Reinforcement Learning with a Terminator

Guy Tennenholtz [*†]
guytenn@gmail.com

Nadav Merlis [*] **Lior Shani** [*] **Shie Mannor** [*†]

Uri Shalit [*] **Gal Chechik** [†‡] **Assaf Hallak** [†] **Gal Dalal** [†]

Abstract

We present the problem of reinforcement learning with exogenous termination. We define the Termination Markov Decision Process (TerMDP), an extension of the MDP framework, in which episodes may be interrupted by an external non-Markovian observer. This formulation accounts for numerous real-world situations, such as a human interrupting an autonomous driving agent for reasons of discomfort. We learn the parameters of the TerMDP and leverage the structure of the estimation problem to provide state-wise confidence bounds. We use these to construct a provably-efficient algorithm, which accounts for termination, and bound its regret. Motivated by our theoretical analysis, we design and implement a scalable approach, which combines optimism (w.r.t. termination) and a dynamic discount factor, incorporating the termination probability. We deploy our method on high-dimensional driving and MinAtar benchmarks. Additionally, we test our approach on human data in a driving setting. Our results demonstrate fast convergence and significant improvement over various baseline approaches.

1 Introduction

The field of reinforcement learning (RL) involves an agent interacting with an environment, maximizing a cumulative reward [Puterman, 2014]. As RL becomes more instrumental in real-world applications [Lazic et al., 2018, Kiran et al., 2021, Mandhane et al., 2022], exogenous inputs beyond the prespecified reward pose a new challenge. Particularly, an external authority (e.g., a human operator) may decide to terminate the agent's operation when it detects undesirable behavior. In this work, we generalize the basic RL framework to accommodate such external feedback.

We propose a generalization of the standard Markov Decision Process (MDP), in which external termination can occur due to a non-Markovian observer. When terminated, the agent stops interacting with the environment and cannot collect additional rewards. This setup describes various real-world scenarios, including: passengers in autonomous vehicles [Le Vine et al., 2015, Zhu et al., 2020], users in recommender systems [Wang et al., 2009], employees terminating their contracts (churn management) [Sisodia et al., 2017], and operators in factories; particularly, datacenter cooling systems, or other safety-critical systems, which require constant monitoring and rare, though critical, human takeovers [Modares et al., 2015]. In these tasks, human preferences, incentives, and constraints play a central role, and designing a reward function to capture them may be highly complex. Instead, we propose to let the agent itself learn these latent human utilities by leveraging the termination events.

We introduce the Termination Markov Decision Process (TerMDP), depicted in Figure 1. We consider a terminator, observing the agent, which aggregates penalties w.r.t. a predetermined, state-action-dependent, yet *unknown*, cost function. As the agent progresses, unfavorable states accumulate costs

[*]Technion, Israel institute of technology
[†]Nvidia Research, Israel
[‡]Bar Ilan University, Israel

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

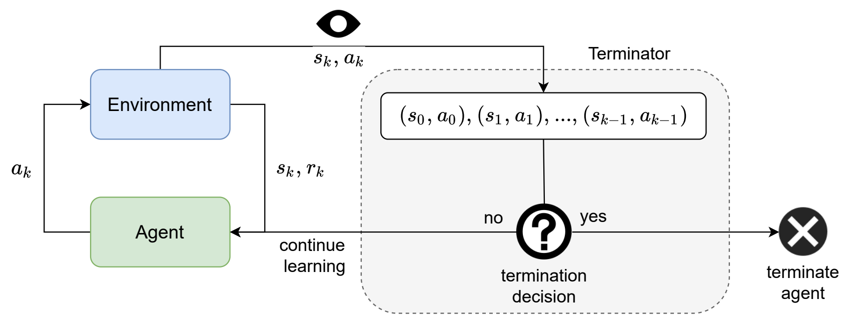

Figure 1: A block diagram of the TerMDP framework. An agent interacts with an environment while an exogenous observer (i.e., terminator) can choose to terminate the agent based on previous interactions. If the agent is terminated, it transitions to a sink state where a reward of 0 is given until the end of the episode.

that gradually increase the terminator's inclination to stop the agent and end the current episode. Receiving merely the sparse termination signals, the agent must learn to behave in the environment, adhering to the terminator's preferences while maximizing reward.

Our contributions are as follows. **(1)** We introduce a novel history-dependent termination model, a natural extension of the MDP framework which incorporates non-trivial termination (Section 2). **(2)** We learn the unknown costs from the implicit termination feedback (Section 3), and provide local guarantees w.r.t. every visited state. We leverage our results to construct a tractable algorithm and provide regret guarantees. **(3)** Building upon our theoretical results, we devise a practical approach that combines optimism with a cost-dependent discount factor, which we test on MinAtar [Young and Tian, 2019] and a new driving benchmark. **(4)** We demonstrate the efficiency of our method on these benchmarks as well as on human-collected termination data (Section 5). Our results show significant improvement over other candidate solutions, which involve direct termination penalties and history-dependent approaches. We also introduce a new task for RL – a driving simulation game which can be easily deployed on mobile phones, consoles, and PC [4].

2 Termination Markov Decision Process

We begin by presenting the termination framework and the notation used throughout the paper. Informally, we model the termination problem using a logistic model of past "bad behaviors". We use an unobserved state-action-dependent cost function to capture these external preferences. As the overall cost increases throughout time, so does the probability of termination.

For a positive integer n, we denote $[n] = \{1, \dots, n\}$. We define the Termination Markov Decision Process (TerMDP) by the tuple $\mathcal{M}_T = (\mathcal{S}, \mathcal{A}, P, R, H, c)$, where $\mathcal{S}$ and $\mathcal{A}$ are state and action spaces with cardinality S and A, respectively, and $H \in \mathbb{N}$ is the maximal horizon. We consider the following protocol, which proceeds in discrete episodes $k = 1, 2, \dots, K$. At the beginning of each episode k, an agent is initialized at state $s_1^k \in \mathcal{S}$. At every time step h of episode k, the agent is at state $s_h^k \in \mathcal{S}$, takes an action $a_h^k \in \mathcal{A}$ and receives a random reward $R_h^k \in [0, 1]$ generated from a fixed distribution with mean $r_h(s_h^k, a_h^k)$. A terminator overseeing the agent utilizes a cost function $c : [H] \times \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ that is unobserved and *unknown to the agent*. At time step h, the episode terminates with probability

$$\rho_h^k(c) = \rho\left(\sum_{t=1}^{h} c_t(s_t^k, a_t^k) - b\right),$$

where $\rho(x) = (1 + \exp(-x))^{-1}$ is the logistic function and $b \in \mathbb{R}$ is a bias term which determines the termination probability when no costs are aggregated. Upon termination, the agent transitions to a terminal state s_{term} which yields no reward, i.e., $r_h(s_{\text{term}}, a) = 0$ for all $h \in [H], a \in \mathcal{A}$. If no termination occurs, the agent transitions to a next state s_{h+1}^k with probability $P_h(s_{h+1}^k | s_h^k, a_h^k)$. Let $t_k^* = \min\{h : s_h^k = s_{\text{term}}\} - 1$ be the time step when the k^{th} episode was terminated. Notice that the termination probability is non-Markovian, as it depends on the entire trajectory history. We also

[4] Code for Backseat Driver and our method, TermPG, can be found at https://github.com/guytenn/Terminator.

note that, when $c \equiv 0$, the TerMDP reduces to a finite horizon MDP with discount factor $\gamma = \rho(-b)$. Finally, we note that our model allows for negative costs. Indeed, these may capture satisfactory behavior, diminishing the effect of previous mistakes, and decreasing the probability of termination.

We define a stochastic, history dependent policy $\pi_h(s_h, \tau_{1:h})$ which maps trajectories $\tau_{1:h} = (s_1, a_1, \dots, s_{h-1}, a_{h-1})$ up to time step h (excluding) and the h^{th} states s_h to probability distributions over $\mathcal{A}$. Its value is defined by $V_h^\pi(s,\tau) = \mathbb{E}\left[\sum_{t=h}^H r_t(s_t, a_t) \;\middle|\; s_h = s, \tau_{1:h} = \tau, a_t \sim \pi_t(s_t, \tau_{1:t})\right]$. With slight abuse of notation, we denote the value at the initial time step by $V_1^\pi(s)$. An optimal policy π^* maximizes the value for all states and histories simultaneously [5]; we denote its value function by V^*. We measure the performance of an agent by its *regret*; namely, the difference between the cumulative value it achieves and the value of an optimal policy,

$$\text{Reg}(K) = \sum_{k=1}^K V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k).$$

Notations. We denote the Euclidean norm by $\|\cdot\|_2$ and the Mahalanobis norm induced by the positive definite matrix $A \succ 0$ by $\|x\|_A = \sqrt{x^T A x}$. We denote by $n_h^k(s,a)$ the number of times that a state action pair (s,a) was visited at the h^{th} time step before the k^{th} episode. Similarly, we denote by $\hat{X}_h^k(s,a)$ the empirical average of a random variable X (e.g., reward and transition kernel) at (s,a) in the h^{th} time step, based on all samples before the k^{th} episode.

We assume there exists a known constant L that bounds the norm of the costs; namely, $\sqrt{\sum_{s,a} \sum_{t=1}^H c_t^2(s_t, a_t)} \leq L$, and denote the set of possible costs by $\mathcal{C}$. We also denote the maximal reciprocal derivative of the logistic function by $\kappa = \max_{h \in [H]} \max_{\{(s_t,a_t)\}_{t=1}^h \in (\mathcal{S}\times\mathcal{A})^h} \left(\dot{\rho}\left(\sum_{t=1}^h c_t(s_t,a_t) - b\right)\right)^{-1}$. This factor will be evident in our theoretical analysis in the next section, as estimating the costs in regions of saturation of the sigmoid is more difficult when the derivative nears zero. Finally, we use $\mathcal{O}(x)$ to refer to a quantity that depends on x up to a poly-log expression in S, A, K, H, L, κ and $\log\left(\frac{1}{\delta}\right)$.

3 An Optimistic Approach to Overcoming Termination

Unlike the standard MDP setup, in the TerMDP model, the agent can potentially be terminated at any time step. Consider the TerMDP model for which the costs are *known*. We can define a Markov policy π_h mapping augmented states $\mathcal{S} \times \mathbb{R}$ to a probability distribution over actions, where here, the state space is augmented by the accumulated costs $\sum_{t=1}^{h-1} c_t(s_t, a_t)$. There exists a policy, which does not use historical information, besides the accumulated costs, and achieves the value of the optimal history-dependent policy (see Appendix B). Therefore, when solving for an optimal policy (e.g., by planning), one can use the current accumulated cost instead of the full trajectory history.

This suggests a plausible approach for solving the TerMDP – first learn the cost function, and then solve the state-augmented MDP for which the costs are known. This, in turn, leads to the following question: **can we learn the costs c from the termination signals?** In what follows, we answer this question affirmatively. We show that by using the termination structure, one can efficiently converge to the true cost function *locally* – for every state and action. We provide uncertainty estimates for the state-wise costs, which allow us to construct an efficient optimistic algorithm for solving the problem.

Learning the Costs. To learn the costs, we show that the agent can effectively gain information about costs even in time steps where no termination occurs. Recall that at any time step $h \in [H-1]$, the agent acquires a sample from a Bernoulli random variable with parameter $p = \rho_h^k(c) = \rho\left(\sum_{t=1}^h c_t(s_t^k, a_t^k) - b\right)$. Notably, a lack of termination, which occurs with probability $1 - \rho_h^k(c)$, is also an informative signal of the unknown costs. We propose to leverage this information by recognizing the costs c as parameters of a probabilistic model, maximizing their likelihood. We use

[5] Such a policy always exists; we can always augment the state space with the history, which would make the environment Markovian and imply the existence of an optimal history-dependent policy [Puterman, 2014].

Algorithm 1 TermCRL: Termination Confidence Reinforcement Learning

require: $\lambda > 0$
for $k = 1, \ldots, K$ **do**
  for $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$ **do**
    $\bar{r}_h^k(s, a) = \hat{r}_h^k(s, a) + b_k^r(h, s, a) + b_k^p(h, s, a)$
    $\bar{c}_h^k(s, a) = \hat{c}_h^k(s, a) - b_k^c(h, s, a)$ // Appendix H.1
  end for
  $\pi^k \leftarrow$ TerMDP-Plan$\Big(\mathcal{M}_T\big(\mathcal{S}, \mathcal{A}, H, \bar{r}^k, \hat{P}^k, \bar{c}^k\big)\Big)$ // Appendix G
  Rollout a trajectory by acting π^k
  $\hat{c}^{k+1} \in \arg\max_{c \in \mathcal{C}} \mathcal{L}_\lambda^k(c)$ // Equation (1)
  Update $\hat{P}^{k+1}(s, a), \hat{r}^{k+1}(s, a), n^{k+1}(s, a)$ over rollout trajectory
end for

the regularized cross-entropy, defined for some $\lambda > 0$ by

$$\mathcal{L}_\lambda^k(c) = \sum_{k'=1}^{k} \sum_{h=1}^{H-1} \left[\mathbb{1}\{h < t_{k'}^*\} \log\big(1 - \rho_h^k(c)\big) + \mathbb{1}\{h = t_{k'}^*\} \log\big(\rho_h^k(c)\big)\right] - \lambda \left\|c\right\|_2^2. \quad (1)$$

By maximizing the cost likelihood in Equation (1), global guarantees of the cost can be achieved, similar to previous work on logistic bandits [Zhang et al., 2016, Abeille et al., 2021]. Particularly, denoting by $\hat{c}^k \in \arg\max \mathcal{L}_\lambda^k(c)$ the maximum likelihood estimates of the costs, it can be shown that for any history, a global upper bound on $\left\|\hat{c}^k - c\right\|_{\Sigma_k}$ can be obtained, where the history-dependent design matrix Σ_k captures the empirical correlations of visitation frequencies (see Appendix J for details). Unfortunately, using $\left\|\hat{c}^k - c\right\|_{\Sigma_k}$ amounts to an intractable algorithm [Chatterji et al., 2021], and thus to an undesirable result.

Instead, as terminations are sampled on *every time step* (i.e., non-terminations are informative signals as well), we show we can obtain a *local* bound on the cost function c. Specifically, we show that the error $\left|\hat{c}_h^k(s, a) - c_h(s, a)\right|$ diminishes with $n_h^k(s, a)$. The following result is a main contribution of our work, and the crux of our regret guarantees later on (see Appendix J for proof).

Theorem 1 (Local Cost Estimation Confidence Bound)**.** *Let $\hat{c}^k \in \arg\max_{c \in \mathcal{C}} \mathcal{L}_\lambda^k(c)$ be the maximum likelihood estimate of the costs. Then, for any $\delta > 0$, with probability of at least $1 - \delta$, for all episodes $k \in [K]$, timesteps $h \in [H-1]$ and state-actions $(s, a) \in \mathcal{S} \times \mathcal{A}$, it holds that*

$$\left|\hat{c}_h^k(s, a) - c_h(s, a)\right| \leq \mathcal{O}\left(\left(n_h^k(s, a)\right)^{-0.5} \sqrt{\kappa S A H L^3} \log\left(\frac{1}{\delta}\left(1 + \frac{kL}{S^2 A^2 H}\right)\right)\right).$$

We note the presence of κ in our upper bound, a common factor [Chatterji et al., 2021], which is fundamental to our analysis, capturing the complexity of estimating the costs. Trajectories that saturate the logistic function lead to more difficult credit assignment. Specifically, when the accumulated costs are high, any additional penalty would only marginally change the termination probability, making its estimation harder. A similar argument can be made when the termination probability is low.

We emphasize that in contrast to previous work on global reward feedback in RL [Chatterji et al., 2021, Efroni et al., 2020b], which focused specifically on settings in which information is provided only at the end of an episode, the TerMDP framework provides us with additional information whenever no termination occurs, allowing us to achieve strong, local bounds of the unknown costs. This observation is crucial for the design of a computationally tractable algorithm, as we will see both in theory as well in our experiments later on.

3.1 Termination Confidence Reinforcement Learning

We are now ready to present our method for solving TerMDPs with unknown costs. Our proposed approach, which we call Termination Confidence Reinforcement Learning (TermCRL), is shown in Algorithm 1. Leveraging the local convergence guarantees of Theorem 1, we estimate the costs by maximizing the likelihood in Equation (1). We compensate for uncertainty in the

Algorithm 2 TermPG

```
require: window w, number of ensembles M, number of rollouts N, number of iterations K,
    policy gradient algorithm ALG-PG
initialize: B_pos ← ∅, B_neg ← ∅, π_θ ← random initialization
for k = 1, ..., K do
    Rollout N trajectories using π_θ, R = { s_1^i, a_1^i, r_1^i, ..., s_{t_i^*}^i, a_{t_i^*}^i, r_{t_i^*}^i }_{i=1}^N.
    for i = 1, ..., N do
        Add t_i^* − 1 negative examples (s_{max{1,l−w+1}}, a_{max{1,l−w+1}}, ..., s_l, a_l)_{l=1}^{t^*−1} to B_neg.
        Add one positive example (s_{max{1,t^*−w+1}}, a_{t^*−max{1,t^*−w+1}}, ..., s_{t^*}, a_{t^*}).
    end for
    Train bootstrap ensemble {c_{φ_m}}_{m=1}^M using binary cross entropy over data B_neg, B_pos.
    Augment states in R by s_l^i ← s_l^i ∪ Σ_{j=1}^{min{w,l}} min_m c_{φ_m}(s_{l−j}^i, a_{l−j}^i).
    Update policy π_θ ← ALG-PG(R) with dynamic discount (see Section 4.2).
end for
```

reward, transitions, and costs by incorporating optimism. We define bonuses for the reward, transition, and cost function by $b_k^r(h,s,a) = \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{n_h^k(s,a)\vee 1}}\right)$, $b_k^p(h,s,a) = \mathcal{O}\left(\sqrt{\frac{SH^2\log(1/\delta)}{n_h^k(s,a)\vee 1}}\right)$, and $b_k^c(h,s,a) = \mathcal{O}\left(\sqrt{\frac{\kappa SAHL^3}{n_h^k(s,a)\vee 1}}\log\left(\frac{1}{\delta}\right)\right)$ for some $\delta > 0$ (see Appendix H.1 for explicit definitions).

We add the reward and transition bonuses to the estimated reward (line 4), while the optimistic cost bonus is applied directly to the estimated costs (line 5). Then, a planner (line 7) solves the optimistic MDP for which the costs are known and are given by their optimistic counterparts. We refer the reader to Appendix G for further discussion on planning in TerMDPs. The following theorem provides regret guarantees for Algorithm 1. Its proof is given in Appendix I and relies on Theorem 1 and the analysis of UCRL [Auer et al., 2008, Efroni et al., 2019].

Theorem 2. *[Regret of TermCRL] With probability at least* $1-\delta$*, the regret of Algorithm 1 is*

$$\mathrm{Reg}(K) \leq \mathcal{O}\left(\sqrt{\kappa S^2 A^2 H^{8.5} L^3 K \log^3\left(\frac{SAHK}{\delta}\right)}\right).$$

Compared to the standard regret of UCRL [Auer et al., 2008], an additional $\sqrt{\kappa A H^4 L^3}$ multiplicative factor is evident in our result, which is due to the convergence rates of the costs in Theorem 1. Motivated by our theoretical results, in what follows we propose a practical approach, inspired by Algorithm 1, which utilizes local cost confidence intervals in a deep RL framework.

4 Termination Policy Gradient

Following the theoretical analysis in the previous section, we propose a practical approach for solving TerMDPs. Particularly, in this section, we devise a policy gradient method that accounts for the unknown costs leading to termination. We assume a stationary setup for which the transitions, rewards, costs, and policy are time-homogeneous. Our approach consists of three key elements: learning the costs, leveraging uncertainty estimates over costs, and constructing efficient value estimates through a dynamic cost-dependent discount factor.

Algorithm 2 describes the Termination Policy Gradient (TermPG) method, which trains an ensemble of cost networks (to estimate the costs and uncertainty) over rollouts in a policy gradient framework. We represent our policy and cost networks using neural networks with parameters $\theta, \{\phi_m\}_{m=1}^M$. At every iteration, the agent rolls out N trajectories in the environment using a parametric policy, π_θ. The rollouts are split into subtrajectories which are labeled w.r.t. the termination signal, where positive labels are used for examples that end with termination. Particularly, we split the rollouts into “windows" (i.e., subtrajectories of length w), where a rollout of length t^*, which ends with termination, is split into t^*-1 negative examples $\left(s_{\max\{1,l-w+1\}}, a_{\max\{1,l-w+1\}}, \ldots, s_l, a_l\right)_{l=1}^{t^*-1}$,

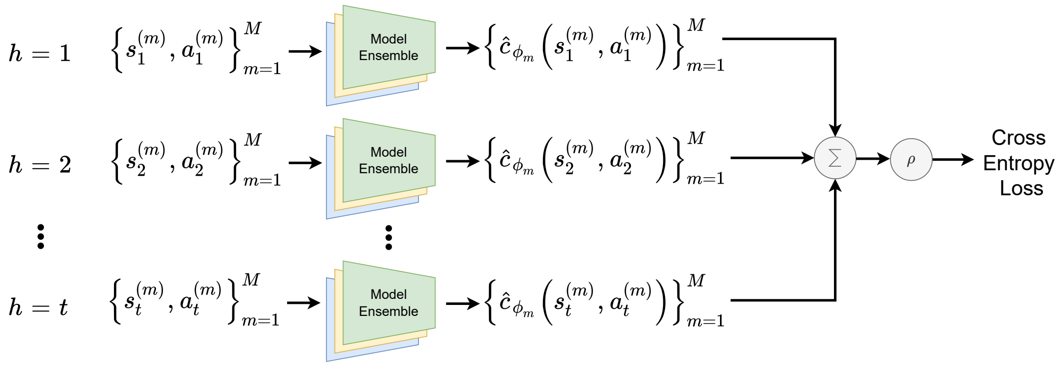

Figure 2: Block diagram of the cost training procedure. Rollouts are split into subtrajectories, labeled according to whether they end in termination. Given the dataset of labeled subtrajectories, an ensemble of M cost networks is trained end-to-end using cross-entropy with bootstrap samples (all time steps share the same ensemble).

and one positive example $\left(s_{\max\{1,t^*-w+1\}}, a_{t^*-\max\{1,t^*-w+1\}}, \dots, s_{t^*}, a_{t^*}\right)$. Similarly, a rollout of length H which does not end with termination contains H negative examples. We note that by taking finite windows, we assume the terminator "forgets" accumulated costs that are not recent - a generalization of the theoretical TerMDP model in Section 2, for which $w = H$. In Section 5, we provide experiments of misspecification of the true underlying window width, where this model assumption does not hold.

4.1 Learning the Costs

Having collected a dataset of positive and negative examples, we train a logistic regression model consisting of an ensemble of M cost networks $\{c_{\phi_m}\}_{m=1}^M$, shared across timesteps, as depicted in Figure 2. Specifically, for an example $\left(s_{\max\{1,l-w+1\}}, a_{\max\{1,l-w+1\}}, \dots, s_l, a_l\right)$ we estimate the termination probability by $\rho\left(\sum_{j=1}^{\min\{w,l\}} c_{\phi_m}(s_{l-j+1}, a_{l-j+1}) - b_m\right)$, where $\{b_m\}_{m=1}^M$ are learnable bias parameters. The parameters are then learned end-to-end using the cross entropy loss. We use the bootstrap method [Bickel and Freedman, 1981, Chua et al., 2018] over the ensemble of cost networks. This ensemble is later used in Algorithm 2 to produce optimistic estimates of the costs. Particularly, the agent policy π_θ uses the current state augmented by the optimistic cummulative predicted cost, i.e., $s_l^{\text{aug}} = (s_l, C_{\text{optimistic}})$, where $C_{\text{optimistic}} = \sum_{j=1}^{\min\{w,l\}} \min_m c_{\phi_m}(s_{l-j+1}, a_{l-j+1})$. Finally, the agent is trained with the augmented states using a policy gradient algorithm `ALG-PG` (e.g., PPO [Schulman et al., 2017], IMPALA [Espeholt et al., 2018]).

4.2 Optimistic Dynamic Discount Factor

While augmenting the state with the optimistic accumulated costs is sufficient for obtaining optimality, we propose to further leverage these estimates more explicitly – noticing that the finite horizon objective we are solving can be cast to a discounted problem. Particularly, it is well known that the discount factor $\gamma \in (0, 1)$ can be equivalently formulated as the probability of "staying alive" (see the discounted MDP framework, Puterman [2014]). Similarly, by augmenting the state s with the accumulated cost $C_h = \sum_{t=1}^h c(s_t, a_t)$, we view the probability $1 - \rho(C_h)$ as a state-dependent discount factor, capturing the probability of an agent in a TerMDP to not be terminated.

We define a dynamic, cost-dependent discount factor for value estimation. We use the state-action value function $Q(s, a, C)$ over the augmented states, defined for any s, a, C by

$$Q^\pi(s, a, C) = \mathbb{E}_\pi\left[\sum_{t=1}^H \left(\prod_{h=1}^t \gamma_h\right) r(s_t, a_t) \,\middle|\, s_1 = s, a_1 = a, C_1 = C\right],$$

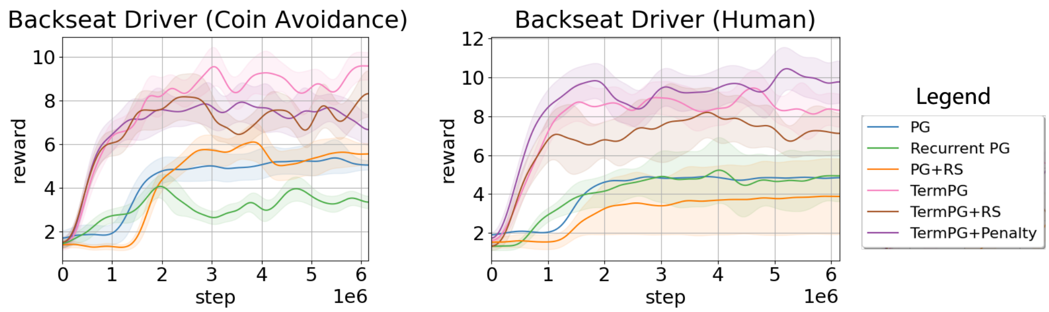

Figure 3: Mean reward with std. over five seeds of "Backseat Driver". Left: coin avoidance; right: human termination. Variants with reward shaping (RS, orange and brown) penalize the agent with a constant value upon termination. The recurrent PG variant (green) uses a history-dependent policy without learning costs. The TermPG+Penalty variant (purple) penalizes the reward at every time step using the estimated costs.

where $\gamma_h = 1 - \rho\Big(C + \sum_{i=2}^{h-1} c(s_i, a_i) - b\Big)$. This yields the Termination Bellman Equations (see Appendix C for derivation)

$$Q^\pi(s, a, C) = r(s, a) + (1 - \rho(C))\mathbb{E}_{s' \sim P(\cdot|s,a), a' \sim \pi(s')}[Q^\pi(s', a', C + c(s', a'))].$$

To incorporate uncertainty in the estimated costs, we use the optimistic accumulated costs $C_{\text{optimistic}} = \sum_{j=1}^{\min\{w,l\}} \min_m c_{\phi_m}(s_{l-j+1}, a_{l-j+1})$. Then, the discount factor becomes $\gamma(C_{\text{optimistic}}) = 1 - \rho(C_{\text{optimistic}} - b)$. Assuming that, w.h.p., optimistic costs are smaller than the true costs, the discount factor decreases as the agent exploits previously visited states.

The dynamic discount factor allows us to obtain a more accurate value estimator. In particular, we leverage the optimistic cost-dependent discount factor $\gamma(C_{\text{optimistic}})$ in our value estimation procedure, using Generalized Advantage Estimation (GAE, Schulman et al. [2015]). As we will show in the next section, using the optimistic discount factor significantly improves overall performance.

5 Experiments

In this section we evaluate the strength of our approach, comparing it to several baselines, including: **(1) PG (naive):** The standard policy gradient without additional assumptions, which ignores termination. **(2) Recurrent PG:** The standard policy gradient with a history-dependent recurrent policy (without cost estimation or dynamic discount factor). As the history is a sufficient statistic of the costs, the optimal policy is realizable. **(3) PG with Reward Shaping (RS):** We penalize the reward upon termination by a constant value, i.e., $r(s, a) - p\mathbb{1}\{s_{\text{term}}\}$, for some $p > 0$. This approach can be applied to any variant of Algorithm 2 or the methods listed above. **(4) TermPG:** Described in Algorithm 2. We additionally implemented two variants of TermPG, including: **(5) TermPG with Reward Shaping:** We penalize the reward with a constant value upon termination. **(6) TermPG with Cost Penalty:** We penalize the reward at every time step by the optimistic cost estimator, i.e., $r - \alpha C_{\text{optimistic}}$ for some $\alpha > 0$. All TermPG variants used an ensemble of three cost networks, and a dynamic cost-dependent discount factor, as described in Section 4.2. We report mean and std. of the total reward (without penalties) for all our experiments.

Backseat Driver (BDr). We simulated a driving application, using MLAgents [Juliani et al., 2018], by developing a new driving benchmark, "Backseat Driver" (depicted in Figure 4), where we tested both synthetic and human terminations. The game consists of a five lane never-ending road, with randomly instantiating vehicles and coins. The agent can switch lanes and is rewarded for overtaking vehicles. In our experiments, states were represented as top view images containing the position of the agent, nearby cars, and coins with four stacked frames. We used a finite window of length 120 for termination (30 agent decision steps), mimicking a passenger forgetting mistakes of the past.

BDr Experiment 1: Coin Avoidance. In the first experiment of Backseat Driver, coins are considered as objects the driver must avoid. The coins signify unknown preferences of the passenger, which are not explicitly provided to the agent. As the agent collects coins, a penalty is accumulated, and the agent is terminated probabilistically according to the logistic cost model in Section 2. We emphasize

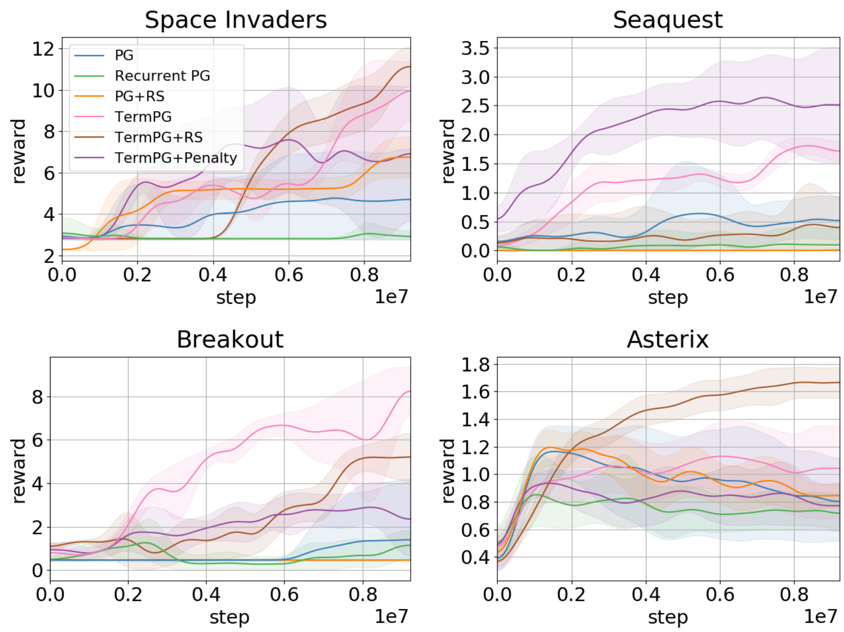

Figure 5: Results for MinAtar benchmarks. All runs were averaged over five seeds. Comparison of best performing TermPG variant to best performing PG variant (relative improvement percentage): 80% in Space Invaders, 150% in Seaquest, 410% in Breakout, and 90% in Asterix.

that, while the coins are visible to the agent (i.e., part of the agent's state), the agent only receives feedback from collecting coins through implicit terminations.

Results for Backseat Driver with coin-avoidance termination are depicted in Figure 3. We compared TermPG (pink) and its two variants (brown, purple) to the PG (blue), recurrent PG (green), and reward shaping (orange) methods described above. Our results demonstrate that TermPG significantly outperforms the history-based and penalty-based baselines. We found TermPG (pink) to perform significantly better, doubling the reward of the best PG variant. All TermPG variants converged quickly to a good solution, suggesting fast convergence of the costs (see Appendix E).

BDr Experiment 2: Human Termination. To complement our results, we evaluated human termination on Backseat Driver. For this, we generated data of termination sequences from agents of varying quality (ranging from random to expert performance). We asked five human supervisors to label subsequences of this data by terminating the agent in situations of "continual discomfort". This guideline was kept ambiguous to allow for diverse termination signals. The final dataset consisted of 512 termination examples. We then trained a model to predict human termination and implemented it into Backseat Driver to simulate termination. We refer the reader to Appendix D for specific implementation details. Figure 3 shows results for human termination in Backseat Driver. As before, a significant performance increase was evident in our experiments. Additionally, we found that using a cost penalty (purple) or termination penalty (brown) for TermPG did not greatly affect performance.

Figure 4: Backseat Driver

MinAtar. We further compared our method to the PG, recurrent PG, and reward shaping methods, on MinAtar [Young and Tian, 2019]. For each environment, we defined cost functions that do not necessarily align with the pre-specified reward, to mimic uncanny behavior that humans are expected to dislike. For example, in Breakout, the agent was penalized whenever the paddle remained in specific regions (e.g., sides of the screen), whereas in Space Invaders, the agent was penalized for

Table 1: Summary of results (top) and ablations for TermPG (bottom). Standard deviation optimism did not have significant impact on performance. Removing optimism or the dynamic discount factor had negative impact on performance. TermPG was found to be robust to model misspecifcations of the accumulated cost window.

	Backseat Driver		MinAtar			
Experiment	**Coin Avoid.**	**Human**	**Space Inv.**	**Seaquest**	**Breakout**	**Asterix**
PG	5.3 ± 0.8	4.9 ± 1.5	5.2 ± 1.8	0.6 ± 0.4	1.4 ± 2.8	0.8 ± 0.3
Recurrent PG	3.4 ± 0.21	5 ± 1.8	2.8 ± 0.05	0.1 ± 0.3	0.7 ± 0.6	0.7 ± 0.2
PG + RS	5.9 ± 1.4	7.4 ± 1.7	7.6 ± 2.3	0.4 ± 0.2	0.5 ± 0.03	0.9 ± 0.2
TermPG (ours)	$\mathbf{8.7 \pm 1.4}$	8.3 ± 1.3	$\mathbf{9.7 \pm 1.1}$	$\mathbf{1.4 \pm 0.8}$	$\mathbf{8.2 \pm 0.3}$	1 ± 0.2
TermPG + RS (ours)	$\mathbf{8.4 \pm 1.3}$	7.7 ± 0.3	$\mathbf{11.8 \pm 0.8}$	0.3 ± 0.6	5.1 ± 1	0.8 ± 0.1
TermPG + Penalty (ours)	6 ± 0.8	$\mathbf{11.8 \pm 1.5}$	7.7 ± 1.4	$\mathbf{2.4 \pm 1}$	2.3 ± 2.3	$\mathbf{1.7 \pm 0.1}$
Ablation Test	**Coin Avoid.**	**Human**	**Space Inv.**	**Seaquest**	**Breakout**	**Asterix**
Optimism with Ensemble Std.	7.6 ± 2.1	7.5 ± 1.1	2.8 ± 0.02	0.9 ± 0.6	10.9 ± 1	1 ± 0.1
No Optimism	7.8 ± 1.3	8.8 ± 1.2	5.2 ± 1.6	0.7 ± 0.3	1.3 ± 0.7	1 ± 0.1
No Dynamic Discount	6.9 ± 0.6	5.9 ± 0.7	4.4 ± 1.8	0.4 ± 0.1	0.5 ± 0.02	0.8 ± 0.1
$\times 0.5$ Window Misspecification	7.2 ± 1.1	7.2 ± 0.5	9.7 ± 3.1	2.4 ± 0.4	7.9 ± 0.8	0.8 ± 0.1
$\times 2$ Window Misspecification	8.3 ± 0.1	8.4 ± 0.2	11.1 ± 3	2.2 ± 0.2	10.3 ± 0.8	1 ± 0.1

“near misses" of enemy bullets. We refer the reader to Appendix D for specific details of the different termination cost functions.

Figure 5 depicts results on MinAtar. As with Backseat Driver, TermPG lead to significant improvement, often achieving a magnitude order as much reward as Recurrent PG. We found that adding a termination penalty and cost penalty produced mixed results, with them being sometimes useful (e.g., Space Invaders, Sequest, Asterix), yet other times harmful to performance (e.g., Breakout). Therefore, we propose to fine-tune these penalties in Algorithm 2. Finally, we note that training TermPG was, on average, 67% slower than PG, on the same machine. Nevertheless, though TermPG was somewhat more computationally expensive, it showed a significant increase in overall performance. A summary of all of our results is presented in Table 1 (top).

Ablation Studies. We present various ablations for TermPG in Table 1 (bottom). First, we tested the effect replacing the type of cost optimism in TermPG. In Section 4, cost optimism was defined using the minimum of the cost ensemble, i.e., $\min\{c_{\phi_m}\}$. Instead, we replaced the cost optimism to $C_{\text{optimistic}} = \text{mean}\{c_{\phi_m}\} - \alpha \text{std}\{c_{\phi_m}\}$, testing different values of α. Surprisingly, this change mostly decreased performance, except for Breakout, where it performed significantly better. Other ablations included removing optimism altogether (i.e., only using the mean of the ensemble), and removing the dynamic discount factor. In both cases we found a significant decrease in performance, suggesting that both elements are essential for TermPG to work properly and utilize the estimator of the unknown costs. Finally, we tested misspecifications of our model by learning with windows that were different from the environment's real cost accumulation window. In both cases, TermPG was suprisingly robust to window misspecification, as performance remained almost unaffected by it.

6 Related Work

Our setup can be linked to various fields, as listed below.

Constrained MDPs. Perhaps the most straightforward motivations for external termination stems from constraint violation [Chow et al., 2018, Efroni et al., 2020a, HasanzadeZonuzy et al., 2020], where strict or soft constraints are introduced to the agent, who must learn to satisfy them. In these setups, which are often motivated by safety [Garcıa and Fernández, 2015], the constraints are usually known. In contrast, in this work, the costs are *unknown* and only implicit termination is provided.

Reward Design. Engineering a good reward function is a hard task, for which frequent design choices may drastically affect performance [Oh et al., 2021]. Moreover, for tasks where humans are involved, it is rarely clear how to engineer a reward, as human preferences are not necessarily known, and humans are non-Markovian by nature [Clarke et al., 2013, Christiano et al., 2017]. Termination can thus be viewed as an efficient mechanism to elicit human input, allowing us to implicitly interpret human preferences and utility more robustly than trying to specify a reward.

Global Feedback in RL. Recent work considered once-per-trajectory reward feedback in RL, observing either the cumulative rewards at the end of an episode [Efroni et al., 2020b, Cohen et al., 2021] or a logistic function of trajectory-based features Chatterji et al. [2021]. While these works are based on a similar solution mechanism, our work concentrates on a new framework, which accounts for non-Markovian termination. Additionally, we provide per-state concentration guarantees of the unknown cost function, compared to global concentration bounds in previous work [Abbasi-Yadkori et al., 2011, Zhang et al., 2016, Qi et al., 2018, Abeille et al., 2021]. Using our local guarantees, we are able to construct a scalable policy gradient solution, with significant improvement over recurrent and reward shaping based approaches.

Preference-based RL. In contrast to traditional reinforcement learning, preference-based reinforcement learning (PbRL) relies on subjective opinions rather than numerical rewards. In PbRL, preferences are captured through probabilistic rankings of trajectories [Wirth et al., 2016, 2017, Xu et al., 2020]. Similar to our work, Christiano et al. [2017] use a regression model to learn a reward function that could account for the preference feedback. Our work considers a different setting in which human feedback is provided through termination, where termination and reward may not align.

7 Discussion

This paper formulated a new model to account for history-dependent exogenous termination in reinforcement learning. We defined the TerMDP framework and proposed a theoretically-guaranteed solution, as well as a practical policy-gradient approach. Our results showed significant improvement of our approach over various baselines. We stress that while it may seem as if the agent has two potentially conflicting goals—avoiding termination and maximizing reward—they are, in fact, aligned. The long-term consequences of actions need to account for longer survival which, in turn, allows for more reward collection. In what follows, we discuss κ, as factored in our regret bounds, as well possible limitations of our work.

The Role of κ As shown in Theorem 2, κ plays a significant role in the regret bound of Algorithm 1. This linear dependence is induced from the confidence bounds of Theorem 1. Informally, κ is negligible whenever the costs c and bias b are "well behaved". Suppose $\sum_{t=1}^{h} c_t(s_t^k, a_t^k) - b \gg 0$. In this case, κ would be large and termination would mostly occur after the first step. As such, estimation of the costs would be hard (see Chatterji et al. [2021]). Alternatively, suppose $\sum_{t=1}^{h} c_t(s_t^k, a_t^k) - b \ll 0$. In this case, κ would also be large. Here, credit assignment would make the cost estimation problem harder, as trajectories would span longer horizons. It is unclear, as to the writing of this work, if other solutions to TerMDPs could bring about stronger regret guarantees that significantly reduce their dependence on κ. We note that lower bounds, which include κ, have previously been established for the estimation problem (see Abeille et al. [2021], Faury et al. [2020], Jun et al. [2021]). Nevertheless, when searching for a policy which maximizes reward, it is unclear if estimation of the costs is indeed necessary for every state. We leave this direction for future work.

Limitations and Negative Societal Impact A primary limitation of our work involves the linear dependence of the logistic termination model. In some settings, it might be hard to capture true human preferences and behaviors using a linear model. Nevertheless, when measured across the full trajectory, our empirical findings show that this model is highly expressive, as we demonstrated on real human termination data (Section 5). Additionally, we note that work in inverse RL [Arora and Doshi, 2021] also assumes such linear dependence of human decisions w.r.t. reward. Future work can consider more involved hypothesis classes, building upon our work to identify the optimal tradeoff between expressivity and convergence rate.

Finally, we note a possible negative societal impact of our work. Termination is strongly motivated by humans interacting with the agent. This may be harmful if not carefully controlled, as learning incorrect or biased preferences may, in turn, result in unfavorable consequences, or if humans engage in adversarial behavior in order to mislead an agent. Our work discusses initial research in this domain. We encourage caution in real-world applications, carefully considering the possible effects of model errors, particularly in applications that affect humans.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Marc Abeille, Louis Faury, and Clément Calauzènes. Instance-wise minimax-optimal algorithms for logistic bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 3691–3699. PMLR, 2021.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

KV Bhagwat and R Subramanian. Inequalities between means of positive operators. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 83, pages 393–401. Cambridge University Press, 1978.

Peter J Bickel and David A Freedman. Some asymptotic theory for the bootstrap. *The annals of statistics*, 9(6):1196–1217, 1981.

Niladri Chatterji, Aldo Pacchiano, Peter Bartlett, and Michael Jordan. On the theory of reinforcement learning with once-per-episode feedback. *Advances in Neural Information Processing Systems*, 34, 2021.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A Lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

AM Clarke, Johannes Friedrich, Walter Senn, EM Tartaglia, Silvia Marchesotti, and Michael H Herzog. Human learning in non-markovian decision making, 2013.

Alon Cohen, Haim Kaplan, Tomer Koren, and Yishay Mansour. Online markov decision processes with aggregate bandit feedback. In *Conference on Learning Theory*, pages 1301–1329. PMLR, 2021.

Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5713–5723, 2017.

Yonathan Efroni, Nadav Merlis, Mohammad Ghavamzadeh, and Shie Mannor. Tight regret bounds for model-based reinforcement learning with greedy policies. *Advances in Neural Information Processing Systems*, 32, 2019.

Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020a.

Yonathan Efroni, Nadav Merlis, and Shie Mannor. Reinforcement learning with trajectory feedback. *arXiv preprint arXiv:2008.06036*, 2020b.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018.

Louis Faury, Marc Abeille, Clément Calauzènes, and Olivier Fercoq. Improved optimistic algorithms for logistic bandits. In *International Conference on Machine Learning*, pages 3052–3060. PMLR, 2020.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Aria HasanzadeZonuzy, Dileep M Kalathil, and Srinivas Shakkottai. Learning with safety constraints: Sample complexity of reinforcement learning for constrained mdps. *arXiv preprint arXiv:2008.00311*, 2020.

Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

Kwang-Sung Jun, Lalit Jain, Blake Mason, and Houssam Nassif. Improved confidence bounds for the linear logistic model and applications to bandits. In *International Conference on Machine Learning*, pages 5148–5157. PMLR, 2021.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, MK Ryu, and Greg Imwalle. Data center cooling using model-predictive control. *Advances in Neural Information Processing Systems*, 31, 2018.

Scott Le Vine, Alireza Zolfaghari, and John Polak. Autonomous cars: The tension between occupant experience and intersection capacity. *Transportation Research Part C: Emerging Technologies*, 52: 1–14, 2015.

Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning*, pages 3053–3062. PMLR, 2018.

Amol Mandhane, Anton Zhernov, Maribeth Rauh, Chenjie Gu, Miaosen Wang, Flora Xue, Wendy Shang, Derek Pang, Rene Claus, Ching-Han Chiang, et al. Muzero with self-competition for rate control in vp9 video compression. *arXiv preprint arXiv:2202.06626*, 2022.

Hamidreza Modares, Isura Ranatunga, Frank L Lewis, and Dan O Popa. Optimized assistive human–robot interaction using reinforcement learning. *IEEE transactions on cybernetics*, 46(3):655–667, 2015.

Inseok Oh, Seungeun Rho, Sangbin Moon, Seongho Son, Hyoil Lee, and Jinyun Chung. Creating pro-level ai for a real-time fighting game using deep reinforcement learning. *IEEE Transactions on Games*, 2021.

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

Yi Qi, Qingyun Wu, Hongning Wang, Jie Tang, and Maosong Sun. Bandit learning with implicit feedback. *Advances in Neural Information Processing Systems*, 31, 2018.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

Dilip Singh Sisodia, Somdutta Vishwakarma, and Abinash Pujahari. Evaluation of machine learning models for employee churn prediction. In *2017 international conference on inventive computing and informatics (icici)*, pages 1016–1020. IEEE, 2017.

Yi-Fan Wang, Ding-An Chiang, Mei-Hua Hsu, Cheng-Jung Lin, and I-Long Lin. A recommender system to avoid customer churn: A case study. *Expert Systems with Applications*, 36(4):8071–8075, 2009.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

Christian Wirth, Johannes Fürnkranz, and Gerhard Neumann. Model-free preference-based reinforcement learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Yichong Xu, Ruosong Wang, Lin Yang, Aarti Singh, and Artur Dubrawski. Preference-based reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 33:18784–18794, 2020.

Kenny Young and Tian Tian. Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments. *arXiv preprint arXiv:1903.03176*, 2019.

Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-Hua Zhou. Online stochastic linear optimization under one-bit feedback. In *International Conference on Machine Learning*, pages 392–401. PMLR, 2016.

Meixin Zhu, Yinhai Wang, Ziyuan Pu, Jingyun Hu, Xuesong Wang, and Ruimin Ke. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *Transportation Research Part C: Emerging Technologies*, 117:102662, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Table of Contents

ORGANIZATION OF THE APPENDIX

This appendix is organized as follows. We begin by further discussing motivations of our setting in autonomous driving and recommender system tasks in Appendix A. The first part of the appendix is then mostly focused on the empirical aspects, while the second part is mostly focused on the theoretical aspects, as well as missing proofs.

In Appendix B we discuss the TerMDP model in which costs are known. In this setting, we show that the costs are indeed sufficient for finding an optimal policy. That is, one need not care about the full history to account for termination, and only the current state and accumulated costs are needed to identify an optimal policy (i.e., achieve the same value as the optimal history-dependent policy).

In Appendix C we discuss the dynamic discount factor and derive the corresponding Termination Bellman Equations, as presented in Section 4.2. In Appendix D we discuss implementation details, including descriptions of the different environment and cost functions, TermPG implementations details and hyperparameters, and the construction of the human termination data. In Appendix E we provide some additional experimental results on the cost error and adding a cost bonus to the reward.

Appendix G is focused on the problem of approximate planning in known TerMDPs. Here, we show that, by discretizing the costs on a grid, we can achieve near optimal performance through an approximate optimistic planner.

Finally, Appendices H to K provide full proofs of Theorems 1 and 2. In Appendix H we define failure events and bonuses for optimism in Algorithm 1. Appendix I analyzes and proves the regret guarantees in Theorem 2, and Appendix J provides proof for Theorem 1. Finally in Appendix K we state auxiliary lemmas used throughout our analysis.

A Motivation

We begin by describing two concrete examples in which non-Markovian termination naturally occurs: overrides in autonomous driving and users abandoning recommender systems.

Autonomous Driving. A myriad of factors affect the quality of a driving policy. These include safety, navigation efficiency, comfort, legislation, as well as specific preferences of the passengers. As these may be difficult to characterize quantitatively, heuristic metrics are derived and optimized. When a policy is released, these unknown factors may be enforced by a driving passenger, overriding the policy. For instance, a passenger might perceive a safe driving as dangerous due to differences between human and autonomous vehicle capabilities, thus halting the driving policy. This complication may be resolved by learning how to overcome such termination from previous occurrences.

Recommender Systems. Companies that recommend items to users may find contrasting goals between maximizing revenue and overall user satisfaction. For example, top-selling items which are recommended repeatedly can antagonize the user. Other popular items might offend the user or undermine the recommendation engine's credibility. Similar to the autonomous driving problem, user abandonment constitutes a critical signal, which may be caused by continual user dissatisfaction. The original criteria should therefore be optimized while also learning and accounting for these non-Markovian and unknown preferences.

B TerMDPs with Known Costs

In this section we define the TerMDP model with known costs. We show that the accumulated costs at time step h, i.e., $C_h = \sum_{t=1}^{h-1} c(s_t, a_t)$ and the current state s_h are a sufficient statistic for computing an optimal policy π^*. That is, the history $\tau_{1:h} = (s_1, a_1, s_2, a_2, \ldots, s_{h-1}, a_{h-1})$ can be replaced by the accumulated costs C_h. To see this, we define an equivalent MDP, for which the state space is augmented by the accumulated costs, and show that an optimal policy of the augmented MDP achieves the optimal value of a history dependent optimal policy of the TerMDP.

We define the augmented MDP $\mathcal{M}_{\text{aug}} = (\mathcal{S}_{\text{aug}}, \mathcal{A}_{\text{aug}}, P_{\text{aug}}, R_{\text{aug}}, H)$, where $\mathcal{S}_{\text{aug}} = \mathcal{S} \times \mathbb{R}$ is the augmented state space, and $\mathcal{A}_{\text{aug}} = \mathcal{A}$ is the (unchanged) action space. The augmented transition function is defined for $s, C \in \mathcal{S} \times \mathbb{R}, a \in \mathcal{A}, s', C' \in \mathcal{S} \times \mathbb{R}$

$$P_{\text{aug}}(s', C' | s, C, a) = \mathbb{1}\{C' = C + c(s,a)\} \cdot \begin{cases} 1 & , s' = s = s_{\text{term}} \\ \rho(C) & , s \neq s' = s_{\text{term}} \\ (1-\rho(C))P(s'|s,a) & , s, s' \neq s_{\text{term}} \\ 0 & , \text{o.w.} \end{cases}.$$

Finally, the augmented reward function R_{aug} satisfies $r_{\text{aug},h}(s^k_{\text{aug},h} = (s^k_h, C), \tilde{a}^k_h) = r_h(s^k_h, a^k_h)$.

Next, consider the TerMDP with known costs $\mathcal{M}_T = (\mathcal{S}, \mathcal{A}, P, R, H, c)$, and let $C_h = \sum_{t=1}^{h-1} c(s_t, a_t)$. Then,

$$\begin{aligned} P(s_{h+1}, C_{h+1} | s_h, a_h, \tau_{1:h}) &= P(s_{h+1}, C_{h+1} | s_h, a_h, \tau_{1:h}, C_h) \\ &= \mathbb{1}\{C_{h+1} = C_h + c(s_h, a_h)\} \cdot \begin{cases} 1 & , s_{h+1} = s_h = s_{\text{term}} \\ \rho(C_h) & , s_h \neq s_{h+1} = s_{\text{term}} \\ (1-\rho(C_h))P(s_{h+1}|s_h,a_h) & , s_h, s_{h+1} \neq s_{\text{term}} \\ 0 & , \text{o.w.} \end{cases} \\ &= P_{\text{aug}}(s_{h+1}, C_{h+1} | s_h, C_h, a_h) = P_{\text{aug}}(s_{\text{aug},h+1} | s_{\text{aug},h}, a_h). \end{aligned}$$

To prove that the costs are sufficient for optimality, we prove a more general result. Particularly, define an MDP $(\mathcal{S}_1 \times \mathcal{S}_2, \mathcal{A}, P, r, H)$, and let $f : \mathcal{S}_2 \mapsto D$, where D is some known domain. Define the following set of deterministic policies

$$\Pi_{\text{aug}} = \{\pi : \mathcal{S}_1 \times \mathcal{S}_2 \mapsto \mathcal{A} : \exists \eta : \mathcal{S}_1 \times D \mapsto [0,1], \pi(s_1, s_2) = \eta(s_1, f(s_2))\}.$$

Define the augmented optimal value for some $s \in \mathcal{S}_1 \times \mathcal{S}_2$

$$V^*_{\text{aug},1}(s_1, s_2) = \max_{\pi \in \Pi_{\text{aug}}} \mathbb{E}\left[\sum_{t=1}^{H} r_t(s_t, a_t) \,\middle|\, s_1 = s_1, s_2 = s_2, a_t \sim \pi_t(s_1, s_2)\right].$$

We will show that if the reward does not depend on $\mathcal{S}_2$, and if $P(s'_1, f(s'_2) | s_1, s_2, a) = P(s'_1, f(s'_2) | s_1, f(s_2), a)$, then $V^*_{\text{aug},1}(s_1, s_2) = V^*_1(s_1, s_2)$. This will prove that costs are indeed sufficient, as playing any policy in Π_{aug} in the original MDP and achieve the same value. To see this, choose $\mathcal{S}_2$ to be the set of possible trajectories in the known TerMDP, and let

$$f(\tau_{1:h}) = f(s_1, a_1, s_2, a_2, \ldots s_{h-1}, a_{h-1}) = \sum_{t=1}^{h-1} c(s_t, a_t).$$

Then, since we previously showed that $r_{\text{aug},h}(s^k_{\text{aug},h} = (s^k_h, C), \tilde{a}^k_h) = r_h(s^k_h, a^k_h)$ and $P(s_{h+1}, C_{h+1} | s_h, a_h, \tau_{1:h}) = P_{\text{aug}}(s_{\text{aug},h+1} | s_{\text{aug},h}, a_h)$, this concludes our claim. The formal result is stated and proved below.

Proposition 1. *Let $\mathcal{M} = (\mathcal{S}_1 \times \mathcal{S}_2, \mathcal{A}, P, r, H)$. Assume for any $s_1, s_2 \in \mathcal{S}_1 \times \mathcal{S}_2$, $a \in \mathcal{A}$, $P(s'_1, f(s'_2) | s_1, s_2, a) = P(s'_1, f(s'_2) | s_1, f(s_2), a)$ and $r(s_1, s_2, a) = g(s_1, a)$, for some deterministic function $g : \mathcal{S}_1 \times \mathcal{A} \mapsto [0,1]$. Then, for any $s_1, s_2 \in \mathcal{S}_1 \times \mathcal{S}_2$,*

$$V^*_{aug,1}(s_1, s_2) = V^*_1(s_1, s_2).$$

Proof. We prove by induction on $h \in [H]$. For $h = H$, the result follows trivially since $V^*_{\text{aug},H}(s_1, s_2) = V^*_H(s_1, s_2) = \max_a r(s_1, s_2, a)$. Next, assume that $V^*_{\text{aug},k+1}(s_1, s_2) = V^*_{k+1}(s_1, s_2)$ for some $k \in \{1, \dots, H-1\}$ and all $s_1, s_2 \in \mathcal{S}_1 \times \mathcal{S}_2$. Then, by the Bellman Equations,

$$
\begin{aligned}
V^*_k(s_1, s_2) &= \max_{a \in \mathcal{A}} r(s_1, s_2, a) + \sum_{s'_1, s'_2 \in \mathcal{S}_1 \times \mathcal{S}_2} P(s'_1, s'_2 | s_1, s_2, a) V^*_{k+1}(s'_1, s'_2) \\
&\overset{\text{induction step}}{=} \max_{a \in \mathcal{A}} r(s_1, s_2, a) + \sum_{s'_1, s'_2 \in \mathcal{S}_1 \times \mathcal{S}_2} P(s'_1, s'_2 | s_1, s_2, a) V^*_{\text{aug},k+1}(s'_1, s'_2) \\
&= \max_{a \in \mathcal{A}} g(s_1, a) + \sum_{s'_1 \in \mathcal{S}_1} \sum_{C \in D} \sum_{s'_2 : f(s'_2) = C} P(s'_1, s'_2 | s_1, s_2, a) V^*_{\text{aug},k+1}(s'_1, s'_2)
\end{aligned}
$$

Next, by Lemma 1, there exists $U^*_{k+1} : \mathcal{S}_1 \times D \mapsto \mathbb{R}$ such that $V^*_{\text{aug},k+1}(s_1, s_2) = U^*_{k+1}(s_1, f(s_2))$, for all $s_1, s_2 \in \mathcal{S}_1 \times \mathcal{S}_2$. Then,

$$
\begin{aligned}
\sum_{C \in D} \sum_{s'_2 : f(s'_2) = C} P(s'_1, s'_2 | s_1, s_2, a) V^*_{\text{aug},k+1}(s'_1, s'_2) &= \sum_{C \in D} \sum_{s'_2 : f(s'_2) = C} P(s'_1, s'_2 | s_1, s_2, a) U^*_{k+1}(s'_1, f(s'_2)) \\
&= \sum_{C \in D} U^*_{k+1}(s'_1, C) \sum_{s'_2 : f(s'_2) = C} P(s'_1, s'_2 | s_1, s_2, a) \\
&= \sum_{C \in D} U^*_{k+1}(s'_1, C) P(s'_1, f(s'_2) = C | s_1, s_2, a).
\end{aligned}
$$

Using the assumption, we have that $P(s'_1, f(s'_2) = C | s_1, s_2, a) = P(s'_1, f(s'_2) = C | s_1, f(s_2), a)$. Then,

$$
\begin{aligned}
\sum_{C \in D} \sum_{s'_2 : f(s'_2) = C} P(s'_1, s'_2 | s_1, s_2, a) V^*_{\text{aug},k+1}(s'_1, s'_2) &= \sum_{C \in D} U^*_{k+1}(s'_1, C) P(s'_1, f(s'_2) = C | s_1, f(s_2), a) \\
&= \sum_{s'_2 \in \mathcal{S}_2} P(s'_1, s'_2 | s_1, f(s_2), a) V^*_{\text{aug},k+1}(s'_1, s'_2).
\end{aligned}
$$

where the last step follows a similar argument as above. Combining the above we get that

$$
\begin{aligned}
V^*_k(s_1, s_2) &= V^*_{\text{aug},k+1}(s'_1, s'_2) \\
&= \max_{a \in \mathcal{A}} g(s_1, a) + \sum_{s'_1, s'_2 \in \mathcal{S}_1 \times \mathcal{S}_2} P(s'_1, s'_2 | s_1, f(s_2), a) V^*_{\text{aug},k+1}(s'_1, s'_2) \\
&= \max_{\eta : \mathcal{S}_1 \times D \mapsto [0,1]} g(s_1, \eta(s_1, f(s_2))) + \sum_{s'_1, s'_2 \in \mathcal{S}_1 \times \mathcal{S}_2} P(s'_1, s'_2 | s_1, f(s_2), \eta(s_1, f(s_2))) V^*_{\text{aug},k+1}(s'_1, s'_2) \\
&= \max_{\eta : \mathcal{S}_1 \times D \mapsto [0,1]} r(s_1, s_2, \eta(s_1, f(s_2))) + \sum_{s'_1, s'_2 \in \mathcal{S}_1 \times \mathcal{S}_2} P(s'_1, s'_2 | s_1, s_2, \eta(s_1, f(s_2))) V^*_{\text{aug},k+1}(s'_1, s'_2) \\
&= \max_{\pi \in \Pi_{\text{aug}}} r(s_1, s_2, \pi(s_1, s_2)) + \sum_{s'_1, s'_2 \in \mathcal{S}_1 \times \mathcal{S}_2} P(s'_1, s'_2 | s_1, s_2, \pi(s_1, s_2)) V^*_{\text{aug},k+1}(s'_1, s'_2) \\
&= V^*_{\text{aug},k}(s_1, s_2)
\end{aligned}
$$

This completes the proof. □

Lemma 1. *Let $\mathcal{M} = (\mathcal{S}_1 \times \mathcal{S}_2, \mathcal{A}, P, r, H)$. Assume for any $s_1, s_2 \in \mathcal{S}_1 \times \mathcal{S}_2$, $a \in \mathcal{A}$, $P(s'_1, f(s'_2) | s_1, s_2, a) = P(s'_1, f(s'_2) | s_1, f(s_2), a)$ and $r(s_1, s_2, a) = g(s_1, a)$, for some deterministic function $g : \mathcal{S}_1 \times \mathcal{A} \mapsto [0, 1]$. Then, for any $k \in [H]$, there exists $U^*_k : \mathcal{S}_1 \times D \mapsto \mathbb{R}$ such that $V^*_{aug,k}(s_1, s_2) = U^*_k(s_1, f(s_2))$, for all $s_1, s_2 \in \mathcal{S}_1 \times \mathcal{S}_2$.*

Proof. We prove by induction on k. For $k = H$, the result follows trivially as $V^*_{\text{aug},H}(s_1, s_2) = \max_a g(s_1, a)$ for all $s_1, s_2 \in \mathcal{S}_1 \times \mathcal{S}_2$. Otherwise, we the $V^*_{\text{aug},k+1}(s_1, s_2) = U^*(s_1, f(s_2))$ is true for some $k \in \{1, \ldots, H-1\}$. Then,

$$
\begin{aligned}
V^*_{\text{aug},k}(s_1, s_2) &= \max_{\pi \in \Pi_{\text{aug}}} r(s_1, s_2, \pi(s_1, f(s_2))) + \sum_{s_1', s_2' \in \mathcal{S}_1 \times \mathcal{S}_2} P(s_1', s_2' | s_1, s_2, \pi(s_1, f(s_2))) V^*_{k+1}(s_1', s_2') \\
&\overset{\text{induction step}}{=} \max_{\pi \in \Pi_{\text{aug}}} r(s_1, s_2, \pi(s_1, f(s_2))) + \sum_{s_1', s_2' \in \mathcal{S}_1 \times \mathcal{S}_2} P(s_1', s_2' | s_1, s_2, \pi(s_1, f(s_2))) U^*_{k+1}(s_1', f(s_2')) \\
&= \max_{\pi \in \Pi_{\text{aug}}} g(s_1, \pi(s_1, f(s_2))) + \sum_{s_1' \in \mathcal{S}_1} \sum_{C \in D} \sum_{s_2' : f(s_2') = C} P(s_1', s_2' | s_1, s_2, \pi(s_1, f(s_2))) U^*_{k+1}(s_1', f(s_2')) \\
&= \max_{\pi \in \Pi_{\text{aug}}} g(s_1, \pi(s_1, f(s_2))) + \sum_{s_1' \in \mathcal{S}_1} \sum_{C \in D} U^*_{k+1}(s_1', C) \sum_{s_2' : f(s_2') = C} P(s_1', s_2' | s_1, s_2, \pi(s_1, f(s_2))) \\
&= \max_{\pi \in \Pi_{\text{aug}}} g(s_1, \pi(s_1, f(s_2))) + \sum_{s_1' \in \mathcal{S}_1} \sum_{f(s_2') \in D} P(s_1', f(s_2') | s_1, s_2, \pi(s_1, f(s_2))) U^*_{k+1}(s_1', f(s_2')) \\
&= \max_{\pi \in \Pi_{\text{aug}}} g(s_1, \pi(s_1, f(s_2))) + \sum_{s_1' \in \mathcal{S}_1} \sum_{f(s_2') \in D} P(s_1', f(s_2') | s_1, f(s_2), \pi(s_1, f(s_2))) U^*_{k+1}(s_1', f(s_2'))
\end{aligned}
$$

By defining

$$
U^*_k(s_1, f(s_2)) = \max_{\pi \in \Pi_{\text{aug}}} g(s_1, \pi(s_1, f(s_2))) + \sum_{s_1' \in \mathcal{S}_1} \sum_{f(s_2') \in D} P(s_1', f(s_2') | s_1, f(s_2), \pi(s_1, f(s_2))) U^*_{k+1}(s_1', f(s_2')),
$$

we see that $V^*_{\text{aug},k}(s_1, s_2) = U^*_k(s_1, f(s_2))$, for any $s_1, s_2 \in \mathcal{S}_1, \mathcal{S}_2$, and the proof is complete. $\square$

C Dynamic Discount Factor

In this section we derive the Termination Bellman Equations, as defined in Section 4.2. The state action value function for a TerMDP with known costs (Appendix B) is defined for $s, C \in \mathcal{S} \times \mathbb{R}$, $a \in \mathcal{A}$, $h \in [H]$, and any policy π by

$$Q_h^\pi(s,C,a) = \mathbb{E}_\pi\left[\sum_{t=h}^{H}\left(\prod_{j=h+1}^{t}\gamma_j\right)r_t(s_t,a_t)\,\middle|\, s_h = s, C_h = C, a_h = a\right],$$

where we denote

$$\gamma_j = 1 - \rho(C_j - b). \tag{2}$$

Then, for any $h \in [H-1]$ we have that

$$\begin{aligned}
&Q_h^\pi(s,C,a)\\
&= \mathbb{E}_\pi\left[\sum_{t=h}^{H}\left(\prod_{j=h+1}^{t}\gamma_j\right)r_t(s_t,a_t)\,\middle|\, s_h = s, C_h = C, a_h = a\right]\\
&= r_h(s,a) + \gamma_{h+1}\mathbb{E}_\pi\left[\sum_{t=h+1}^{H}\left(\prod_{j=h+2}^{t}\gamma_j\right)r_t(s_t,a_t)\,\middle|\, s_h = s, C_h = C, a_h = a\right]\\
&= r_h(s,a) + \gamma_{h+1}\mathbb{E}_{s'\sim P(\cdot|s,a),a'\sim\pi(s')}\left[\mathbb{E}_\pi\left[\sum_{t=h+1}^{H}\left(\prod_{j=h+2}^{t}\gamma_j\right)r_t(s_t,a_t)\,\middle|\, \substack{s_h=s,C_h=C,a_h=a,\\ s_{h+1}=s',C_{h+1}=C+s(s',a'),a_{h+1}=a'}\right]\right]\\
&= r_h(s,a) + \gamma_{h+1}\mathbb{E}_{s'\sim P(\cdot|s,a),a'\sim\pi(s')}\left[\mathbb{E}_\pi\left[\sum_{t=h+1}^{H}\left(\prod_{j=h+2}^{t}\gamma_j\right)r_t(s_t,a_t)\,\middle|\, s_{h+1}=s',C_{h+1}=C+s(s',a'),a_{h+1}=a'\right]\right]\\
&= r_h(s,a) + \gamma_{h+1}\mathbb{E}_{s'\sim P(\cdot|s,a),a'\sim\pi(s')}\left[Q_{h+1}^\pi(s', C + c(s',a'), a')\right]\\
&= r_h(s,a) + (1-\rho(C-b))\mathbb{E}_{s'\sim P(\cdot|s,a),a'\sim\pi(s')}\left[Q_{h+1}^\pi(s', C + c(s',a'), a')\right].
\end{aligned}$$

We get the Termination Bellman Equations

$$Q_h^\pi(s,C,a) = r_h(s,a) + (1-\rho(C-b))\mathbb{E}_{s'\sim P(\cdot|s,a),a'\sim\pi(s')}\left[Q_{h+1}^\pi(s', C + c(s',a'), a')\right].$$

Similarly, we can define an infinite horizon setting, for which

$$Q^\pi(s,C,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\left(\prod_{j=2}^{t}\gamma_j\right)r_t(s_t,a_t)\,\middle|\, s_1 = s, C_1 = C, a_1 = a\right],$$

which yields the infinite horizon Termination Bellman Equations

$$Q^\pi(s,C,a) = r(s,a) + (1-\rho(C-b))\mathbb{E}_{s'\sim P(\cdot|s,a),a'\sim\pi(s')}[Q^\pi(s', C + c(s',a'), a')]$$

Having defined the Termination Bellman Equations, we use them for value estimation. Specifically, we consider estimating the value by minimizing the TD-error, for some s, C, a, s', a'

$$\left|Q(s,C,a) - r(s,a) - (1-\rho(C-b))Q(s', C + c(s',a'), a')\right|.$$

In our implementation, we used Generalized Advantage Estimation (GAE, Schulman et al. [2015]), which uses an exponential average over estimates of the TD-error. We used the discount factor in Equation (2) for the TD estimates. That is, an exponential average over

$$\hat{A}^{(n)} = \sum_{t=0}^{\infty}\left(\prod_{j=2}^{n-1}\gamma_j\right)r_t(s_t,a_t) + \left(\prod_{j=2}^{n}\gamma_j\right)\hat{V}(s_n) - \hat{V}(s_1).$$

See Schulman et al. [2015] for specific implementation details of GAE.

D Implementation Details

In this section we describe in detail the implementation details of the used environments, cost functions, algorithm, and human termination procedure.

D.1 Environments

We begin by describing the environments and cost functions used in our paper. In all environments we used a bias $b = 6$ for termination, which was unknown to the training agent. For all our environments (except human termination, see Appendix D.3), we used a window size of $w = 30$.

Backseat Driver. The environment was built using Unity's MLAgents [Juliani et al., 2018]. We incorporated a reward and cost function in the environment as follows. A reward of $+0.1$ was given to the agent for any car that was onscreen and behind it (i.e., the agent was awarded for overtaking other vehicles). The environment was set to accumulate a cost of $+1$ for every coin that was collected. No negative reward was given by the environment upon death or termination.

In Backseat Driver, the agent can take one of three actions: change lane left, change lane right, or stay in place. If the agent hits another car the episode is terminated and the agent is reset. The state is represented by a stack of three binary maps consisting of the agent position, the other vehicle positions, and the coin positions (top view). Similar to previous work, we used the past four time steps (stacked) as the agent's state. That is, at time t, the agent receives 12 binary maps consisting of the agent position, other vehicle positions, and coin positions for steps $t, t-1, t-2, t-3$. For efficient learning, we repeated every action four times in the environment.

MinAtar. We used four of the original MinAtar benchmarks [Young and Tian, 2019], enforcing termination using a predefined cost-function. For each of the environments, we used a cost function which was dependent on the agent's position w.r.t. objects and areas of the state. Specifically, we defined costs as follows.

- **Space Invaders:** The agent controls a spaceship and must shoot other alien spacecrafts, while they also shoot bullets at the agent.

  The agent was penalized ($c = 1$) for bullets passing at distance $d = 1$ from its current position. That is, near-misses of enemy bullets penalized the player. Ideally, the agent must learn to play the game while avoiding dangerous states in which one erroneous action can lose the game.
- **Seaquest:** In this environment, the agent controls a submarine. The agent must shoot enemies, avoid hitting objects (enemies, fish, or bullets), and carefully replenish its oxygen by moving to the surface.

  The agent was penalized ($c = 1$) whenever it was positioned mid-depth. That is, denoting by D the maximum reachable depth, the agent was penalized whenever it was positioned at depth $D/2$. Ideally, the agent would remain close to the surface or deep in the water, avoiding unnecessary switches, to minimize termination.
- **Breakout:** In this game, the agent controls a paddle and can move it (left or right), bouncing a ball, which can hit rows of bricks. The agent is rewarded for breaking these bricks. If the paddle misses the ball, the agent loses and resets.

  The agent was penalized ($c = 1$) whenever it was positioned at the far-most left or right positions of the screen, i.e., close to the edges of the screen. Ideally, the agent would remain close to the center of the screen, where it can quickly reach the ball, and avoid getting "stuck" at the sides of the screen.
- **Asterix:** The agent can move in any of the directions: up, down, left, or right, while avoiding spawned enemies, and picking up treasure. If the player hits an enemy, the agent loses and resets.

  The agent was penalized ($c = 1$) whenever it was positioned at distance $d = 1$ from an enemy. Similar to previous environments, this cost function was used to encourage the agent to stay away from enemies. Ideally, the agent should not be too close to enemies, for which one erroneous action could lose the game.

Table 2: Hyperparameters used to train PPO agent

Name	Value	Comments
Batch size	32	
Learning rate	5×10^{-5}	
Rollout size	1024	
Num epochs	3	How many training epochs to do after each rollout
Entropy coef.	0	
kl coef	0.2	Initial coefficient for KL divergence
kl target	0.01	Target value for KL divergence
GAE λ	1	The GAE (lambda) parameter
Num workers	8	

Table 3: CNN architecture hyperparameters for agent policy and cost networks

Input size	42×42
Num Filters	$16, 32, 256$
Stride	$4, 4, 11$
Padding	$2, 2, 1$
Activations	Relu
Post-network FC hidden	$400, 256$

D.2 TermPG

TermPG (Algorithm 2) uses two key elements - a policy gradient algorithm (`ALG-PG`), and a training procedure for learning the costs and designing a dynamic cost-dependent discount factor, as described in Section 4. For the policy gradient algorithm, we used Proximal Policy Optimization (PPO, see Schulman et al. [2017]), implemented in RLlib [Liang et al., 2018]. We mostly used the default hyperparameters – full specification is given in Table 2. We used the same hyperparemeters for the TermPG variants and the standard PG variants described in Section 5, with one important exception – the PG variants used a constant discount factor $\gamma = 0.99$, whereas the TermPG variants did not use a discount factor [6], as the dynamic discount factor was used instead (Section 4.2). The agent used a convolutional neural network (CNN), with hyperparameters given Table 3.

For learning the costs, we used a model of three identical cost networks. At every iteration, rollouts that were collected in the environments were added to a finite buffer (FIFO) for training the cost networks. We labeled the rollouts according to whether they ended in termination. For training the cost networks we then sampled a (different) rollout from the buffer for each of the cost networks. We then split each of the rollouts to their respective windows ($t^* - 1$ negative labels and 1 positive label for a trajectory ending in termination). Finally, we trained them the cost networks end-to-end using the cross entropy loss (see Figure 2). We repeated this procedure for 30 steps. We used the same network for training the cost model as we used for the PPO agent, with hyperparameters given in Table 3. Hyperparameters for the TermPG algorithm are given in Table 4.

D.3 Human Termination Data

In this subsection we describe in detail our process of creating and training the human termination data for Backseat Driver. To begin, we used the Backseat Environment without termination (i.e., we considered the standard infinite horizon MDP). We trained an agent using PPO on the infinite horizon environment (without termination), saving snapshots of the agent during training. Once training was over we used the snapshots to generate trajectories from the different agents, constructing a large dataset of over 5000 trajectories of different quality.

To obtain termination feedback, we randomly sampled trajectories from the generated dataset and played them back to a human supervisor, who was instructed to terminate the agent. The instructions

[6] We also used a constant discount factor of $\gamma = 0.99$ for the ablation test of removing the dynamic discount factor in TermPG (Section 5, Table 1).

Table 4: TermPG hyperparameters

Name	Value	Comments
Learning rate	1×10^{-3}	
Bonus coefficient	1	Coefficient used for cost bonus (optimism)
Num Ensemble	3	
Cost-net train steps	30	
Cost-net batch size	t^*	Each network receives all windows in trajectory (t^* examples for every network)
Replay buffer size	1000	Number of trajectories held in buffer
Window size	30	

for termination were to terminate the agent whenever it drives in a manner that is uncomfortable. In many scenarios the human supervisors terminated the agent when it switched lanes back and forth for no apparent reason (e.g., not for overtaking other vehicles). We used five different human supervisors for labeling random trajectories in the dataset, generating a total of 512 positive termination examples.

Once the data generation was complete, we trained a classifier for the termination problem with different window size. We found that a window size of $w = 35$ best fit the termination accuracy (on a validation set). Finally, we used the trained model as an estimated termination signal for Backseat Driver. We emphasize that, while we used the same termination window ($w = 35$) for training the TermPG agent, we ran experiments with $\times x2$, $\times 0.5$ misspecification of window size, showing that these did not hurt performance (see Section 5, Table 1).

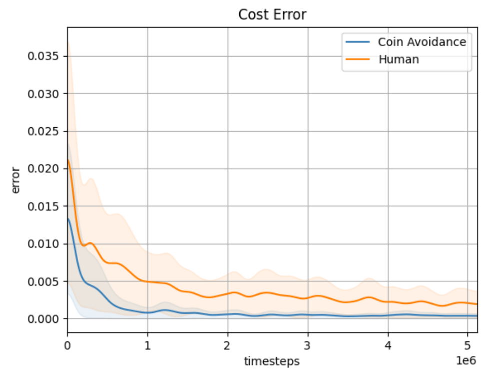

Figure 6: Cost Error for Backseat Driver

E Additional Results

Cost Error. We plot the cost error (l_2 norm) of the estimated costs in Backseat Driver in both the synthetic and human termination settings in Figure 6. It can be seen that costs converge quickly, allowing for efficient learning. The speedy convergence suggests that TermPG can utilize the costs efficiently, enabling it to converge quickly to a good solution, as seen in our experiments in Section 5.

Cost Bonus. In TermPG (Algorithm 2) optimism is used by augmenting the state with an optimistic cost function. Nevertheless, this information is implicit, as it only passes through the state. We tested the affect of adding optimism in costs (to encourage exploration in areas of uncertainty in costs) directly through the reward function. Specifically, we used an uncertainty (defined by the ensemble of cost networks) to add a bonus to the reward

$$r(s_t, a_t) \leftarrow r(s_t, a_t) + \alpha U(C_t),$$

where $U(C_t)$ is either the max-min difference or the standard deviation in the ensemble outputs, and $\alpha > 0$ is a hyperparameter for choosing the degree of optimism in the reward.

Results for training TermPG with a cost bonus are presented in Table 5. Evidently, adding a cost bonus mostly hurt performance, even with small values of α. These results suggest that adding additional cost bonus to the reward is not beneficial, and using optimism in the states is sufficient.

Table 5: Cost bonus for TermPG (other TermPG results are provided for reference.)

	Backseat Driver		**MinAtar**			
Experiment	**Coin Avoid.**	**Human**	**Space Inv.**	**Seaquest**	**Breakout**	**Asterix**
TermPG	$\mathbf{8.7 \pm 1.4}$	8.3 ± 1.3	$\mathbf{9.7 \pm 1.1}$	$\mathbf{1.4 \pm 0.8}$	$\mathbf{8.2 \pm 0.3}$	1 ± 0.2
TermPG + RS	$\mathbf{8.4 \pm 1.3}$	7.7 ± 0.3	$\mathbf{11.8 \pm 0.8}$	0.3 ± 0.6	5.1 ± 1	0.8 ± 0.1
TermPG + Penalty	6 ± 0.8	$\mathbf{11.8 \pm 1.5}$	7.7 ± 1.4	$\mathbf{2.4 \pm 1}$	2.3 ± 2.3	$\mathbf{1.7 \pm 0.1}$
Cost Bonus with $\alpha = 0.1$	6.6 ± 0.45	7.9 ± 0.7	6.5 ± 1.46	1.4 ± 1.1	2.2 ± 0.2	0.9 ± 0.2
Cost Bonus with $\alpha = 1$	7.3 ± 0.7	4.8 ± 1.14	2.8 ± 0.02	1.9 ± 0.1	0.2 ± 0.05	0.6 ± 0.3

F Additional Theoretical Notations

This section adds further notations needed for the sections that follow. Particularly, we define notations that are not provided in the paper for clarity, yet are beneficial for our theoretical analysis.

Throughout the paper, we work w.r.t. the natural filtration

$$\mathcal{F}_k = \sigma\Big(\big\{(s_h^1, a_h^1, R_h^1)\big\}_{h=1}^{H}, \dots, \big\{(s_h^k, a_h^k, R_h^k)\big\}_{h=1}^{H}, s_1^{k+1}\Big).$$

We use the notation $\tau_{1:h}^k$ when referring to the trajectory of the agent at the k^{th} episode. For brevity, we denote $\tau^k = \tau_{1:H}^k$. Similarly, we denote the empirical visitation up to the h^{th} time step $\hat{d}_h^k(t, s, a) = \mathbb{1}\big\{t \le h, s_t^k = s, a_t^k = a\big\}$. With abuse of notation, we assume the bias term b is consumed by the cost c, such that the first state always contains the bias term, i.e., $c(s_1, a_1) := c(s_1, a_1) - b$. Using this notation, we can write the termination probability after the h^{th} time step in vector notation as $\rho\big(\langle \hat{d}_h^k, c\rangle\big)$. We then define the total transition function by $T_h^{P,c}(s'|s_h^k, a_h^k, \tau_{1:h}^k) = \Big(1 - \rho\big(\langle \hat{d}_h^k, c\rangle\big)\Big) P_h(s'|s_h^k, a_h^k)$ for any non-terminal $s' \in \mathcal{S}$ and $T_h^{P,c}(s_{\text{term}}|s_h^k, a_h^k, \tau_{1:h}^k) = \rho\big(\langle \hat{d}_h^k, c\rangle\big)$. For brevity, when working with the real kernel and costs P, c, we omit them from the notation and use T_h. Finally we denote by $\bar{X}$ optimism for some quantity X, e.g., $\bar{V}$ is the optimistic value.

G Approximate Planning in TerMDPs

We consider the following planning procedure for a known TerMDP $\mathcal{M}_T = (\mathcal{S}, \mathcal{A}, P, R, H, c)$ (see Appendix B). For the approximate planning, we quantize the costs in this TerMDP to a resolution Δc. Concretely, denote the lattice of resolution Δx by $\mathcal{G}_{\Delta x} = \{\dots, -2\Delta x, \Delta x, 0, \Delta x, 2\Delta x, \dots\}$. We then define the discretized cost function, for a discretization parameter Δc as

$$c_q = \left\lfloor \frac{c}{\Delta c} \right\rfloor \Delta c$$

In this case, the accumulated cost must also lie on a grid such that $C = \sum_t c_{q,t} \in \mathcal{G}_{\Delta c}$. We denote by $c_{\max} = \max_{s,a} |c(s,a)|$. Notice that $c_{\max} \leq L$, where L is known, but $c_{\max}$ is usually much smaller. We denote the number of bins of the costs by N, which is generally bounded by $N \leq 2\frac{c_{\max}}{\Delta c} + 1$. For non-negative costs, we will show that a much smaller N suffices (Appendix G.1.2). Then, due to the grid structure, the accumulated costs can obtain at most HN different values. This allows us to perform standard planning procedure in episodic MDPs (using dynamic programming/value iteration) with a state space of size NHS, which requires computation complexity of $\mathcal{O}\big(H^3 A S^2 N^2\big)$ [Puterman, 2014]. In the rest of this section, we show how to choose Δc so that this approximate planner will yield a near-optimal policy. Particularly, we show that $\Delta c = \mathcal{O}(1/\sqrt{K})$ maintains the same regret guarantees (namely, incurs lower-order regret penalty) while enabling tractable planning.

G.1 Approximation Bounds for Quantized TerMDPs

Denote the quantized TerMDP by $\mathcal{M}_T^q = (\mathcal{S}, \mathcal{A}, P, R, H, c_q)$. For a policy π, we denote by V^π, V_q^π its value in $\mathcal{M}_T$ and $\mathcal{M}_T^q$, respectively. We define a policy π_q as the solution to the planning problem in the quantized TerMDP. In what follows we show that for any $\epsilon > 0$ one can ensure $V_1^*(s_1) - V_1^{\pi_q}(s_1) < \epsilon$ for some choice of Δc.

Notice that we always rounded-down the costs (decreased the termination probability) and, thus, $V^\pi \leq V_q^\pi$ for any π. Particularly, for π^*, we get that

$$V^* = V^{\pi^*} \overset{\text{optimism}}{\leq} V_q^{\pi^*} \overset{\substack{\text{optimality of} \\ \pi_q \text{ in } \mathcal{M}_T^q}}{\leq} V_q^{\pi_q} = V^{\pi_q} + \big(V_q^{\pi_q} - V^{\pi_q}\big).$$

Therefore, it is enough to show that $V_q^{\pi_q} - V^{\pi_q} < \epsilon$. We will show that a stronger results, that $V_q^\pi - V^\pi < \epsilon$ for any π.

By the value difference lemma (Lemma 8), for any policy π

$$\begin{aligned}
V_{q,1}^\pi(s_1) - V_1^\pi(s_1) &= \sum_{h=1}^H \mathbb{E}\Big[\Big(T_h^{P,c_q} - T_h^{P,c}\Big)(\cdot|s_h, a_h, \tau_{1:h})^T V_1^\pi(\cdot, \tau_{1:h+1})\Big] \\
&\leq \sum_{h=1}^H \mathbb{E}\Big[\Big\|\Big(T_h^{P,c_q} - T_h^{P,c}\Big)(\cdot|s_h, a_h, \tau_{1:h})\Big\|_1 \|V_1^\pi(\cdot, \tau_{1:h+1})\|_\infty\Big] \\
&\leq H \sum_{h=1}^H \mathbb{E}\Big[\Big\|\Big(T_h^{P,c_q} - T_h^{P,c}\Big)(\cdot|s_h, a_h, \tau_{1:h})\Big\|_1\Big] \\
&= H \sum_{h=1}^H \mathbb{E}\left[\Big|\rho\big(\langle \hat{d}_h, c_q\rangle\big) - \rho\big(\langle \hat{d}_h, c\rangle\big)\Big| \underbrace{\|P_h(\cdot|s_h, a_h)\|_1}_{=1} + \Big|\Big(1 - \rho\big(\langle \hat{d}_h, c_q\rangle\big)\Big) - \Big(1 - \rho\big(\langle \hat{d}_h, c\rangle\big)\Big)\Big|\right] \\
&= 2H \sum_{h=1}^H \mathbb{E}\Big[\Big|\rho\big(\langle \hat{d}_h, c_q\rangle\big) - \rho\big(\langle \hat{d}_h, c\rangle\big)\Big|\Big]
\end{aligned}$$

G.1.1 General Case: Signed Costs

By the Lipschitz-continuity of the logistic function, we have that

$$\begin{aligned}
V^{\pi}_{q,1}(s_1) - V^{\pi}_{1}(s_1) &\le 2H \sum_{h=1}^{H} \mathbb{E}\left[\left|\rho\big(\langle \hat{d}_h, c_q\rangle\big) - \rho\big(\langle \hat{d}_h, c\rangle\big)\right|\right] \\
&\le \frac{H}{2} \sum_{h=1}^{H} \mathbb{E}\left[\left|\langle \hat{d}_h, c_q - c\rangle\right|\right] \\
&\le \frac{H}{2} \sum_{h=1}^{H} \mathbb{E}\left[\underbrace{\left\|\hat{d}_h\right\|_1}_{=h} \underbrace{\|c_q - c\|_\infty}_{\le \Delta c}\right] \\
&\le \frac{H^3 \Delta c}{2}
\end{aligned}$$

We get that, for any $\epsilon > 0$, using $\Delta c \le \frac{2\epsilon}{H^3}$ we get that $V^{\pi}_{q,1}(s_1) - V^{\pi}_{1}(s_1) \le \epsilon$. Note that, this choice of grid resolution induces $N = \mathcal{O}\big(H^3 \frac{c_{\max}}{\epsilon}\big)$

G.1.2 Better Approximation for Non-Negative Costs

We remind the reader that in Appendix E we incorporated the bias b into the costs c, such that $c(s_1, a_1) := c(s_1, a_1) - b$. This was helpful for utilizing inner product notations in our analysis. In this part, we assume the costs are positive, yet we do not limit the choice of bias b. Specifically, we assume that for all $h \ge 2$, and for all $s, a \in \mathcal{S} \times \mathcal{A}$, $c(s_h, a_h) \ge 0$. We further assume that for all $s, a \in \mathcal{S} \times \mathcal{A}$, $c(s_1, a_1) \ge -b$. This relates to a TerMDP with positive costs and an arbitrary bias b.

Let $C^* \in (0, Hc_{\max}]$ (will be explicitly chosen later). We show that, in the case of non-negative costs, it can be beneficial to clip the accumulated cost, once it is larger than C^* (even if it is smaller than $Hc_{\max}$). Particularly, if $\sum_{t=1}^{h} c_{q,t}(s_t, a_t) \ge C^*$, we represent the accumulated cost by C^*. This, in turn, implies that the accumulated costs can have at most $N = \left\lfloor \frac{C^*+b}{\Delta c} \right\rfloor + 1$ bins.

Notice that by the definition of the logistic function, for any C_0 and any $C \ge C_0$, it holds that

$$\rho(C) \ge \rho(C_0) = (1 + \exp(-C_0))^{-1} \ge 1 - \exp(-C_0).$$

Therefore, since $\rho(C) \le 1$, we have for any $C \ge C_0$ that $|\rho(C) - \rho(C_0| \le \exp(-C_0)$.

Also, denote $h^* = \min\Big\{h \in [H] : \langle \hat{d}_h, c_q\rangle > C^*\Big\}$. Then,

$$\begin{aligned}
V^{\pi}_{q,1}(s_1) - V^{\pi}_{1}(s_1) &\le 2H \sum_{h=1}^{H} \mathbb{E}\left[\left|\rho\big(\langle \hat{d}_h, c_q\rangle\big) - \rho\big(\langle \hat{d}_h, c\rangle\big)\right|\right] \\
&= 2H\mathbb{E}\left[\sum_{h=1}^{h^*-1} \left|\rho\big(\langle \hat{d}_h, c_q\rangle\big) - \rho\big(\langle \hat{d}_h, c\rangle\big)\right| + \sum_{h=h^*}^{H} \left|\rho\big(\langle \hat{d}_h, c_q\rangle\big) - \rho\big(\langle \hat{d}_h, c\rangle\big)\right|\right] \\
&\le 2H\mathbb{E}\left[\frac{1}{4}(h^*-1)^2 \Delta c + (H - h^*)\exp(-C^*)\right] \\
&\le \frac{H^3 \Delta c}{2} + 2H^2 \exp(-C^*).
\end{aligned}$$

In the second inequality, we also used the lipschitz property of the logistic function. Moreover, we used the fact that the costs are non-negative – if the accumulated cost is larger than C^* at h^*, then it is larger than C^* for any $h \ge h^*$.

Finally, for any $\epsilon > 0$, we set $\Delta c = \frac{\epsilon}{H^3}$ and $C^* = \log\Big(\frac{4H^2}{\epsilon}\Big)$, which ensure an error smaller than ϵ. This parameter choice induces $N = \mathcal{O}\left(H^3 \frac{\log\left(\frac{H^2}{\epsilon}\right) + b}{\epsilon}\right)$ – potentially much smaller than the general case for which we get $\mathcal{O}\big(H^3 \frac{c_{\max}}{\epsilon}\big)$.

G.2 Regret Bound for Approximate Planner

We integrate the approximate planner into Algorithm 1 and show that the regret bound is only mildly affected for $\epsilon = \mathcal{O}\left(1/\sqrt{K}\right)$ (or $\Delta c = \mathcal{O}\left(\frac{1}{H^3\sqrt{K}}\right)$). Particularly, instead of accurately solving the TerMDP $\left(\mathcal{S}, \mathcal{A}, H, \bar{r}^k, \hat{P}^k, \bar{c}^k\right)$, we apply our approximate planner and denote its output policy by π_q^k. Its value in the optimistic TerMDP is denoted by $\bar{V}_1^{\pi_q^k}(s_1^k)$.

Now, following the regret analysis of Appendix I, we have that

$$\begin{aligned}
\mathrm{Reg}(K) &= \sum_{k=1}^{K} V_1^*(s_1^k) - V_1^{\pi_q^k}(s_1^k) \\
&\leq \sum_{k=1}^{K} \bar{V}_1^k(s_1^k) - V_1^{\pi_q^k}(s_1^k) \\
&= \sum_{k=1}^{K} \bar{V}_1^{\pi_q^k}(s_1^k) - V_1^{\pi_q^k}(s_1^k) + \sum_{k=1}^{K} \bar{V}_1^k(s_1^k) - \bar{V}_1^{\pi_q^k}(s_1^k)
\end{aligned}$$

The first term can be bounded exactly as in the regret analysis of Appendix I, while the second bound is controlled by the approximation error of the planner, namely ϵK. Thus, choosing $\epsilon = \mathcal{O}\left(1/\sqrt{K}\right)$ leads to a negligible second term, allowing efficient planning while maintaining the stated regret bound of Theorem 2.

H Failure Events and Optimism

In this section we focus on defining failure events and derive the bonuses described in Section 3. Particularly, we will use these to show that the optimistic TerMDP (line 7 of Algorithm 1), used for planning, satisfies the needed optimism (i.e., larger than the optimal value), which will be used in the proof of Theorem 2.

We define the following failure events.

$$F_k^r = \left\{ \exists s, a, h : \ |r_h(s,a) - \hat{r}_h^k(s,a)| > \sqrt{\frac{2\log\frac{2SAHK}{\delta'}}{n_h^k(s,a) \vee 1}} \right\}$$

$$F_k^p = \left\{ \exists s, a, h : \ \left\| P_h(\cdot \mid s,a) - \hat{P}_h^k(\cdot \mid s,a) \right\|_1 > \sqrt{\frac{4S\log\frac{3SAHK}{\delta'}}{n_h^k(s,a) \vee 1}} \right\}$$

$$F^n = \left\{ \sum_{k=1}^K \mathbb{E}\left[\sum_{h=1}^H \frac{1}{\sqrt{n_h^k(s_h^k, a_h^k) \vee 1}} \mid \mathcal{F}_{k-1} \right] > 16H^2 \log\left(\frac{1}{\delta'}\right) + 4SAH^2 + 2\sqrt{2}\sqrt{SAH^2K\log HK} \right\}$$

$$F_k^c = \left\{ \exists s, a, h : \ \left| c_h(s,a) - \hat{c}_h^k(s,a) \right| > 24\sqrt{\kappa SAH^{2.5}}(L+1)^{1.5}\log^2\left(\frac{4}{\delta'}\left(1 + \frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right) \frac{1}{\sqrt{n_h^k(s,a) + 4\frac{SAH}{L\sqrt{H}+0.5}}} \right\}$$

Furthermore, the following relations hold.

- Let $F^r = \bigcup_{k=1}^K F_k^r$. Then $\Pr\{F^r\} \le \delta'$, by Hoeffding's inequality, and using a union bound argument on all s, a, and all possible values of $n_k(s,a)$ and k. Furthermore, for $n(s,a) = 0$ the bound holds trivially since $r \in [0,1]$.
- Let $F^P = \bigcup_{k=1}^K F_k^p$. Then $\Pr\{F^p\} \le \delta'$, holds by [Weissman et al., 2003] while applying union bound on all s, a, and all possible values of $n_h^k(s,a)$ and k. Furthermore, for $n(s,a) = 0$ the bound holds trivially.
- $\Pr\{F^n\} \le \delta'$ by Lemma 9.
- Let $F^c = \bigcup_{k=1}^K F_k^c$. Then $\Pr\{F^c\} \le \delta'$ for all $k \ge 1$ by Lemma 6.

We define the good event as the event where all failure events do not occur for all $k \in [K]$, namely, $\mathcal{G} = \neg F^r \bigcap \neg F^p \bigcap \neg F^n \bigcap \neg F^c$. Then, the following holds:

Lemma 2 (Good event)**.** *Setting $\delta' = \frac{\delta}{4}$ then $\Pr\{F^r \bigcup F^p \bigcup F^n \bigcup F^c\} \le \delta$. When the failure events does not hold we say the algorithm is outside the failure event, or inside the good event $\mathcal{G}$.*

H.1 Optimism

Following the events in $\mathcal{G}$ (Lemma 2) we define the following bonuses.

$$b_k^r(h,s,a) = \sqrt{\frac{2\log\frac{8SAHK}{\delta}}{n_h^k(s,a) \vee 1}}$$

$$b_k^p(h,s,a) = H\sqrt{\frac{4S\log\frac{12SAHK}{\delta}}{n_h^k(s,a) \vee 1}}$$

$$b_k^c(h,s,a) = 24\sqrt{\kappa SAH^{2.5}}(L+1)^{1.5}\log^2\left(\frac{16}{\delta}\left(1 + \frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right) \frac{1}{\sqrt{n_h^k(s,a) + 4\frac{SAH}{L\sqrt{H}+0.5}}} \ .$$

The total reward bonus is defined as $b_k^{rp}(h,s,a) = b_k^r(h,s,a) + b_k^p(h,s,a)$. Adding the reward bonus and subtracting the cost bonus leads to an optimistic MDP, as we prove in the following lemma.

Lemma 3 (Optimism)**.** *Let $\bar{V}^k$ be the optimal value of the optimistic MDP $\overline{\mathcal{M}}_T\big(\mathcal{S}, \mathcal{A}, H, \bar{r}^k, \hat{P}^k, \bar{c}^k\big)$, clipped by H, i.e., for all $h \in [H], s \in \mathcal{S}$*

$$\bar{V}_h^k(s, \tau_{1:h}) = \min\Big\{V^*_{\overline{\mathcal{M}}_T}(s, \tau_{1:h}), H\Big\}.$$

Then, under the good event $\mathcal{G}$, for any $k \in [K]$, $s \in [S]$, history $\tau_{1:h}$ and $h \in [H]$, it holds that $\bar{V}_h^k(s, \tau_{1:h}) \geq V^*(s, \tau_{1:h})$

Proof. We prove the claim by an induction over h.

First, the induction trivially holds for $h = H+1$, since $\bar{V}_h^k(s, \tau_{1:h}) = V^*(s, \tau_{1:h}) = 0$.

Now, let $h \in [H]$ and assume that the claim holds for $h' = h+1, \forall s \in \mathcal{S}$ and $\tau_{1:h'}$.

Denote:

$$a^* \in \arg\max_a \big\{r_h(s,a) + T_h(\cdot \mid s, a, \tau_{1:h})^T V_{h+1}^*(\cdot, \tau_{1:h} \cup (s,a))\big\}.$$

Recall that $\tau_{1:h} = (s_{h-1}, a_{h-1}, \ldots, s_1, a_1)$ is the history up to time h, and for brevity, denote $\tau^*_{1:h+1} \triangleq (s, a^*) \cup \tau_{1:h} = (s, a^*, s_{h-1}, a_{h-1}, \ldots, s_1, a_1)$, the history up to time $h+1$ when visiting s and playing a^* on the h^{th} timestep. Also, with some abuse of notation, we let $[TV](s,a,\tau) = T(\cdot|s,a,\tau)^T V(\cdot, (s,a) \cup \tau)$ and similarly define $[PV](s,a,\tau) = P(\cdot|s,a)^T V(\cdot, (s,a) \cup \tau)$. Finally, denote by d^τ the vector with elements $d^\tau(h,s,a) = 1$ if $(s_h, a_h) \in \tau$ and 0 otherwise. Then,

$$\begin{aligned}
&\bar{V}_h^k(s, \tau_{1:h}) - V_h^*(s, \tau_{1:h}) \\
&\quad = \max_a \Big\{\bar{r}_h^k(s,a) + \Big[T_h^{\hat{P}_k, \bar{c}_k} \bar{V}_{h+1}^k\Big](s, a, \tau_{1:h})\Big\} - \max_a \big\{r_h(s,a) + \big[T_h V_{h+1}^*\big](s,a,\tau_{1:h})\big\} \\
&\quad \overset{(1)}{\geq} \bar{r}_h^k(s,a^*) + \Big[T_h^{\hat{P}_k, \bar{c}_k} \bar{V}_{h+1}^k\Big](s, a^*, \tau_{1:h}) - r_h(s,a^*) - \big[T_h V_{h+1}^*\big](s,a^*,\tau_{1:h}) \\
&\quad \overset{(2)}{\geq} \bar{r}_h^k(s,a^*) + \Big[T_h^{\hat{P}_k, \bar{c}_k} V_{h+1}^*\Big](s, a^*, \tau_{1:h}) - r_h(s,a^*) - \big[T_h V_{h+1}^*\big](s,a^*,\tau_{1:h}) \\
&\quad = \bar{r}_h^k(s,a^*) - r_h(s,a^*) + \Big[\Big(T_h^{\hat{P}_k, \bar{c}_k} - T_h\Big) V_{h+1}^*\Big](\tau^*_{1:h+1}) \\
&\quad = \bar{r}_h^k(s,a^*) - r_h(s,a^*) + \Big[\Big(T_h^{\hat{P}_k, \bar{c}_k} - T_h^{P, \bar{c}_k}\Big) V_{h+1}^*\Big](\tau^*_{1:h+1}) + \Big[\Big(T_h^{P, \bar{c}_k} - T_h\Big) V_{h+1}^*\Big](\tau^*_{1:h+1}) \\
&\quad \overset{(3)}{=} \bar{r}_h^k(s,a^*) - r_h(s,a^*) \\
&\qquad + \Big(1 - \rho\big(\big\langle d^{\tau^*_{h+1}}, \bar{c}\big\rangle\big)\Big)\Big[\Big(\hat{P}_h - P\Big) V_{h+1}^*\Big](\tau^*_{1:h+1}) \\
&\qquad + \Big(\rho\big(\big\langle d^{\tau^*_{h+1}}, c\big\rangle\big) - \rho\big(\big\langle d^{\tau^*_{h+1}}, \bar{c}\big\rangle\big)\Big)\big[P V_{h+1}^*\big](\tau^*_{1:h+1}) \\
&\quad \overset{(4)}{=} \underbrace{\hat{r}_h^k(s,a^*) - r_h(s,a^*) + b_k^r(h,s,a^*)}_{(a)} \\
&\qquad + \underbrace{\Big(1 - \rho\big(\big\langle d^{\tau^*_{h+1}}, \bar{c}\big\rangle\big)\Big)\Big[\Big(\hat{P}_h - P\Big) V_{h+1}^*\Big](\tau^*_{1:h+1}) + b_k^p(h,s,a^*)}_{(b)} \\
&\qquad + \underbrace{\Big(\rho\big(\big\langle d^{\tau^*_{h+1}}, c\big\rangle\big) - \rho\big(\big\langle d^{\tau^*_{h+1}}, \hat{c} - b_k^c\big\rangle\big)\Big)\big[P V_{h+1}^*\big](\tau^*_{1:h+1})}_{(c)}
\end{aligned}$$

Relation (1) is is due to the definition of the $\max$-operator and the fact that a^* is the optimal action in the true MDP. Next, (2) is due to the induction step that $\bar{V}_h^k \geq V_h^*$. (3) is by the TerMDP model assumption (and since the value at termination is 0), and (4) is by the definitions of $\bar{r}_h^k(s,a)$ and $\bar{c}$.

We now turn to bound each of the three terms under the good event.

Term (a): Reward Optimism.

$$(a) = \hat{r}_h^k(s,a^*) - r_h(s,a^*) + b_k^r(h,s,a^*) \geq -b_k^r(h,s,a^*) + b_k^r(h,s,a^*) = 0,$$

where the first transition holds under the good event, and specifically, event $\neg F^r$).

Term (b): Transition Optimism.

$$\begin{aligned}
(b) &= \Big(1 - \rho\big(\langle d^{\tau^*_{h+1}}, \bar{c}\rangle\big)\Big)\Big[\Big(\hat{P}_h - P\Big)V^*_{h+1}\Big](\tau^*_{1:h+1}) + b_k^p(h,s,a^*) \\
&\geq -\Big|1 - \rho\big(\langle d^{\tau^*_{h+1}}, \bar{c}\rangle\big)\Big| \Big\|\hat{P}_h(\cdot \mid s,a^*) - P(\cdot \mid s,a^*)\Big\|_1 \big\|V^*_{h+1}(\cdot, \tau^*_{1:h+1})\big\|_\infty + b_k^p(h,s,a^*) \\
&\geq -H \Big\|\hat{P}_h(\cdot \mid s,a^*) - P(\cdot \mid s,a^*)\Big\|_1 + b_k^p(h,s,a^*) \\
&\geq -b_k^p(h,s,a^*) + b_k^p(h,s,a^*) \\
&= 0.
\end{aligned}$$

The first transition is by Hölder's inequality. The second is by the fact $\forall x,\ \rho(x) \in [0,1]$ and $\big\|V^*_{h+1}(\cdot, \tau^*_{1:h+1})\big\|_\infty \leq H$. The third transition is by under the good event, and specifically, event $\neg F^p$.

Term (c): Termination Cost Optimism. First, notice that under the good event (and specifically, event $\neg F^c$, see Lemma 6) and by the definition of the bonus b_k^c, it holds for any h,s,a that $\hat{c}_h(s,a) - b_k^c(h,s,a) \leq c_h(s,a)$. Therefore, as for any h,s,a and τ, $d^\tau(h,s,a) \geq 0$, it also holds that $\Big\langle d^{\tau^*_{h+1}}, \hat{c} - b_k^c \Big\rangle \leq \Big\langle d^{\tau^*_{h+1}}, c \Big\rangle$. Finally, by the monotonicity of ρ and the non-negativity of $\big[PV^*_{h+1}\big](\tau^*_{1:h+1})$, we have that

$$\begin{aligned}
(c) &= \Big(\rho\big(\langle d^{\tau^*_{h+1}}, c\rangle\big) - \rho\big(\langle d^{\tau^*_{h+1}}, \hat{c} - b_k^c\rangle\big)\Big)\big[PV^*_{h+1}\big](\tau^*_{1:h+1}) \\
&\geq \Big(\rho\big(\langle d^{\tau^*_{h+1}}, c\rangle\big) - \rho\big(\langle d^{\tau^*_{h+1}}, c\rangle\big)\Big)\big[PV^*_{h+1}\big](\tau^*_{1:h+1}) \\
&= 0
\end{aligned}$$

By plugging in the bounds for each of the above terms we get that

$$\bar{V}_h^k(s,\tau_{1:h}) - V_h^*(s,\tau_{1:h}) \geq 0,$$

which concludes the proof by the induction hypothesis. □

I Regret Analysis

We state the main result (Theorem 2) explicitly below. We note that in Theorem 2 we used the $\mathcal{O}$-notation, which assumes that $L = \mathcal{O}(SAH)$.

Theorem 3 (Regret of TermCRL)**.** *With probability at least* $1-\delta$*, the regret of Algorithm 1 is*

$$\begin{aligned}\operatorname{Reg}(K) \leq & \left(16H^2 \log\left(\frac{4}{\delta}\right) + 4SAH^2 + 2\sqrt{2}\sqrt{SAH^2K\log HK}\right) \\ & \times \Bigg(2\sqrt{2\log\frac{8SAHK}{\delta}} + 2H\sqrt{4S\log\frac{12SAHK}{\delta}} \\ & + 32H^2\sqrt{\kappa SAH}(L+1)^{1.5}\log\left(\frac{16}{\delta}\left(1+\frac{k(L+0.5)}{16S^2A^2H}\right)\right)\left(\sqrt{\frac{L+0.5}{4SAH}}\vee 1\right)\Bigg).\end{aligned}$$

Proof. Under the good event, the regret is bounded by

$$\begin{aligned}\operatorname{Reg}(K) &= \sum_{k=1}^{K} V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k) \\ &\overset{(1)}{\leq} \sum_{k=1}^{K} \bar{V}_1^k(s_1^k) - V_1^{\pi^k}(s_1^k) \\ &\overset{(2)}{=} \sum_{k=1}^{K}\sum_{h=1}^{H} \mathbb{E}\Big[(\bar{r}_h^k - r)(s_h^k,a_h^k) + (T_h^{\hat{P}_k,\bar{c}_k} - T_h)(\cdot|s_h^k,a_h^k,\tau_{1:h}^k)^T \bar{V}_{h+1}^{\pi^k}(\cdot,\tau_{1:h+1}^k) \;\Big|\; \mathcal{F}_{k-1}\Big] \\ &\overset{(3)}{\leq} \sum_{k=1}^{K}\sum_{h=1}^{H} \mathbb{E}\big[(\bar{r}_h^k - r)(s_h^k,a_h^k)|\mathcal{F}_{k-1}\big] + \sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\Big[\Big\|(T_h^{\hat{P}_k,\bar{c}_k} - T_h)(\cdot|s_h^k,a_h^k,\tau_{1:h}^k)\Big\|_1 \Big\|\bar{V}_{h+1}^{\pi^k}(\cdot,\tau_{1:h+1}^k)\Big\|_\infty \;\Big|\; \mathcal{F}_{k-1}\Big] \\ &\leq \underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H} \mathbb{E}\big[(\bar{r}_h^k - r)(s_h^k,a_h^k) \;\big|\; \mathcal{F}_{k-1}\big]}_{(a)} \\ &\quad \underbrace{+ H\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\Big[\Big\|(T_h^{\hat{P}_k,\bar{c}_k} - T_h^{P,\bar{c}_k})(\cdot|s_h^k,a_h^k,\tau_{1:h}^k)\Big\|_1 \;\Big|\; \mathcal{F}_{k-1}\Big]}_{(b)} \\ &\quad \underbrace{+ H\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\Big[\Big\|(T_h^{P,\bar{c}_k} - T_h)(\cdot|s_h^k,a_h^k,\tau_{1:h}^k)\Big\|_1 \;\Big|\; \mathcal{F}_{k-1}\Big]}_{(c)}.\end{aligned}$$

(1) is due to the optimism of the value function (see Lemma 3) and (2) is by the value difference lemma (Lemma 8). Notice that to use this lemma, we extended the state space to include the previously visited trajectory. (3) is due to Hölder's inequality.

Term (a): Reward Concentration. Under the Good event (see Lemma 2), and by the definition of the bonus terms b_k^r we have that

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\big[(\bar{r}_h^{k-1} - r)(s_h^k,a_h^k) \;\big|\; \mathcal{F}_{k-1}\big] \quad \leq \sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\big[2b_k^r(h,s_h^k,a_h^k) + b_k^p(h,s_h^k,a_h^k) \;\big|\; \mathcal{F}_{k-1}\big].$$

Term (b): Transition Concentration. Recall that

$$T_h^{P,c}(\cdot|s_h^k,a_h^k,\tau_{1:h}^k) = \rho\big(\langle\hat{d}_h^k,c\rangle\big)P(\cdot|s_h^k,a_h^k),$$

and since $\rho \in (0,1)$ we have that

$$\sum_{k=1}^{K}\sum_{h=1}^{H} H\mathbb{E}\Big[\Big\|(T_h^{\hat{P}_k,\bar{c}_k} - T_h^{P,\bar{c}_k})(\cdot|s_h^k, a_h^k, \tau_{1:h}^k)\Big\|_1 |\mathcal{F}_{k-1}\Big]$$

$$\le \sum_{k=1}^{K}\sum_{h=1}^{H} H\mathbb{E}\Big[\Big\|(\hat{P}_k - P)(\cdot|s_h^k, a_h^k)\Big\|_1 |\mathcal{F}_{k-1}\Big]$$

$$\le \sum_{k=1}^{K}\sum_{h=1}^{H} \mathbb{E}\big[b_k^p(h, s_h^k, a_h^k)|\mathcal{F}_{k-1}\big],$$

where the last transition is by the good event (Lemma 2) and the definition of b_k^p.

Term (c): Termination Cost Concentration.

$$H\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\Big[\Big\|(T_h^{P,\bar{c}_k} - T_h)(\cdot|s_h^k, a_h^k, \tau_{1:h}^k)\Big\|_1 \Big| \mathcal{F}_{k-1}\Big]$$

$$= H\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\left[\Big|\rho\big(\langle \hat{d}_h^k, \bar{c}^k\rangle\big) - \rho\big(\langle \hat{d}_h^k, c\rangle\big)\Big| \underbrace{\|P_h(\cdot|s_h^k, a_h^k)\|_1}_{=1} + \Big|\Big(1-\rho\big(\langle \hat{d}_h^k, \bar{c}^k\rangle\big)\Big) - \Big(1-\rho\big(\langle \hat{d}_h^k, c\rangle\big)\Big)\Big| |\mathcal{F}_{k-1}\right]$$

$$= 2H\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\Big[\Big|\rho\big(\langle \hat{d}_h^k, \bar{c}^k\rangle\big) - \rho\big(\langle \hat{d}_h^k, c\rangle\big)\Big| \Big| \mathcal{F}_{k-1}\Big]$$

$$\overset{(i)}{\le} 2H\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\Big[\Big|\Big\langle \hat{d}_h^k, \bar{c}^k - c\Big\rangle\Big| \Big| \mathcal{F}_{k-1}\Big]$$

$$\overset{(ii)}{\le} 2H\sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{t=1}^{h}\mathbb{E}\big[\,|\bar{c}_t^k(s_t^k, a_t^k) - c_t(s_t^k, a_t^k)| \;|\; \mathcal{F}_{k-1}\big]$$

$$\overset{(iii)}{\le} 4H\sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{t=1}^{h}\mathbb{E}\big[\, b_k^c(t, s_t^k, a_t^k) \;|\; \mathcal{F}_{k-1}\big]$$

$$\le 4H^2\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\big[\, b_k^c(h, s_h^k, a_h^k) \;|\; \mathcal{F}_{k-1}\big],$$

(i) is by the fact $\rho(\cdot)$ is 1-Lipschitz. (ii) is by the fact $\forall h, s, a\ \hat{d}_h^k \in \{0,1\}$. (iii) is by the definition of $\bar{c}$ and under the good event by Lemma 6.

Combining all the above terms, we get

$$\mathrm{Reg}(K) \le 4\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\big[b_k^r(h, s_h^k, a_h^k) + b_k^p(h, s_h^k, a_h^k) + H^2 b_k^c(h, s_h^k, a_h^k)|\mathcal{F}_{k-1}\big].$$

Now, plugging in the bonus terms and bounding,

$$b_k^c(h,s,a) = 24\sqrt{\kappa SAH^{2.5}}(L+1)^{1.5}\log^2\left(\frac{16}{\delta}\left(1+\frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right)\frac{1}{\sqrt{n_h^k(s,a) + 4\frac{SAH}{L\sqrt{H}+0.5}}}$$

$$\le 24\sqrt{\kappa SAH^{2.5}}(L+1)^{1.5}\log^2\left(\frac{16}{\delta}\left(1+\frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right)\frac{\left(\sqrt{\frac{L+0.5}{4SA\sqrt{H}}}\vee 1\right)}{\sqrt{n_h^k(s,a)\vee 1}}\ ,$$

we get

$$
\begin{aligned}
\operatorname{Reg}(K) \leq & \, 2\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\left[\sqrt{\frac{2\log\frac{8SAHK}{\delta}}{n_h^k(s_k,a_k)\vee 1}}+H\sqrt{\frac{4S\log\frac{12SAHK}{\delta}}{n_h^k(s_k,a_k)\vee 1}}\,\middle|\,\mathcal{F}_{k-1}\right] \\
& + 4H^2\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}\left[24\sqrt{\kappa SAH^{2.5}}(L+1)^{1.5}\log^2\left(\frac{16}{\delta}\left(1+\frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right)\frac{\left(\sqrt{\frac{L+0.5}{4SA\sqrt{H}}}\vee 1\right)}{\sqrt{n_h^k(s,a)\vee 1}}\,\middle|\,\mathcal{F}_{k-1}\right] \\
\leq & \left(16H^2\log\left(\frac{4}{\delta}\right)+4SAH^2+2\sqrt{2}\sqrt{SAH^2K\log HK}\right)\times \\
& \left(2\sqrt{2\log\frac{8SAHK}{\delta}}+2H\sqrt{4S\log\frac{12SAHK}{\delta}}\right. \\
& \qquad \left.+92H^2\sqrt{\kappa SAH^{2.5}}(L+1)^{1.5}\log^2\left(\frac{16}{\delta}\left(1+\frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right)\left(\sqrt{\frac{L+0.5}{4SA\sqrt{H}}}\vee 1\right)\right) \\
= & \, \mathcal{O}\left(\sqrt{\kappa S^2A^2H^{8.5}L^3K\log^3\left(\frac{SAHK}{\delta}\right)}\right),
\end{aligned}
$$

where the last inequality is by Lemma 9. □

J Cost Concentration

In the section to follow, we provide a local concentration bound for estimating the costs in the termination model. This local concentration result will allow us to use the cost bonus defined in Appendix H, which is crucial for achieving tight regret guarantees, when the costs are unknown. Specifically, this concentration result serves as the base for dealing with the cost optimism in Lemma 3 and cost concentration in Theorem 2.

We first start by describing the optimization procedure used for learning the costs (Appendix J.1). Then, by formulating this procedure as an instance of the logistic bandits problem, we obtain a concentration result for the cost, which *globally bounds the distance between the empirical and cost vectors* (Appendix J.2). Finally, we use this global concentration result together with the structure of the termination MDP to provide a local concentration bound for the costs which *holds independently for any* h, s, a (Appendix J.3).

Before diving into the details, we first present additional notations required to the analysis in this appendix. First, to address time steps after termination, we define $\hat{d}_h^k(t, s, a) = 0$ for any $t > t_k^*$ and $\forall t, s, a$. This allows to add these vectors to the regression problem without affecting its solution. Moreover, we denote the episode-wise Gram matrix by $A_k \triangleq \sum_{h=1}^{t^*} \hat{d}_h^k {\hat{d}_h^k}^T$ and the regularized Gram matrix up to episode k by $V_k \triangleq \lambda I + \sum_{k'=1}^{k-1} A_{k'}$, for some $\lambda > 0$.

J.1 Optimization Procedure

In a vanilla MDP, at each time step, the agent takes an action and transitions to the next state. Instead, in the termination model considered in this paper, the agent potentially gets a termination signal from the environment. Importantly, the agent can gain information about the termination probabilities even in time steps when no termination occurs. Formally, at any time step up to the time of termination $h \in [H-1]$ (the last time step always ends in termination), the agent acquires a sample from a Bernoulli random variable, where 1 means termination ($h = t_k^*$), and 0 otherwise ($h \neq t_k^*$). Since the termination probability is a logistic function of the occupancy vector, finding the termination costs can be done using a logistic regression. This leads us to solve the following optimization problem in Algorithm 1 (Line 10),

$$\hat{c}_{k+1} \in \arg\max_{c \in \mathcal{C}} \sum_{k=1}^{K} \sum_{h=1}^{H-1} \Big[\mathbb{1}\{h < t_k^*\} \log\Big(1 - \rho\big(\langle \hat{d}_h^k, c\rangle\big)\Big) + \mathbb{1}\{h = t_k^*\} \log\Big(\rho\Big(\langle \hat{d}_h^k, c\rangle\Big)\Big)\Big] - \lambda \left\|c\right\|_2^2, \tag{3}$$

where $\mathcal{C}$ is the set of possible costs. Specifically, by the fact we only observe termination on the trajectories we played, the feedback in this problem is bandit feedback. In the next section, we use known results for the logistic bandit problem Abeille et al. [2021], and apply them to acquire a global concentration bound for the costs learned in this procedure.

J.2 Global Cost Concentration

We start by stating concentration results for logistic bandits. Let $x_t \in \mathcal{X} \subset \mathbb{R}^d$ be a sequence of contexts s.t. $\|x\| \leq 1$ for all $x \in \mathcal{X}$ and assume that y_t are Bernoulli random variable such that $\mathbb{E}[y_t | x_t] = \rho(x_t^T \theta_*)$, for some $\theta^* \in \Theta$. Next, define the regularized logistic loss w.r.t. $\lambda > 0$ by $\mathcal{L}_t(\theta) = -\sum_{s=1}^{t-1} \big(y_s \log \rho(x_s^T \theta) + (1 - y_s) \log(1 - \rho(x_s^T \theta))\big) + \lambda \left\|\theta\right\|_2^2$. We denote the unconstrained minimizer of the loss by $\bar{\theta}_t$ and its minimizer over Θ by $\hat{\theta}_t$. Finally, with slight abuse of notation, let $H_t(\theta) = \sum_{s=1}^{t-1} \dot{\mu}(x_s^T \theta) x_s x_s^T + \lambda I$.

We now focus on the confidence set $\mathcal{E}_t \delta)$, defined in [Abeille et al., 2021] as

$$\mathcal{E}_t(\delta) = \left\{\theta \in \Theta | \mathcal{L}_t(\theta) - \mathcal{L}_t(\bar{\theta}_t) \leq \beta_t(\delta)^2\right\}$$

for some appropriate $\beta_t(\delta) > 0$. By Proposition 1 and Lemma 1 of [Abeille et al., 2021], it holds that $\theta^* \in \mathcal{E}_t(\delta)$. Thus, under this event, the set is not empty. In particular, when the set is not empty, it must also contains $\hat{\theta}_t$, as the constrained minimizer of the cost. Combining this conclusion with Lemma 1 of [Abeille et al., 2021], we obtain the following lemma.

Lemma 4 (Logistic Regression Concentration)**.** *With probability of at least* $1-\delta$*, for all* $t \geq 1$*,*

$$\left\|\hat{\theta}_t - \theta_*\right\|_{H_t(\theta_*)} \leq (2+2L)\gamma_t(\delta) + 2\sqrt{1+L}\left(\gamma_t(\delta) + \frac{\gamma_t(\delta)^2}{\sqrt{\lambda}}\right) \quad , \tag{4}$$

where $L = \max_{\theta\in\Theta} \|\theta\|_2$ *and* $\gamma_t(\delta) = \sqrt{\lambda}\big(L+\frac{1}{2}\big) + \frac{d}{\sqrt{\lambda}}\log\big(\frac{4}{\delta}\big(1+\frac{t}{16d\lambda}\big)\big)$.

This result can be applied to our algorithm as follows:

Corollary 1 (Global Cost Bound)**.** *Let* $\lambda = \frac{d}{L\sqrt{H}+0.5}$, $\mathcal{X} = \big\{x \in \mathbb{R}^{SAH} : x_i \in \{0,1\}, \sum_i x_i \leq H\big\}$ *and* $\kappa = \big(\min_{x\in\mathcal{X}} \dot{\mu}(x^T c)\big)^{-1}$*. With probability of at least* $1-\delta$*, it holds that*

$$\left\|\hat{c}^k - c\right\|_{V_k} \leq 12\sqrt{\kappa SAH^{2.5}}(L+1)^{1.5}\log^2\left(\frac{4}{\delta}\left(1+\frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right).$$

Proof. We apply Lemma 4 with $x_{H(k-1)+h} = \hat{d}^k_h$ if $h \leq \min(t^*_k, H-1)$ and $x_{H(k-1)+h} = 0$ otherwise, where $k > 0$ is the episode and $h \in [H]$ is the step within the episode. Notice that zero contexts $x_{H(k-1)+h} = 0$ do not affect the regression solution, so steps after termination are ignored. Also, in our case, we have $\|x\|^2 \leq H$, instead of $\|x\| \leq 1$, so to compensate it, the parameter L will be scaled by $\sqrt{H}$. Under our assumptions, we have $d = SAH$, and thus $\gamma_k(\delta) \leq 2\sqrt{SAH(L\sqrt{H}+0.5)}\log\Big(\frac{4}{\delta}\Big(1+\frac{k(L\sqrt{H}+0.5)}{16S^2A^2H}\Big)\Big)$ and

$$\begin{aligned}
\left\|\hat{c}^k - c\right\|_{H_t(c)} &\leq (2+2L\sqrt{H})\gamma_t(\delta) + 2\sqrt{1+L\sqrt{H}}\left(\gamma_t(\delta) + \frac{\gamma_t(\delta)^2}{\sqrt{\lambda}}\right) \\
&\leq 4(1+L\sqrt{H})\gamma_t(\delta) + \frac{\sqrt{1+L\sqrt{H}}}{\sqrt{\lambda}}\gamma_t(\delta)^2 \\
&\leq 12\sqrt{SAH}(L\sqrt{H}+1)^{1.5}\log^2\left(\frac{4}{\delta}\left(1+\frac{k(L\sqrt{H}+0.5)}{16S^2A^2H}\right)\right) \\
&\leq 12\sqrt{SAH^{2.5}}(L+1)^{1.5}\log^2\left(\frac{4}{\delta}\left(1+\frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right)
\end{aligned}$$

We conclude the proof by using the fact that $\kappa H_k(c) \succeq V_t$ (Lemma 5) for $\theta^* = c$. □

Lemma 5 (Connection between Local and Global Design Matrices)**.** *For any* θ*, if* $\kappa_{\mathcal{X}}(\theta) = \frac{1}{\min_{x\in\mathcal{X}}\dot{\mu}(x^T\theta)}$*, then it holds that*

$$\kappa_{\mathcal{X}}(\theta)H_t(\theta) \succeq V_t \quad .$$

Proof. For any θ,

$$\begin{aligned}
H_t(\theta) &= \sum_{s=1}^{t-1}\dot{\mu}(x_s^T\theta)x_s x_s^T + \lambda_t I \\
&\succeq \min_{x\in\mathcal{X}}\dot{\mu}(x_s^T\theta)\sum_{s=1}^{t-1}x_s x_s^T + \lambda_t I \\
&= \frac{1}{\kappa_{\mathcal{X}}(\theta)}\sum_{s=1}^{t-1}x_s x_s^T + \lambda_t I \\
&\overset{1<\kappa_{\mathcal{X}}(\theta)}{\succeq} \frac{1}{\kappa_{\mathcal{X}}(\theta)}\left(\sum_{s=1}^{t-1}x_s x_s^T + \lambda_t I\right) = \frac{1}{\kappa_{\mathcal{X}}(\theta)}V_t \quad .
\end{aligned}$$

□

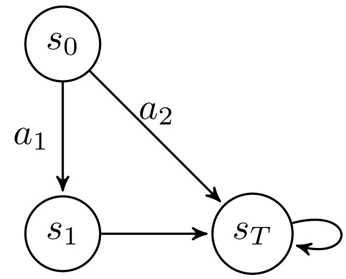

Figure 7: Example for the necessity of step-wise feedback for local estimation. Plot depicts a deterministic MDP which always begins at state s_0. Assume that we only receive a binary signal of whether there was termination at the end of the episode, but do not observe it during the episode. If we only observe the trajectory $s_0 \to s_1 \to s_T$, identifying the state in which the termination occurred is not possible, until we further observe the trajectory $s_0 \to s_T$. That is, there is no way to know whether termination occurred due to state s_0 or s_1. In contrast, in the TerMDP model we view termination signals at every iteration, allowing us to identify the costs locally w.r.t. every state and action.

J.3 From Global to Local Cost Concentration

In our setting, the global cost concentration also implies a component-wise (local) concentration of the costs. This is in stark contrast to standard regression results, where specific coordinates cannot always be recovered. To see the intuition behind it, assume for simplicity that we observe the exact termination probability (rather than just a random termination signal). Notably, on each time step up to the termination, we could directly reconstruct the partial sums $\sum_{t=1}^{h} c_t(s_t^k, a_t^k)$ by inverting the logistic function. Then, the costs of all visited states could be directly calculated through the difference between the cumulative costs of any two consecutive steps. Notice that we can calculate this difference only since we get feedback on *every* time step up to termination. Were we only to observe feedback on the cumulative costs at the termination time, then we would not be able to guarantee to reconstruct anything but the global cumulative costs we were given (see example in Figure 7). In this case, the global (vector-wise) concentration used in logistic bandits (see Corollary 1 in Appendix J.2) cannot not be improved.

Lemma 6 (Local Cost Estimation Confidence Bound)**.** *For any $\delta > 0$, with probability of at least $1-\delta$, for any episode $k \in [K]$, it holds that for all $h \in [H-1], s \in \mathcal{S}, a \in \mathcal{A}$,*

$$\left|\hat{c}_h^k(s,a) - c_h(s,a)\right| \le 24\sqrt{\kappa SAH^{2.5}}(L+1)^{1.5}\log\left(\frac{4}{\delta}\left(1+\frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right)\frac{1}{\sqrt{n_h^k(s,a)+4\frac{SAH}{L\sqrt{H}+0.5}}}.$$

Proof. For any $k \in [K]$, $h \in [H-1]$, $s \in \mathcal{S}$ and $h \in \mathcal{S}$, we can bound

$$\left|c_h^k(s,a) - c_h(s,a)\right| = \left|\langle e_{h,s,a}, \hat{c}_k - c\rangle\right| \le \|e_{h,s,a}\|_{V_k^{-1}} \|\hat{c}_k - c\|_{V_k}, \tag{5}$$

where $e_{h,s,a} \in \mathbb{R}^{HSA}$ is a unit vector in the (h,s,a) coordinate, and the inequality is due to Cauchy-Schwartz. We now turn to bound $\|e_{h,s,a}\|_{V_k^{-1}}$.

$$\begin{aligned}
V_k^{-1} &\preceq \left(\lambda I + \sum_{k'=1}^{k-1} A_{k'} \mathbb{1}\left\{(h,s,a) \in \tau^{k'}\right\}\right)^{-1} \\
&= \left(\sum_{k'=1}^{k-1}\left(\frac{\lambda}{n_h^{k-1}(s,a)} I + A_{k'}\right)\mathbb{1}\left\{(h,s,a) \in \tau^{k'}\right\}\right)^{-1} \\
&\preceq \frac{1}{\left(n_h^{k-1}(s,a)\right)^2} \sum_{k'=1}^{k-1}\left(\frac{\lambda}{n_h^{k-1}(s,a)} I + A_{k'}\right)^{-1} \mathbb{1}\left\{(h,s,a) \in \tau^{k'}\right\},
\end{aligned}$$

where $n_h^{k-1}(s,a) = \sum_{k'=1}^{k-1} \mathbb{1}\left\{(h,s,a) \in \tau^{k'}\right\}$ and the third transition is due to HM-AM inequality for positive matrices Bhagwat and Subramanian [1978].

Using this relation, we can bound $\|e_{h,s,a}\|^2_{V_k^{-1}}$ by bounding the maximal eigenvalue of each of the summands. We do so in Lemma 7, and obtain the following bound:

$$\begin{aligned}
\|e_{h,s,a}\|^2_{V_k^{-1}} &= e^T_{h,s,a} V_k^{-1} e_{h,s,a} \\
&\leq \frac{1}{\left(n_h^{k-1}(s,a)\right)^2} \sum_{k'=1}^{k-1} e^T_{h,s,a} \left(\frac{\lambda}{n_h^{k-1}(s,a)} I + A_{k'} \right)^{-1} \mathbb{1}\left\{(h,s,a) \in \tau^{k'}\right\} e_{h,s,a} \\
&\leq \frac{1}{\left(n_h^{k-1}(s,a)\right)^2} \sum_{k'=1}^{k-1} \frac{1}{\frac{1}{4} + \frac{\lambda}{n_h^{k-1}(s,a)}} \mathbb{1}\left\{(h,s,a) \in \tau^{k'}\right\} \qquad \text{(Lemma 7)} \\
&= \frac{n_h^{k-1}(s,a)}{\left(n_h^{k-1}(s,a)\right)^2} \frac{1}{\frac{1}{4} + \frac{\lambda}{n_h^{k-1}(s,a)}} \\
&= \frac{4}{n_h^{k-1}(s,a) + 4\lambda}.
\end{aligned}$$

By plugging into eq. (5), we obtain that for any k and any h, s, a

$$\left|c_h^k(s,a) - c_h(s,a)\right| \leq \|\hat{c}_k - c\|_{V_k} \|e_{h,s,a}\|_{V_k^{-1}} \leq \|\hat{c}_k - c\|_{V_k} \sqrt{\frac{4}{n_h^{k-1}(s,a) + 4\lambda}}.$$

Finally, by Corollary 1 with $d = SAH$, with probability of at least $1 - \delta$ it holds that

$$\left|c_h^k(s,a) - c_h(s,a)\right| 24\sqrt{\kappa SAH^{2.5}} (L+1)^{1.5} \log\left(\frac{4}{\delta}\left(1 + \frac{k(L+0.5)}{16S^2A^2\sqrt{H}}\right)\right) \frac{1}{\sqrt{n_h^k(s,a) + 4\frac{SAH}{L\sqrt{H}+0.5}}},$$

which concludes the proof.

□

Lemma 7 (Inverse Eigenvalues Bound). *If $(h,s,a) \in \tau^k$ and $e_{h,s,a} \in \mathbb{R}^{HSA}$ is a unit vector in the coordinate (h,s,a), then $e^T_{h,s,a}(\lambda I + A_k)^{-1} e_{h,s,a} \leq \frac{1}{\frac{1}{4}+\lambda}$.*

Proof. Throughout the proof, we assume for brevity that there was termination in all episodes, i.e., $t_k^* \leq H - 1$ for all $k \in [K]$. Otherwise, the exact same proof follows by replacing t_k^* by $\min\{t_k^*, H-1\}$ (namely, treating the lack of termination feedback at the last timestep of the episodes). With some abuse of notations, we also use $e_i \in \mathbb{R}^{HSA}$ to denote the unit vector in the i-th coordinate.

To simplify the proof, it would be helpful to assume that the t-th coordinate of the empirical occupancy vector represents the state that was visited on the t-th time step (namely, coordinates are sorted by their visitation order). States that were not visited can be arbitrarily ordered. Formally, this can be done using any permutation matrix P_k such that $e_{t,s_t^k,a_t^k} = P_k e_t$ for all $t \in [t_k^*]$ (and other coordinates can be arbitrarily permuted). In particular, denoting $\bar{e}_t = \sum_{i=1}^t e_i = \big(\underbrace{1,\dots,1}_{t-\text{times}}, 0, \dots, 0\big)^T$, we can write $\hat{d}_t^k = \sum_{i=1}^t e_{i,s_i^k,a_i^k} = \sum_{i=1}^t P_k e_i = P_k \bar{e}_t$.

Then, we have that

$$\begin{aligned}
e_{h,s,a}^T(\lambda I + A_k)^{-1}e_{h,s,a} &= e_{h,s,a}^T\left(\lambda I + \sum_{t=1}^{t_k^*}\hat{d}_t^k {\hat{d}_t^k}^T\right)^{-1} e_{h,s,a} \\
&= e_{h,s,a}^T\left(\lambda I + \sum_{t=1}^{t_k^*}P_k\bar{e}_t\bar{e}_t^T P_k^T\right)^{-1} e_{h,s,a} \\
&= e_{h,s,a}^T\left(P_k\left(\lambda I + \sum_{t=1}^{t_k^*}\bar{e}_t\bar{e}_t^T\right)P_k^T\right)^{-1} e_{h,s,a} \\
&= e_{h,s,a}^T P_k\left(\lambda I + \sum_{t=1}^{t_k^*}\bar{e}_t\bar{e}_t^T\right)^{-1} P_k^T e_{h,s,a} \quad ,
\end{aligned}$$

where the two last relations is since permutation matrices are orthogonal, namely $P_k^{-1} = P_k^T$.

Now, notice that P_k permutes the first t_k^* components to the visited (t, s_t^k, a_t^k) tuples. Thus, its inverse P_k^T permutes visited tuples (t, s_t^k, a_t^k) to the t-th coordinate, namely $P_k^T e_{t,s_t^k,a_t^k} = e_t$ and

$$e_{h,s,a}^T(\lambda I + A_k)^{-1}e_{h,s,a} = e_h^T\left(\lambda I + \sum_{t=1}^{t_k^*}\bar{e}_t\bar{e}_t^T\right)^{-1} e_h \quad .$$

Moreover, $\lambda I + \sum_{t=1}^{t_k^*}\bar{e}_t\bar{e}_t^T$ is a block-diagonal matrix whose first block is located at its first t_k^* coordinates. Thus, each block can be inverted independently, and as e_h is located in the first block of the matrix, w.l.o.g. we can only focus on the $t_k^* \times t_k^*$ first-block of the matrix. We denote this block by $B \in \mathbb{R}^{t_k^* \times t_k^*}$, and if $u_h \in \mathbb{R}^{t_k^*}$ is a unit vector in the h^{th} coordinate, we can write $e_h^T\left(\lambda I + \sum_{t=1}^{t_k^*}\bar{e}_t\bar{e}_t^T\right)^{-1} e_h = u_h^T(\lambda I + B)^{-1}u_h$.

Directly calculating of the sum $\sum_{t=1}^{t_k^*}\bar{e}_t\bar{e}_t^T$, one can easily see that $B(i,j) = t_k^* + 1 - \max\{i,j\}$, as we now illustrate:

$$\begin{pmatrix}
t_k^* & t_k^*-1 & t_k^*-2 & \dots & 1 \\
t_k^*-1 & t_k^*-1 & t_k^*-2 & \dots & 1 \\
t_k^*-2 & t_k^*-2 & t_k^*-2 & \dots & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
1 & 1 & 1 & 1 & 1
\end{pmatrix}$$

This matrix can be easily diagonalized, which can be used to calculate its inverse:

$$\left(\begin{array}{ccccc|ccccc}
t_k^* & t_k^*-1 & t_k^*-2 & \dots & 1 & 1 & 0 & 0 & \dots & 0 \\
t_k^*-1 & t_k^*-1 & t_k^*-2 & \dots & 1 & 0 & 1 & 0 & \dots & 0 \\
t_k^*-2 & t_k^*-2 & t_k^*-2 & \dots & 1 & 0 & 0 & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 1
\end{array}\right)$$

$$= \left(\begin{array}{ccccc|ccccc}
1 & 0 & 0 & \dots & 0 & 1 & -1 & 0 & \dots & 0 \\
1 & 1 & 0 & \dots & 0 & 0 & 1 & -1 & \dots & 0 \\
1 & 1 & 1 & \dots & 0 & 0 & 0 & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 1
\end{array}\right)$$

$$= \left(\begin{array}{ccccc|ccccc}
1 & 0 & 0 & \dots & 0 & 1 & -1 & 0 & \dots & 0 \\
0 & 1 & 0 & \dots & 0 & -1 & 2 & -1 & \dots & 0 \\
0 & 0 & 1 & \dots & 0 & 0 & -1 & 2 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 2
\end{array}\right)$$

In the first step, we subtracted rows $i + 1$ from the i rows, and in the second step, we subtracted the $i - 1$ rows from the i rows. Thus, the inverse can be explicitly written as follows:

$$B^{-1}_{i,j} = \begin{cases} 1 & i = j = 1 \\ 2 & i = j > 1 \\ -1 & i = j - 1 \text{ or } i = j + 1 \\ 0 & \text{o.w.} \end{cases}$$

Notice that the absolute values of all rows is smaller than 4. Then (e.g., by Gershgorin circle theorem), $\lambda_{\max}(B^{-1}) \leq 4$, and since B is PSD, $\lambda_{\min}(B) \geq \frac{1}{4}$. Finally, we get the desired result by bounding

$$\begin{aligned} e^T_{h,s,a}(\lambda I + A_k)^{-1} e_{h,s,a} &= u^T_h (\lambda I + B)^{-1} u_h \\ &\leq \underbrace{\|u_h\|_2}_{=1} \lambda_{\max}\Big((\lambda I + B)^{-1}\Big) \\ &= \frac{1}{\lambda_{\min}(\lambda I + B)} \leq \frac{1}{\frac{1}{4} + \lambda} \ . \end{aligned}$$

Remark. Notice that the same conclusion still holds in the extreme case of $t^* \leq 2$: we get $\lambda_{\max}(B^{-1}) \leq 3$ for $t^* = 2$ and $\lambda_{\max}(B^{-1}) = 1$ (as B contains a single element). □

K Useful Lemmas

Lemma 8 (Value difference lemma, e.g., Dann et al. [2017], Lemma E.15)**.** *Consider two MDPs* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, H)$ *and* $\mathcal{M}' = (\mathcal{S}, \mathcal{A}, P', r', H)$*. For any policy* π *and any* s, h*, the following relation holds:*

$$V_h^\pi(s;\mathcal{M}) - V_h^\pi(s;\mathcal{M}')$$
$$= \mathbb{E}\left[\sum_{h'=h}^{H} (r_{h'}(s_{h'}, a_{h'}) - r'_{h'}(s_{h'}, a_{h'})) + (P - P')(\cdot \mid s_{h'}, a_{h'})^T V_{h'+1}^\pi(\cdot;\mathcal{M}') | s_h = s, \pi, P\right]$$

The following lemma is due to Efroni et al. [2020b], with the only exception that the original lemma assumes a stationary MDP and therefore, S translates to SH in the following.

Lemma 9 (Expected Cumulative Visitation Bound, Lemma 22, Efroni et al. [2020b])**.** *Let* $\{\mathcal{F}_k\}_{k=1}^K$ *be the natural filtration. Then, with probability greater than* $1 - \delta$ *it holds that*

$$\begin{aligned}\sum_{k=1}^{K} \mathbb{E}\left[\sum_{h=1}^{H} \frac{1}{\sqrt{n_h^k(s_h^k, a_h^k) \vee 1}} \mid \mathcal{F}_{k-1}\right] &\le 16H^2 \log\left(\frac{1}{\delta}\right) + 4SAH^2 + 2\sqrt{2}\sqrt{SAH^2K\log HK} \\ &= \mathcal{O}\left(H\left(HSA + H\log\left(\frac{1}{\delta}\right)\right) + \sqrt{SAH^2K\log HK}\right) \\ &= \tilde{\mathcal{O}}\left(\sqrt{H^2SAK}\right)\end{aligned}$$