementary to prove. To keep the appendix self-contained, we restate them here.

Theorem 4 (LD)**.** *The λ-discrepancy satisfies $\Lambda^\pi_\gamma = 0$ for every $\gamma \in (0,1)$ if and only if the marginal expected rewards at every timestep coincide in the effective MDP and the POMDP, that is*

$$\mathbb{E}^\pi_{\text{MC}}\left[r^{(t)} \,\middle|\, \omega^{(0)} = \omega\right] = \mathbb{E}^\pi_{\text{TD}}\left[r^{(t)} \,\middle|\, \omega^{(0)} = \omega\right], \quad \forall \omega \in \Omega, t \in \mathbb{N}_0. \tag{3}$$

Otherwise, $\Lambda^\pi_\gamma \neq 0$ for all $\gamma \in (0,1)$ except those in a set of measure zero.

Proof. First note that for any $\gamma \in (0,1)$, and any $\omega \in \Omega$,

$$V^\pi_{\text{MC},\gamma}(\omega) = \mathbb{E}^\pi_{\text{MC}}\left[\sum_{t=0}^{H_{\max}} \gamma^t r^{(t)} \,\middle|\, \omega\right] = \sum_{t=0}^{H_{\max}} \gamma^t \mathbb{E}^\pi_{\text{MC}}\left[r^{(t)} \,\middle|\, \omega^{(0)} = \omega\right], \tag{10}$$

and similarly for $V^\pi_{\text{TD},\gamma}$. This is a polynomial in γ, which is uniquely determined by its coefficients, hence (3) is equivalent to $V^\pi_{\text{MC},\gamma}(\omega) = V^\pi_{\text{TD},\gamma}(\omega)$ for all ω and γ, that is to $\Lambda^\pi_\gamma = 0$ for all γ.

It remains to show that if (3) does not hold, then $\Lambda^\pi_\gamma \neq 0$ for almost-all (that is, all outside of a Lebesgue-null set) $\gamma \in (0,1)$. Recall that by (10), $V^\pi_{\text{MC},\gamma}(\omega)$ for fixed ω as a function of γ is a polynomial and therefore real analytic (Krantz & Parks, 1992). Since Ω is finite,

$$(\Lambda^\pi_\gamma)^2 = \sum_{\omega\in\Omega} \left(V^\pi_{\text{MC},\gamma}(\omega) - V^\pi_{\text{TD},\gamma}(\omega)\right)^2$$

is a sum of finitely many real analytic functions and therefore also real analytic in γ. Finally, the fact that real analytic functions are either zero everywhere or only on a set of Lebesgue measure zero (Mityagin, 2020) implies that, if $\Lambda^\pi_\gamma \neq 0$ for *at least one* γ (equivalently if (3) doesn't hold), then $\Lambda^\pi_\gamma \neq 0$ for *almost-all* γ. □

Using the notion of a separating class, we can prove a slightly stronger version of Theorem 5.

Theorem 5* (GVD, I)**.** *The generalized value discrepancy satisfies $\Lambda^\pi_{f,\gamma} = 0$ for all $f\colon \Omega \to \mathbb{R}$ in a separating class $\mathcal{F}$ of Ω and all $\gamma \in (0,1)$ if and only if the observation marginals coincide in the effective MDP and the POMDP, that is*

$$P^\pi_{\text{MC}}\left(\omega^{(t)} = \cdot \,\middle|\, \omega\right) = P^\pi_{\text{TD}}\left(\omega^{(t)} = \cdot \,\middle|\, \omega\right), \quad \forall \omega \in \Omega, t \in \mathbb{N}_0. \tag{5}$$

Otherwise, if (5) doesn't hold, then

(i) There exists an $f \in \mathcal{F}$ such that $\Lambda^\pi_{f,\gamma} \neq 0$ for almost-all $\gamma \in (0,1)$.

(ii) $\Lambda^\pi_{f,\gamma} \neq 0$ for almost-all pairs (f,γ).

This implies Theorem 5 because, if Ω is finite, then the functions of the form $\mathbf{1}_{\{\omega=\cdot\}}$ form a separating class by Example 2.

Proof of Theorem 5.* For any given $f \in \mathcal{F}$, we can show exactly as in the proof of Theorem 4, by replacing $r^{(t)} = R(\omega^{(t)})$ with $f(\omega^{(t)})$, that $\Lambda^{\pi}_{\gamma,f} = 0$ for all $\gamma \in (0,1)$ if and only if

$$\mathbb{E}^{\pi}_{\text{MC}}\left[f(\omega^{(t)}) \,\middle|\, \omega\right] = \mathbb{E}^{\pi}_{\text{TD}}\left[f(\omega^{(t)}) \,\middle|\, \omega\right], \qquad \forall \omega \in \Omega, t \in \mathbb{N}_0. \tag{11}$$

Now it is precisely the definition of a separating class that, for fixed ω and t, (11) holds for all f in that class of functions if and only if (5) holds (for that ω and t).

Now suppose that (5) doesn't hold. We can proceed as in the proof of Theorem 4 to show that $\gamma \mapsto \Lambda^{\pi}_{\gamma,f}$ is real analytic for any fixed $f \in \mathcal{F}$, which again implies that either $\Lambda^{\pi}_{f,\gamma} = 0$ for all γ, or $\Lambda^{\pi}_{f,\gamma} \neq 0$ for almost all γ (Mityagin, 2020). There has to be at least one $f \in \mathcal{F}$ for which the latter is the case, otherwise $\Lambda^{\pi}_{f,\gamma} = 0$ for all $f \in \mathcal{F}$ and all $\gamma \in (0,1)$, contradicting the first part of the theorem.

For (ii), note that since Ω is finite, a function $f\colon \Omega \to \mathbb{R}$ can be identified with an $|\Omega|$-dimensional vector, and we can prove similarly to before that $(\Lambda^{\pi}_{f,\gamma})^2$ is real analytic jointly in γ and f (or more precisely the entries of the vector that defines f). The statement then follows with another application of (Mityagin, 2020). □

A.3 Proof of Theorem 6

Unlike Theorems 4 and 5, Theorem 6 has an involved proof that requires a number of technical lemmas. We begin by restating it.

Theorem 6 (GVD, II)**.** *The GVD satisfies $\Lambda^{\pi}_{f,\gamma} = 0$ for all $f\colon \Omega \to \mathbb{R}$ (equivalently, for all f of the form $\mathbf{1}_{\{\omega = \cdot\}}$) and all $\gamma\colon \Omega \to (0,1)$ if and only if $P^{\pi}_{\text{MC}} = P^{\pi}_{\text{TD}}$.*

To motivate how we might proceed, consider for a moment our task: we have given two probability measures $P = P^{\pi}_{\text{MC}}$ and $Q = P^{\pi}_{\text{TD}}$ on $\Omega^\star$, and wish to show that $P = Q$ must follow from equalities of the form

$$P\left[\gamma(\omega^{(0)}) \dots \gamma(\omega^{(t)}) f(\omega^{(t+1)})\right] = Q\left[\gamma(\omega^{(0)}) \dots \gamma(\omega^{(t)}) f(\omega^{(t+1)})\right]. \tag{12}$$

(At first we only have equality of sums of such terms, but it is relatively straight-forward to show that we also have (12).) The left-hand side is equal to

$$P\left[\gamma(\omega^{(0)}) \dots \gamma(\omega^{(t)}) P\left[f(\omega^{(t+1)} \,\middle|\, \omega^{(0:t)}\right]\right],$$

and similarly for the right-hand side. Since the product of γ's is invariant under a permutation of $\omega^{(0)}, \dots, \omega^{(t)}$, the very best we could hope to deduce from (12) is that

$$P\left(\omega^{(t+1)} = \cdot \,\middle|\, \{\{\omega^{(0)}, \dots, \omega^{(t)}\}\}\right) = Q\left(\omega^{(t+1)} = \cdot \,\middle|\, \{\{\omega^{(0)}, \dots, \omega^{(t)}\}\}\right) \tag{13}$$

where $\{\{\omega^{(0)}, \dots, \omega^{(t)}\}\}$ is the unordered (multi-)set of the first $t+1$ observations in the trajectory, that is we condition on the "bag" of observations seen thus far, without their ordering. Luckily, and somewhat surprisingly, it turns out that the "very best" case is indeed the one our universe has elected to satisfy, and the first step in showing this is to prove that functions of the form $(x_1, \dots, x_n) \mapsto \gamma(x_1) \dots \gamma(x_n)$ are separating "modulo permutations". Once (13) is established, we will blessed by fortune for a second time because it turns out, again to our surprise, that (13) for all t already implies $P = Q$ (in the specific case where $P = P^{\pi}_{\text{MC}}$ and $Q = P^{\pi}_{\text{TD}}$; the first surprising fact holds generally).

We start with some technical lemmas. For $n \in \mathbb{N}$, denote by S_n the set of bijections from $[n]$ to $[n]$, where $[n] = \{1, \dots, n\}$. Let X again be a generic Polish space. Then for a vector $x = (x_1, \dots, x_n) \in X^n$, and $\sigma \in S_n$, we write $x \circ \sigma$ for the vector $(x_{\sigma(1)}, \dots, x_{\sigma(n)})$. We define an equivalence relation on X^n by

$$x \sim y \iff \exists \sigma \in S_n\colon x = y \circ \sigma,$$

that is $x \sim y$ if the entries of x are a permutation of the entries of y. We can thus think of the quotient $X^n/\sim$ as the set of multi-sets $\{\{x_1, \dots, x_n\}\}$ of size n.

Lemma 8. *The quotient $X^n/\sim$ is a Polish space, and if d is a metric that generates the topology on X, and we denote by d again the associated L^p-distance on X^n (where $p \in [1, \infty]$ does not matter), then the topology on $X^n/\sim$ is generated by the metric*

$$d([x],[y]) \coloneqq \inf\{d(x', y') \colon x' \in [x], y' \in [y]\} = \inf\{d(x, y \circ \sigma) \colon \sigma \in S_n\}.$$

Proof. We first show that the given function is a metric (with this choice of metric that is not true for a general quotient metric space!). We have $d([x],[x]) = 0$, and $d([x],[y]) \geq 0$. Suppose $d([x],[y]) = 0$, then it means that x is a permutation of y so $[x] = [y]$. For the triangle inequality, let $x, y, z \in X^n$, and let $\sigma, \tau \in S_n$ such that

$$d([x],[y]) = d(x, y \circ \sigma), \quad d([y],[z]) = d(y, z \circ \tau) = d(y \circ \sigma, z \circ \tau \circ \sigma),$$

so

$$d([x],[z]) \leq d(x, z \circ \tau \circ \sigma) \leq d(x, y \circ \sigma) + d(y \circ \sigma, z \circ \sigma \circ \tau) = d([x],[y]) + d([y],[z]).$$

Separability is inherited from X^n. For completeness, suppose $([x_n])_{n\in\mathbb{N}}$ is a Cauchy sequence. We can extract a subsequence $([x_{k(n)}])_{n\in\mathbb{N}}$ such that $d([x_{k(n)}],[x_{k(n+1)}]) \leq 2^{-n}$. Then we can choose an arbitrary representative for $x_{k(1)}$, and then iteratively, given the representative $x_{k(n)}$ that we've chosen, we can choose the representative $x_{k(n+1)}$ such that $d(x_{k(n)}, x_{k(n+1)})$ is the same as the distance of their equivalence classes, which is no larger than 2^{-n}. This means that the sequence is Cauchy in X^n and hence converges, so this subsequence also converges on the level of equivalence classes, which implies that, in fact, the entire sequence converges to the same limit in $X^n/\sim$.

To show that this metric generates the quotient topology, we show that closed sets are the same. A set $S \subset X^n/\sim$ is closed in the quotient topology iff the set of $S' = \{x \colon [x] \in S\}$ is closed in X^n (in the topology and w.r.t. the metric, which are the same here). If that's the case, and $[x_n] \to [x]$ for $[x_n] \in S$, then we can choose any representative $x \in [x]$, and a sequence of representatives such that $x_n \to x$ in X^n, so $x \in S'$, so $[x] \in S$. Conversely, if S is closed in the metric topology in $X^n/\sim$, and if $x_n \in S'$ and $x_n \to x$, then $[x_n] \to [x]$, so $[x] \in S$, so $x \in S'$. □

Lemma 9. *Let X be a Polish space, and $n \in \mathbb{N}$. Then the set of functions of the form*

$$[x] \mapsto \gamma(x_1) \cdot \ldots \cdot \gamma(x_n)$$

for $\gamma \colon X \to \mathbb{R}$ continuous and bounded, is a separating class of $X^n/\sim$.

Proof. It is a well-known fact in probability theory that a sufficient condition for a set $\mathcal{F}$ of bounded and continuous functions $Y \to \mathbb{R}$ on some Polish space Y to be separating, is for $\mathcal{F}$ to be closed under pairwise multiplication (i.e. $f, g \in \mathcal{F}$ implies $fg \in \mathcal{F}$), and for $\mathcal{F}$ to separate points (i.e. if $x \neq y$ then there is some $f \in \mathcal{F}$ with $f(x) \neq f(y)$). See for example Theorem 3.4.5(a) of Ethier & Kurtz (1986), or page 2 of Blount & Kouritzin (2010) for a more recent discussion.

Firstly, the set of functions under consideration is closed under pairwise multiplication, since

$$\Big(\gamma(x_1)\dots\gamma(x_n)\Big) \cdot \Big(\nu(x_1)\dots\nu(x_n)\Big) = \gamma(x_1)\nu(x_1) \cdot \ldots \cdot \gamma(x_n)\nu(x_n),$$

so it remains to show it also separates points. Indeed, supoose that $[x] \neq [y]$, i.e. x and y are different as multi-sets. If they are also different as sets, i.e. there is some $z \in X$ that appears as an entry in say x but not in y, then we can find a $\gamma \colon X \to \mathbb{R}$ such that $\gamma(z) = 0$ but $\gamma(y_1) = \ldots = \gamma(y_n) = 1$ (where $y = (y_1, \dots, y_n)$); indeed we can prescribe values of γ on any finite number of isolated points and extend it to a continuous bounded function on all of X by Tietze's extension theorem (Tietze, 1915). Then $\gamma(x_1)\dots\gamma(x_n) = 0$ but $\gamma(y_1)\dots\gamma(y_n) = 1$. If x, y are the same as sets but different as multi-sets, then we can choose $\gamma \colon X \to \mathbb{R}$ in such a way that it attains a different prime number on each of the elements of the two (identical) sets, and, since prime factorizations are unique, we can infer the multiplicity of each element from the product, and therefore distinguish x and y. □

Now we can prove surprising fact number one. In accordance with (6), write

$$F_{f,\gamma}\colon X^\star \to \mathbb{R}; \quad (x_{1:n}) \mapsto \sum_{k=1}^{n} \gamma(x_1)\dots\gamma(x_{k-1})f(x_k)$$

for bounded and continuous functions $f, \gamma\colon X \to \mathbb{R}$.

Lemma 10. *If P, Q are probability measures on $X^\star$, then $P[F_{f,\gamma}] = Q[F_{f,\gamma}]$ for all bounded and continuous $f\colon X \to \mathbb{R}$ and $\gamma\colon X \to (0,1)$ if and only if $P(x_1 = \cdot) = Q(x_1 = \cdot)$ and*

$$P\left(x_{n+1} \mid \{\{x_1, \dots, x_n\}\}\right) = Q\left(x_{n+1} \mid \{\{x_1, \dots, x_n\}\}\right) \tag{14}$$

for all $n \in \mathbb{N}_0$ and $x_1, \dots, x_n \in X$.

Proof. For a function $\gamma\colon X \to \mathbb{R}$, denote by $\gamma^\times\colon X^\star/\sim \to \mathbb{R}$ the function $\{\{x_1, \dots, x_n\}\} \mapsto \gamma(x_1)\dots\gamma(x_n)$.

Suppose first that (14) holds. Then, by induction,

$$P(\{\{x_1, \dots, x_n\}\}) = Q(\{\{x_1, \dots, x_n\}\}) \tag{15}$$

for all $n \in \mathbb{N}$ and $x_1, \dots, x_n \in X$. Now fix f and γ, then

$$\begin{aligned} P\left[\gamma(x_1)\dots\gamma(x_n)f(x_{n+1})\right] &= P\left[\gamma^\times(\{\{x_1, \dots, x_n\}\})f(x_{n+1})\right] \\ &= \int \gamma^\times(\{\{x_1, \dots, x_n\}\}) \left(\int f(x_{n+1}) P(\mathrm{d}x_{n+1} \mid \{\{x_1, \dots, x_n\}\}) \right) P(\mathrm{d}\{\{x_1, \dots, x_n\}\}), \end{aligned} \tag{16}$$

which equals the same expression with Q substituted for P by (14) and (15), and hence we proved $P[F_{f,\gamma}] = Q[F_{f,\gamma}]$.

Now suppose that $P[F_{f,\gamma}] = Q[F_{f,\gamma}]$ for all f, γ. Firstly by choosing $\gamma \equiv 0$ we get $P[f(x_1)] = Q[f(x_2)]$ for all f, which implies $P(x_1 = \cdot) = Q(x_1 = \cdot)$. Now for any fixed f, γ, we have $P[F_{f,\varepsilon\gamma}] = Q[F_{f,\varepsilon\gamma}]$ for any $\varepsilon > 0$, which reads

$$\sum_{k=0}^{n} \varepsilon^k P\left[\gamma(x_1)\dots\gamma(x_k)f(x_{k+1})\right] = \sum_{k=0}^{n} \varepsilon^k Q\left[\gamma(x_1)\dots\gamma(x_k)f(x_{k+1})\right].$$

Taking the k'th derivative with respect to ε and letting $\varepsilon \to 0$ gives

$$P\left[\gamma(x_1)\dots\gamma(x_k)f(x_{k+1})\right] = Q\left[\gamma(x_1)\dots\gamma(x_k)f(x_{k+1})\right],$$

so this holds for all $k \in \mathbb{N}$, f, and γ. We already proved $P(x_1) = Q(x_1)$, which is just (14) for $n = 0$, so we may assume by means of induction that we have proved (14) up until some fixed $k \in \mathbb{N}_0$. In particular, as argued before, we have (15) up until the same value of k. We can use (16) again to obtain

$$\begin{aligned} &\int \gamma^\times(\{\{x_1, \dots, x_k\}\}) \left(\int f(x_{k+1}) P(\mathrm{d}x_{k+1} \mid \{\{x_1, \dots, x_k\}\}) \right) P(\mathrm{d}\{\{x_1, \dots, x_k\}\}) \\ &\quad = P\left[\gamma(x_1)\dots\gamma(x_k)f(x_{k+1})\right] = Q\left[\gamma(x_1)\dots\gamma(x_k)f(x_{k+1})\right] \\ &\quad = \int \gamma^\times(\{\{x_1, \dots, x_k\}\}) \left(\int f(x_{k+1}) Q(\mathrm{d}x_{k+1} \mid \{\{x_1, \dots, x_k\}\}) \right) Q(\mathrm{d}\{\{x_1, \dots, x_k\}\}) \\ &\quad = \int \gamma^\times(\{\{x_1, \dots, x_k\}\}) \left(\int f(x_{k+1}) Q(\mathrm{d}x_{k+1} \mid \{\{x_1, \dots, x_k\}\}) \right) P(\mathrm{d}\{\{x_1, \dots, x_k\}\}), \end{aligned}$$

where we used (15) in the final step. Varying γ for fixed f then implies by Lemma 9 that

$$\int f(x_{k+1}) P(\mathrm{d}x_{k+1} \mid \{\{x_1, \dots, x_k\}\}) = \int f(x_{k+1}) Q(\mathrm{d}x_{k+1} \mid \{\{x_1, \dots, x_k\}\})$$

for P-almost all $\{\{x_1, \dots, x_k\}\}$, for every f. Since there exists a countable separating system for the f's (X is Polish) we can choose a common set of exceptions for all the f's, which yields the claim for $k + 1$. □

Now for the second surprising fact, which is that, in the particular case that we are interested in, $X = \Omega$, $P = P^\pi_{\text{MC}}$, and $Q = P^\pi_{\text{TD}}$, (14) already implies $P = Q$.

Lemma 11. *If* (14) *holds for* $X = \Omega$, $P = P^\pi_{\text{MC}}$, *and* $Q = P^\pi_{\text{TD}}$, *then* $P^\pi_{\text{MC}} = P^\pi_{\text{TD}}$.

Proof. We will show by contradiction that for both P^π_{MC} and P^π_{TD}, any multi-set $\{\{\omega^{(0)}, \ldots, \omega^{(H)}\}\}$ has either probability zero or a *unique* representative (i.e. ordering) with positive probability. Given that, there is no aliasing in (14) and we immediately get

$$P^\pi_{\text{MC}}(\omega^{(t+1)} \mid \omega^{(0:t)}) = P^\pi_{\text{TD}}(\omega^{(t+1)} \mid \omega^{(0:t)})$$

for all t, that is $P^\pi_{\text{MC}} = P^\pi_{\text{TD}}$.

Now assume for contradiction that at least one of the two has aliasing in the sense that there exists some multiset $\{\{\omega^{(0)}, \ldots, \omega^{(k)}\}\}$, two different orderings of which are possible in, say, P^π_{MC}. Now define a directed graph on the vertex set $\{0, \ldots, k\}$ in which an edge from i to j is present if and only if $P^\pi_{\text{MC}}(\omega^{(j)} \mid \omega^{(i)}) > 0$ (since one-step transition probabilities are the same for P^π_{MC} and P^π_{TD}, the graph would look the same if we had used P^π_{TD}). Then our assumption implies that this graph has a cycle. Indeed, suppose without loss of generality that the first possible ordering is $(\omega^{(0)}, \ldots, \omega^{(k)})$, implying edges $0 \to 1 \to \ldots \to k$, and let the second possible ordering be $(\omega^{(\sigma(0))}, \ldots, \omega^{(\sigma(k))})$, implying edges $\sigma(0) \to \sigma(1) \to \ldots \to \sigma(k)$. Now if we assume that this does not induce a cycle, then that must mean that $\sigma(i+1) > \sigma(i)$ for all i, but that could only happen if $\sigma(i) = i$ for all i which is not true. But since P^π_{TD} is Markov, any directed path in this graph is a trajectory with positive probability in P^π_{TD}, and the existence of a cycle implies that arbitrarily long trajectories are possible in P^π_{TD}. But by assumption, trajectories in the real POMDP, and therefore in P^π_{MC} are bounded by $H_{\max}$, so if $(\omega^{(0:t)})$ is some trajectory that is possible in P^π_{TD} with $t > H_{\max}$, then, using (14),

$$0 < P^\pi_{\text{TD}}(\{\{\omega^{(0)}, \ldots, \omega^{(t)}\}\}) = P^\pi_{\text{MC}}(\{\{\omega^{(0)}, \ldots, \omega^{(t)}\}\}) = 0,$$

a contradiction. □

Putting things together, we obtain a proof of Theorem 6.

Proof of Theorem 6. If $P^\pi_{\text{MC}} = P^\pi_{\text{TD}}$ then $\Lambda^\pi_{f,\gamma} = 0$ for all f, γ. If, conversely, the latter holds, then Lemma 10 implies (14), which by Lemma 11 implies $P^\pi_{\text{MC}} = P^\pi_{\text{TD}}$. □

Appendix B. Closed-Form Gradient Optimization Details

Closed-form experiments were conducted with the JAX (Bradbury et al., 2018) library, largely following the same closed-form iterative optimization set up as Allen et al. (2024) with the following hyperparameters:

Hyperparameter	
Step size	0.01
n_{steps}	50K
Optimizer	Adam
λ_0	0
λ_1 (LD & GVD)	1
γ_{min} (GVD)	0.8
γ_{max} (GVD)	0.99

Table 1: Hyperparameters used for all closed-form experiments

where LD and GVD stand for optimization hyperparameters for the λ-discrepancy and general value discrepancy respectively. The optimization was done between n_{steps} of policy optimization, memory

improvement, and a final round of policy optimization. For T-Maze and Parity Check environments, our agents are trained on 100K steps, with an additional hyperparameter sweep for both LD and GVD algorithms. We sweep the following hyperparameters for both algorithms:

Hyperparameter	Values
λ_0	$[0.0, 0.2, 0.4]$
λ_1	$[0.6, 0.8, 1.0]$

(a) Hyperparameters swept for closed-form λ-discrepancy minimization

Hyperparameter	Values
n_γ	$[1, 3, 5]$
γ_{min}	$[0.3, 0.6, 0.9]$

(b) Hyperparameters swept for closed-form general value discrepancy minimization.

Table 2: Hyperparameters swept for both closed-firm λ-discrepancy and general value discrepancy minimization.

In the GVD settings for T-Maze and parity, we considered multiple γ parameters, as well as multiple minimum γ parameters for varying γ. Overall, we swept the same number of hyperparameters between our λ-discrepancy baseline, and our GVD algorithm. While our closed-form optimization uses the same optimization procedure as in Allen et al. (2024), the `memory_improvement` procedure was altered for minimizing the general value discrepancy $\Lambda^{\pi}_{f,\gamma}$.

Appendix C. Deep Reinforcement Learning Experimental Details

We elucidate experimental details in our deep reinforcement learning experiments. All base algorithms are variations of the recurrent PPO algorithm with added auxiliary losses. We describe both the successor feature auxiliary loss and general value discrepancy auxiliary loss below.

C.1 Successor Feature Learning

Learning successor features in deep reinforcement learning has shown to be unstable, since the trivial representation (mapping everything to a single point) reduces successor feature losses to 0. Previous work has shown that the differences between two subsequent observations act as a stable learning signal for successor features (Jaderberg et al., 2017). To extend for large observation spaces, we consider successor features over fixed random projections (Achlioptas, 2003) of our observations. Random projections are apt for auxiliary task learning because of a few useful properties. They almost always preserve distances, which means that discrepancies over a function of randomly projected observations also almost always preserve distances, and are usually much smaller than raw observations. We use the following fixed linear random projection:

$$F_{i,j} = \sqrt{3/k} \times \begin{cases} 1 & \text{w.p. } \frac{1}{6} \\ 0 & \text{w.p. } \frac{2}{3} \\ -1 & \text{w.p. } \frac{1}{6} \end{cases} \tag{17}$$

where F is a $k \times n_{proj}$ randomly initialized matrix. k is the size of the original feature vector, and n_{proj} is the size of the randomly projected feature vector.

We ran experiments on different basis features for successor feature learning, and found that the *difference* between features across two subsequent time steps (Jaderberg et al., 2017) were best for voth successor feature learning and general value discrepancy learning.

C.2 Deep General Value Discrepancy

In order to learn the auxiliary task of general value discrepancy minimization, we are required to learn two successor feature heads, each corresponding to a λ. This means that our overall architecture has a policy head, a value prediction head, and two successor feature prediction heads. We sweep the following hyperparameters for the algorithms we compare:

Hyperparameter	
Step size	$[2.5 \times 10^{-3}, 2.5 \times 10^{-4}, 2.5 \times 10^{-5}, 2.5 \times 10^{-6}]$
λ_1	$[0.1, 0.5, 0.7, 0.9, 0.95]$
λ_2 (LD & GVD)	$[0.1, 0.5, 0.7, 0.9, 0.95]$
β (LD & GVD)	$[0, 0.25, 0.5]$

Table 3: Hyperparameters swept across all algorithms. Rows labelled with λ-discrepancy are hyperparameters swept specific to our algorithm.

We weight the L_{GVD} loss by the β parameter, and also have a separate weight for the L_{SF} loss, which we keep fixed at 0.25 for all runs involving successor features for general value discrepancy minimization.