Byzantine-Robust Online and Offline Distributed Reinforcement Learning

Yiding Chen*
University of Wisconsin-Madison

Xuezhou Zhang*
Princeton University

Kaiqing Zhang
University of Maryland, College Park

Mengdi Wang
Princeton University

Xiaojin Zhu
University of Wisconsin-Madison

Abstract

We consider a distributed reinforcement learning setting where multiple agents separately explore the environment and communicate their experiences through a central server. However, α-fraction of agents are adversarial and can report arbitrary fake information. Critically, these adversarial agents can collude and their fake data can be of any sizes. We desire to robustly identify a near-optimal policy for the underlying Markov decision process in the presence of these adversarial agents. Our main technical contribution is COW, a novel algorithm for the *robust mean estimation from batches* problem, that can handle arbitrary batch sizes. Building upon this new estimator, in the offline setting, we design a Byzantine-robust distributed pessimistic value iteration algorithm; in the online setting, we design a Byzantine-robust distributed optimistic value iteration algorithm. Both algorithms obtain near-optimal sample complexities and achieve superior robustness guarantee than prior works.

1 INTRODUCTION

Distributed learning systems have been one of the main driving forces to recent successes of deep learning (Verbraeken et al., 2020; Goyal et al., 2017; Abadi et al., 2016). Advances in designing efficient distributed optimization algorithms(Horgan et al., 2018) and deep learning infrastructures (Espeholt et al., 2018) have enabled the training of powerful models with hundreds of billions of parameters (Brown et al., 2020). However, new challenges emerge with the outsourcing of computation and data collection. In particular, distributed systems have been found vulnerable to Byzantine failure (LAMPORT et al., 1982), meaning there could be agents with failure that may send arbitrary information to the central server. Even a small number of Byzantine machines that send out moderately corrupted data can lead to a significant loss in performance (Yin et al., 2018; Ma et al., 2019; Zhang et al., 2020a), which raise security concern in real-world applications such as chatbot (Neff and Nagy, 2016) and autonomous vehicles (Eykholt et al., 2018; Ma et al., 2021). In addition, other desired properties are chased after, such as protecting the data privacy of individual data contributors (Sakuma et al., 2008; Liu et al., 2019) and reducing communication cost (Dubey and Pentland, 2021). These challenges require new algorithmic design on the server side, which is the main focus of this paper.

When it comes to reinforcement learning (RL), distributed learning has been prevalent in many large-scale decision-making problems even before the deep learning era, such as cooperative learning in robotics systems (Ding et al., 2020), power grids optimization (Yu et al., 2014) and automatic traffic control (Bazzan, 2009). Unlike supervised learning, where the data distribution of interest is often fixed prior, reinforcement learning requires active exploration on the agent's side to discover the optimal policy for the current task, thus creating new challenges in achieving the above desiderata while exploring in an unknown environment.

This paper studies this exact problem:

Can we design a distributed RL algorithm that is sample efficient and robust to Byzantine agents while having small communication costs and promoting data privacy?

We study Byzantine-robust RL in both the online and offline settings: In the online setting, a central server is designed to outsource exploration tasks to m agents iteratively, the agents collect experiences and send them back to the server, and the server uses the data to update its policy; In the offline setting, a central server collects logged data from m agents and uses the data to identify a good policy without additional

---

* Equal Contribution. Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

interaction with the environment. However, among the m agents, an α-fraction is Byzantine, meaning they can send arbitrary data in both the online and offline settings. We summarize our contributions as follows:

1. We design COW, a robust mean estimation algorithm for learning from batches. By utilizing the batch structure, the estimation error of our algorithm vanishes with more data. Compared to prior works (Qiao and Valiant, 2017; Chen et al., 2020; Jain and Orlitsky, 2021; Yin et al., 2018), our algorithm adapts to arbitrary batch sizes, which is desired in many applications of interest.
2. We design BYZAN-UCBVI, a Byzantine-Robust variant of optimistic value iteration for online RL, by calling COW as a subroutine. We show that BYZAN-UCBVI achieves near-optimal regret with α-fraction Byzantine agents. Meanwhile, BYZAN-UCBVI also enjoys a logarithmic communication cost and switching cost (Bai et al., 2019; Zhang et al., 2020b; Gao et al., 2021), and preserves data privacy of individual agents.
3. We design BYZAN-PEVI, a Byzantine-Robust variant of pessimistic value iteration for offline RL, again utilizing COW as a subroutine. Despite the presence of Byzantine agents, we show that BYZAN-PEVI can learn a near-optimal policy with a polynomial number of samples when certain coverage conditions are satisfied (Zhang et al., 2021a).

2 RELATED WORK

Reinforcement Learning: Reinforcement learning aims to find the optimal policy in a Markov Decision Process (MDP) (Sutton and Barto, 2018). Here we mainly survey prior works that introduce ideas and theoretical tools that inspire our work. (Azar et al., 2017; Dann et al., 2017) show that UCB-style algorithms achieve minimax regret bound in tabular MDPs. Recent work extends the theoretical understanding to RL with function approximation (Jin et al., 2020; Yang and Wang, 2019, 2020). Our analysis for the online RL algorithm follows the theoretical framework of *optimism in the face of uncertainty*, yet the technical steps differ significantly from the above works. (Jin et al., 2021; Rashidinejad et al., 2021) use a pessimistic strategy to efficiently learn a near-optimal policy in the offline setting. The same principle is utilized in the design of our offline RL algorithm. Recently, (Bai et al., 2019; Zhang et al., 2020b; Gao et al., 2021) study low switching-cost RL algorithm, meaning the learning agent only performs a small number of policy changes. Our algorithm borrows ideas from these works to simultaneously achieve small communication costs and statistical robustness.

Distributed Reinforcement Learning: Parallel RL deploys large-scale models in distributed system (Kretchmar, 2002). (Horgan et al., 2018; Espeholt et al., 2018) provide distributed architecture for deep reinforcement learning by parallelizing the data-generating process. (Dubey and Pentland, 2021; Agarwal et al., 2021; Chen et al., 2021) provide the first sets of theoretical guarantees for performance and communication cost in parallel RL. We take a step further to study the Byzantine-robust problem in distributed RL.

Robust Statistics: Robust statistics studies learning with corrupted datasets and has a long history (Huber, 1992; Tukey, 1960). In modern machine learning, models are high-dimensional. Recent work provides sample and computationally efficient algorithms for robust mean and covariance estimation in high dimension (Diakonikolas et al., 2016, 2017; Lai et al., 2016). Shortly after, those robust mean estimators are applied to robust supervised learning (Diakonikolas et al., 2019; Prasad et al., 2018) and RL (Zhang et al., 2021a,b). A line of work of particular interest to us studies robust learning from data batches (Qiao and Valiant, 2017; Chen et al., 2020; Jain and Orlitsky, 2021; Yin et al., 2018). They consider a setting where the data is collected from many distinct data sources, and a fraction of the data sources is corrupted. By exploiting the batch structure of the data, these algorithms can achieve significantly higher accuracy than in the non-batch setting (Diakonikolas et al., 2016). However, to our best knowledge, all of these works study batches with equal sizes, which does not often capture situations in practice. In contrast, our algorithm in Section 3 works for arbitrarily different batch sizes and achieves a near-optimal rate *adaptively*.

Byzantine-Robust Distributed Learning: Byzantine-Robust learning algorithm studies learning under Byzantine failure (LAMPORT et al., 1982). (Chen et al., 2017) provides a Byzantine gradient descent via the geometric median of mean estimation for the gradients. (Yin et al., 2018) provides robust distributed gradient descent algorithms with optimal statistics rates. These works also restrict to a setting where each worker handles the same number of gradient computations. As we will show later, their algorithm and rate will no longer be optimal when the batch sizes differ.

Corruption-Robust RL And Byzantine-Robust RL: There is a line of work studying adversarial attack against reinforcement learning (Ma et al., 2019; Zhang et al., 2020a; Huang et al., 2017), and corruption robust reinforcement RL for online (Zhang et al., 2021b; Lykouris et al., 2021) and offline (Zhang et al., 2021a) settings. (Jadbabaie et al., 2022) studies Byzantine-Robust linear bandits in the federated setting. Unlike our setting, they allow different agents to be subject to Byzantine attacks in different episodes. Our algorithm enjoys a better regret bound and communication cost. (Fan et al., 2021) provides a Byzantine-robust policy gradient algorithm that is guaranteed to converge to an approximately stationary point, whereas our algorithm guarantees to find an approximately optimal policy. (Dubey and Pentland, 2020) studies Byzantine-Robust multi-armed bandit, where the corruption can only come from a fixed distribution. We study a more difficult MDP setting and allow the corruption to be arbitrary.

3 ROBUST MEAN ESTIMATION FROM UNTRUTHFUL BATCHES

To prepare for our discussion of byzantine-robust RL, we first discuss an important subproblem called *robust mean estimation from batches*, which captures many of the unique properties and challenges byzantine-robust RL faces. Indeed, our byzantine-robust RL algorithms will crucially be built upon the algorithm we design for this preliminary problem.

Definition 3.1 (Robust mean estimation from batches)**.** *There are m data providers indexed by $\{1, 2, \ldots, m\} =: [m]$. Among these providers, we denote the indices of uncorrupted (good) providers by $\mathcal{G} \subseteq [m]$ and the indices of corrupted (bad) providers by $\mathcal{B} = [m] \backslash \mathcal{G}$, where $|\mathcal{B}| = \alpha m$. Each provider $j \in [m]$ sends a data batch $x_j^{[n_j]} := \{x_j^1, \ldots, x_j^{n_j}\}$ to the server, where the batch size n_j can be arbitrary. For $j \in \mathcal{G}$, its batch consists of i.i.d. samples drawn from the same σ-subGaussian distribution $\mathcal{D}$ with mean μ (i.e. $\mathbb{E}_{X\sim\mathcal{D}}[X] = \mu$ and $\mathbb{E}_{X\sim\mathcal{D}}\left[\exp\left(s\left(X-\mu\right)\right)\right] \le \exp\left(\sigma^2 s^2/2\right)$, $\forall s \in \mathbb{R}$.). For $j \in \mathcal{B}$, $x_j^{[n_j]}$ can be arbitrary.*

Definition 3.1 considers a robust learning problem from batches where we allow arbitrarily different batch sizes. The corruption level α is the fraction of bad providers not data points; it is possible that a bad provider j has an overwhelmingly large n_j compared to other providers. In contrast, prior works (Qiao and Valiant, 2017; Chen et al., 2020; Jain and Orlitsky, 2021) have only studied the setting with (roughly) equal batch sizes. In many real-world crowd-sourcing applications, large and small data providers can differ drastically in the amount of data they provide, so our framework above captures broader application scenarios than prior works.

For this problem, we propose the COW (clique-overweight) algorithm (Algorithm 1). Given the empirical means of the batches $\hat{\mu}_j := \frac{1}{n_j}\sum_{i=1}^{n_j} x_j^i$, $j = 1, \ldots, m$, batch sizes $n_1, \ldots, n_m$, subGaussian parameter σ, corruption level $\alpha < 1/2$, and confidence level $\delta > 0$, COW first constructs a confidence interval I_j for the true mean μ on Line 1 using each batch j, where $I_j = \mathbb{R}$ if $n_j = 0$. With large probability, all good providers' intervals I_j should intersect because they contain μ. Define an undirected graph with nodes $I_1 \ldots I_m$, and I_i, I_j is connected by an edge if and only if $I_i \cap I_j \neq \emptyset$. Then we anticipate the good providers to form a large clique of size $(1-\alpha)m$. Accordingly, the algorithm finds the maximum clique in this graph. Of course, the maximum clique may contain some bad providers and miss some good providers. The second part of the algorithm reduces the influence of any "overweight" providers by cutting their effective batch size on Line 4, thus preventing bad providers in the clique to overwhelm the final mean estimate on Line 5.

Algorithm 1 COW

Require: Batch empirical means: $\hat{\mu}_1, \ldots, \hat{\mu}_m$; batch sizes: $n_1, \ldots, n_m$; subGaussian parameter σ; corruption level α; confidence level δ

1: $I_j \leftarrow \left[\hat{\mu}_j - \frac{\sigma}{\sqrt{n_j}}\sqrt{2\log\frac{2m}{\delta}}, \hat{\mu}_j + \frac{\sigma}{\sqrt{n_j}}\sqrt{2\log\frac{2m}{\delta}}\right]$, $\forall j \in [m]$

2: $C^* \leftarrow \mathrm{argmax}_{C \subseteq [m]: \bigcap_{j\in C} I_j \neq \emptyset} |C|$

3: $n^{\mathrm{cut}} \leftarrow$ the $(2\alpha m + 1)$-th largest batch size

4: $\tilde{n}_j \leftarrow \min(n_j, n^{\mathrm{cut}}), \forall j \in [m]$

5: **return** $\hat{\mu} \leftarrow \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in C^*}\tilde{n}_j\hat{\mu}_j$, Error $\leftarrow$ (1)

There can be multiple maximum cliques in Line 2; we break ties arbitrarily. A maximum clique can be computed efficiently.

We show that Algorithm 1 achieves the following guarantee.

Theorem 3.2. *Under Definition 3.1, if $n^{\mathrm{cut}} > 0$ and $\alpha < \frac{1}{2}$, then with probability at least $1-\delta$, the estimation error $|\hat{\mu} - \mu|$ of $\hat{\mu}$ returned by Algorithm 1 satisfies:*

$$\frac{2}{\sqrt{\sum_{j\in[m]}\tilde{n}_j}}\sigma\sqrt{2\log\frac{2}{\delta}} + \frac{8\alpha m\sqrt{n^{\mathrm{cut}}}}{\sum_{j\in[m]}\tilde{n}_j}\sigma\sqrt{2\log\frac{2m}{\delta}} \tag{1}$$

where n^{cut} and $\tilde{n}_j$'s are defined in Line 3 and Line 4 in Algorithm 1.

A few remarks are in order.

Remark 3.3 (Compare to prior work)**.** *Note that compared to prior works (Yin et al., 2018), our algorithm allows arbitrary batch sizes. Even if some agents report $n_j = 0$, as long as $n^{\mathrm{cut}} > 0$, i.e. there are at least $2\alpha m + 1$ agents reporting non-zero n_j's, our estimator will return a well-behaved estimator. In contrast, algorithms designed for equal batches will provably fail if the batches are imbalanced. (Yin et al., 2018) calculates the trimmed-mean of the empirical means of each batch. Suppose the clean data distribution is Gaussian $N(\mu, 1)$ and $3\alpha m$ batches have size $n^* >> m > 1$ while the rest of the batches have size 1, then the error of trimmed-mean is $\tilde{O}\left(\frac{1}{\sqrt{m}} + \alpha\right)$, Importantly, $O\left(\frac{1}{\sqrt{m}}\right)$ is much larger than $\tilde{O}\left(\frac{1}{\sqrt{m+\alpha m n^*}}\right)$, the optimal statistical rate without data corruption. On the contrary, Algorithm 1 returns an estimation with error $\tilde{O}\left(\frac{1}{\sqrt{m+\alpha m n^*}} + \frac{\alpha m n^*}{m+\alpha m n^*}\frac{1}{\sqrt{n^*}}\right) \le \tilde{O}\left(\frac{1}{\sqrt{n^*}}\right) << \tilde{O}\left(\frac{1}{\sqrt{m}}\right)$.*

Remark 3.4 (Equal batch size case)**.** *On the other hand, in case of equal batch sizes, i.e. $n_1 = \cdots = n_m = n$, (1) becomes $O\left(\frac{\sigma}{\sqrt{n}}\left(\frac{1}{\sqrt{m}} + \alpha\sqrt{\log m}\right)\right)$. This recovers the rate in (Yin et al., 2018), which is optimal (up to logarithmic factors). Therefore, our result strictly generalizes prior works on robust estimation from batches.*

Remark 3.5 (Robust mean estimation v.s. robust mean estimation from batches)**.** *In classical robust mean estimation setting (Huber, 1992; Diakonikolas et al., 2016), the optimal error rate is $O\left(\sigma\left(\alpha + \frac{1}{\sqrt{m}}\right)\right)$ given m total samples and α fraction corrupted samples. In contrast, due to having access to the data source ID, i.e. the batch indices, the learner can achieve significantly improved robustness. To see this, notice that the equal batch setting can be viewed as robust mean estimation from m data points $\hat{x}_j$'s. When the batch size n becomes larger, $\hat{x}_j$ has a smaller variance $\frac{\sigma^2}{n}$ and thus the error of robust mean estimation becomes $O\left(\frac{\sigma}{\sqrt{n}}\left(\alpha + \frac{1}{\sqrt{m}}\right)\right)$, which matches the above rate (up to logarithmic factors).*

Remark 3.6 (Dependency on the largest batches)**.** *Our bound in (1) does not depend on the largest $2\alpha m$ n_j's. This implies that even if some clean agents have infinite samples, the algorithm cannot achieve an error that diminishes to zero. This might not look ideal at first glance, but we show this is inevitable information-theoretically. Interested readers are referred to Theorem A.1.*

Remark 3.7 (Technical extensions)**.** *When the good data batch is subject to point-wise perturbation of magnitude at most ϵ, a variant of Algorithm 1 (Algorithm 4* PERT-COW*, see Section A.2) suffers at most a 2ϵ term in the error upper bound in addition to (1). In addition, Algorithm 1 does not require the exact dataset as input, but only the empirical mean and batch sizes of each data batch. As we shall see next, these two properties allow us to use* PERT-COW *in our byzantine-robust online RL algorithm to achieve low communication costs and preserve data privacy.*

4 BYZANTINE-ROBUST RL IN PARALLEL MDPS

Now, we are ready to study the problem of Byzantine-robust reinforcement learning in parallel *Markov Decision Processes* (MDPs). We consider a setting with one central server and m agents, α fraction of which may be adversarial. We postpone the precise interaction protocols between the server and agents to Section 5 and Section 6.

In both online and offline settings, we focus on finite horizon episodic tabular MDPs $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, H, \mu_1)$. Where $\mathcal{S}$ is the finite state space with $|\mathcal{S}| = S$; $\mathcal{A}$ is the finite action space with $|\mathcal{A}| = A$; $\mathcal{P} = \{P_h\}_{h=1}^H$ is the sequence of transition probability matrix, meaning $\forall h \in [H]$, $P_h : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ and $P_h(\cdot|s,a)$ specifies the state distribution in step $h+1$ if action a is taken from state s at step h; $\mathcal{R} = \{R_h\}_{h=1}^H$ is the sequence of bounded stochastic reward function, meaning $\forall h \in [H]$, $R_h(s,a)$ is the stochastic reward bounded in $[0,1]$ associated with taking action a in state s at step h; H is the time horizon; μ_1 is the initial state distribution. For simplicity, we assume μ_1 is deterministic and has probability mass 1 on state s_1.

Within each episode, the MDP starts at state s_1. At each step h, the agent observes the current state s_h and takes an action a_h and receives a stochastic reward $R_h(s_h, a_h)$. After that, the MDP transits to the next state s_{h+1}, which is drawn from $P_h(\cdot|s,a)$. The episode terminates after the agent takes action a_H in state s_H and receives reward $R_H(s_H, a_H)$ at step H.

A policy π is a sequence of functions $\{\pi_1, \ldots, \pi_H\}$, each maps from state space $\mathcal{S}$ to action space $\mathcal{A}$. The value function $V_h^\pi : \mathcal{S} \mapsto [0, H-h+1]$, is the expected sum of future rewards by taking action according to policy π, i.e. $V_h^\pi(s) := \mathbb{E}\left[\sum_{t=h}^H R_t(s_t, \pi_t(s_t)) \middle| s_h = s\right]$, where the expectation is w.r.t. to the stochasticity of state transition and reward in the MDP. Similarly, we define the state-action value function $Q_h^\pi : \mathcal{S} \times \mathcal{A} \mapsto [0, H-h+1]$: $Q_h^\pi(s,a) := \mathbb{E}\left[R_h(s,a)\right] + \mathbb{E}\left[\sum_{t=h+1}^H R_t(s_t, \pi_t(s_t)) \middle| s_h = s, a_h = a\right]$ Let $\pi^* = \left\{\pi^h\right\}$ be an optimal policy and let $V_h^*(s) := V_h^{\pi^*}(s,a)$, $Q_h^*(s) := Q_h^{\pi^*}(s,a), \forall h, s, a$.

For any $f : \mathcal{S} \mapsto [0, H]$, We define the Bellman operator by: $\left(\mathbb{B}_h f\right)(s,a) = \mathbb{E}\left[R_h(s,a)\right] + \mathbb{E}_{s' \sim P_h(\cdot|s,a)}[f(s')]$ Then the Bellman equation is given by:

$$
\begin{aligned}
V_h^\pi(s) =& Q_h^\pi(s, \pi_h(s)) && (2)\\
Q_h^\pi(s,a) =& \left(\mathbb{B}_h V_{h+1}^\pi\right)(s,a) && (3)\\
V_{H+1}^\pi(s) =& 0. && (4)
\end{aligned}
$$

The Bellman optimality equation is given by:

$$
\begin{aligned}
V_h^*(s) =& \max_{a \in \mathcal{A}} Q_h^*(s,a) && (5)\\
Q_h^*(s,a) =& \left(\mathbb{B}_h V_{h+1}^*\right)(s,a) && (6)\\
V_{H+1}^*(s) =& 0. && (7)
\end{aligned}
$$

We define the state distribution at step h by following policy π as $d_h^\pi(s) := P_h^\pi(s_h = s)$, and the state trajectory distribution of π as: $d^\pi := \{d_h^\pi\}_{h=1}^H$. The goal is to find a policy that maximizes the reward, i.e. find a $\hat{\pi}$, s.t. $V_1^{\hat{\pi}}(s_1) = V_1^*(s_1) = \max_\pi V_1^\pi(s_1)$. To measure the performance of our RL algorithms, we use suboptimality as our performance metric for offline settings and use regret as our performance metric for online settings. We formalize these two measures in their corresponding sections below.

5 BYZANTINE-ROBUST ONLINE RL

In the online setting, we assume that a central server and m agents aim to collaboratively minimize their total regrets. The agents and server collaborate by following a communication protocol to decide when to synchronize and what information to communicate. Unlike the standard distributed RL setting, we assume α-fraction of the agents are Byzantine:

Definition 5.1 (Distributed online RL with Byzantine corruption)**.** *There are m agents consisting of two types:*

- $(1-\alpha)m$ ***good agents, denoted by*** $\mathcal{G}$*: Each of the good agents interacts with a copy of $\mathcal{M}$ and communicates its observations to the server following the interaction protocol;*
- αm ***bad agents, denoted by*** $\mathcal{B}$*: The bad agents are allowed to communicate arbitrarily.*

Because the server has no control over the bad agents, we only seek to minimize the error incurred by the good agents. Formally, we use regret as our performance measure for the online RL algorithm:

$$\operatorname{Regret}(K) = \sum_{k=1}^{K} \sum_{j \in \mathcal{G}} \left(V_1^*(s_1) - V_1^{\pi_k^j}(s_1) \right), \tag{8}$$

where π_k^j is the policy used by agent j in episode k. At the same time, because of the distributed nature of our problem, we want to synchronize between the servers and agents only if necessary to reduce the communication cost.

Based on these considerations, we propose the BYZAN-UCBVI algorithm (Algorithm 2). We highlight the following key features of BYZAN-UCBVI:

1. **Low-switching-cost algorithm design**: the server will check the synchronization criteria in Line 6 when receiving agent requests. Each good agent will request synchronization if and only if any of their own (s, a, h) counts doubles (Line 23). Importantly, our agents do not need to know other agents' (s, a, h) counts to decide if synchronization is necessary. This design choice reduces the number of policy switches, synchronization rounds, and total communication costs, all from $O(K)$ to $O(\log K)$. Compared to the $O(\sqrt{K})$ communication steps in (Jadbabaie et al., 2022), ours is much lower. Unlike (Dubey and Pentland, 2021), our agents do not need to know other agents' transition counts to decide whether to synchronize.
2. **Homogeneous policy execution**: In any episode k, our algorithm is designed so that all good agents are running the same policy π_k. This ensures that the robust mean estimation achieves the smallest estimation error. Recall that the samples in the large batches are wasted if the batch sizes are severely imbalanced (cf. Section 3).

Algorithm 2 BYZAN-UCBVI (K, δ, α)

1: [S]$\hat{V}_{H+1}(\cdot) \leftarrow 0$, $\hat{Q}_{H+1}(\cdot,\cdot) \leftarrow 0$, $\text{SyncCount}_j \leftarrow -1, \forall j \in [m]$, $\text{Sync}_j \leftarrow \text{TRUE}, \forall j \in [m]$ $\delta' \leftarrow \frac{\delta}{(SAHKm)^{3S}}$, $\epsilon \leftarrow \frac{1}{SAHKm}$ {We use [S] to denote the action of central server}
2: [A]$N_h^j(s,a) \leftarrow 0$, $D_h^j \leftarrow \emptyset, \forall (j,h,s,a) \in \mathcal{G} \times [H] \times \mathcal{S} \times \mathcal{A}$ {We use [A] to denote the action of agents}
3: **for** episode $k \in [K]$ **do**
4:   [S] Receive $\text{Sync}_1, \text{Sync}_2, \dots, \text{Sync}_m$
5:   **for** agent $j \in [m]$ **do**
6:     **if** Sync_j and $\text{SyncCount}_j \le SAH \log_2 K$ **then**
7:       [S] $\text{SyncCount}_j \leftarrow \text{SyncCount}_j + 1$
8:       [S] SYNCHRONIZE $\leftarrow$ TRUE
9:   **if** SYNCHRONIZE **then**
10:     [A] $N_{h,j}^{\text{old}}(s,a) \leftarrow N_h^j(s,a), \forall s,a,h,j$
11:     **for** $h = H, H-1, \dots, 1$ **do**
12:       [S] Communicate $\hat{V}_{h+1}(\cdot)$ to each agent
13:       **for** $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
14:         [A] send $x_j \leftarrow \frac{\sum_{(s,a,r,s') \in D_h^j} r + \hat{V}_{h+1}(s')}{N_h^j(s,a)}$, $n_j \leftarrow N_h^j(s,a)$ to Server, $\forall j \in \mathcal{G}$
15:         [S] $\left(\hat{\mathbb{B}}_h \hat{V}_{h+1}\right)(s,a), \Gamma_h(s,a) \leftarrow$ PERT-COW $\left(x_{[m]}, n_{[m]}, H-h+1, \alpha, \epsilon, \delta'\right)$
16:         $\Gamma_h(s,a) \leftarrow \min(H-h+1, \Gamma_h(s,a)+\epsilon)$
17:       [S] Compute $\bar{Q}_h, \hat{Q}_h, \hat{\pi}_h, \hat{V}_h$ as in (9)-(12).
18:   [S] SYNCHRONIZE $\leftarrow$ FALSE
19:   **for** $j \in \mathcal{G}$ **do**
20:     [A] $\text{Sync}_j \leftarrow$ FALSE
21:     [A] Sample $\left\{(s_h^{j,k}, a_h^{j,k}, r_h^{j,k}, s_{h+1}^{j,k})\right\}_{h \in [H]}$ under $\{\hat{\pi}_h\}_{h=1}^H$
22:     [A] $\forall h$, $N_h^j(s_h^{j,k}, a_h^{j,k}) \leftarrow N_h^j(s_h^{j,k}, a_h^{j,k}) + 1$, $D_h^j \leftarrow D_h^j \cup \left\{(s_h^{j,k}, a_h^{j,k}, r_h^{j,k}, s_{h+1}^{j,k})\right\}$
23:     [A] Send Sync request to Server, if $\text{Sync}_j \leftarrow \mathbf{1}\left\{\max_{s,a,h} \frac{N_h^j(s,a)}{N_{h,j}^{\text{old}}(s,a)} \ge 2\right\}$ is TRUE.
24: **return** $\{\hat{\pi}_h\}_{h=1}^H$

3. **Robust UCBVI updates**: During synchronization, the central server performs policy update using a variant of the UCBVI algorithm (Azar et al., 2017): for $h = H, H-1, \dots, 1$, compute:

$$\bar{Q}_h(\cdot,\cdot) = \left(\hat{\mathbb{B}}_h \hat{V}_{h+1}\right)(\cdot,\cdot) + \Gamma_h(\cdot,\cdot) \tag{9}$$

$$\hat{Q}_h(\cdot,\cdot) = \min\left\{\bar{Q}_h(\cdot,\cdot), H-h+1\right\}^+ \tag{10}$$

$$\hat{\pi}_h(\cdot) = \operatorname*{argmax}_a \hat{Q}_h(\cdot, a) \tag{11}$$

$$\hat{V}_h(\cdot) = \max_a \hat{Q}_h(\cdot, a). \tag{12}$$

The main difference lies in Line 15, where we replace the standard mean and confidence interval estimation with our PERT-COW algorithm (Algorithm 4). Instead of estimating the transition matrix and reward function, we directly estimate the Bellman operator given an estimated value function $\hat{V}_{h+1}$. The server gathers sufficient statistics from agents in Line 14. According to Algorithm 4, when $n^{\text{cut}} \leq 0$, the $\Gamma_h(s,a)$ in Line 15 is set to be ∞ as a trivial error bound. Line 16 adjust the bonus to be the range of the value function. The an additional ϵ is an adjustment for ϵ-cover argument in the proof of Theorem 5.2.

We are now ready to present the following regret bound for BYZAN-UCBVI.

Theorem 5.2 (Regret bound). *Under Definition 5.1, if $\alpha \leq \frac{1}{3}\left(1-\frac{1}{m}\right)$, for all $\delta < \frac{1}{4}$, with probability at least $1-3\delta$, the total regret of Algorithm 2 is at most*

$$\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\left(V_1^*(s_1) - V_1^{\hat{\pi}_k^j}(s_1)\right) \tag{13}$$

$$=\tilde{O}\left((1+\alpha\sqrt{m})H^2S\sqrt{AmK\log(1/\delta)}\right). \tag{14}$$

Remark 5.3 (Understanding the regret bound). *In Algorithm 2, the good agents are using the same policy, and thus for all $j \in \mathcal{G}$, $\hat{\pi}_k^j = \hat{\pi}_k$, where $\hat{\pi}_k$ is the policy calculated by the server in k-th episode. By utilizing the batch structure, Algorithm 2 achieves a regret sublinear in K, even under Byzantine attacks. Our regret is only $O(\sqrt{mK}+\alpha m\sqrt{K}))$ compared to the $O(m\sqrt{K}+m\alpha^{1/4}K^{3/4})$ regret in (Jadbabaie et al., 2022). When $\alpha \leq 1/\sqrt{m}$, the dominating term $\sqrt{mK}$ is optimal even in the clean setting (Azar et al., 2017).*

Remark 5.4 (The Breakdown point). *We require α to be smaller than $\frac{1}{3}$ because we can show that with high probability, all of the good agents will have visitation on some (s,a) pair and simply restricting $\alpha \leq \frac{1}{3}$ ensures the n^{cut} in Algorithm 4 is greater than 0, which meets the requirement in Theorem 3.2 and allows for a cleaner exposition of Theorem 5.2.*

Remark 5.5 (Communication cost). *Because each agent runs K episodes in total, the count of each of the (s,a,h) tuples doubles at most $\lfloor \log_2 K \rfloor$ times during training. Thus each good agent will send at most $SAH\lfloor \log_2 K \rfloor$ sync requests. The bad agents can only send a logarithmic number of effective requests because of the checking step in Line 6. As a result, there will be at most $mSAH\lfloor \log_2 K \rfloor$ synchronization episodes in total. The communication inside one synchronization episode includes the following: at least one agent sends a sync request; inside the value iteration, the server will send estimated value functions at H steps to each agent; Each good agent will send the estimated Bellman operator for each (s,a) pair at H steps and the counts to the server. Importantly, the agents only need to send summary statistics instead of the raw dataset to the server. This preserves the data privacy of individual agents (Sakuma et al., 2008; Liu et al., 2019).*

Remark 5.6 (Switching cost). *Switching cost measures the number of policy changes. Algorithms with low switching costs are favorable in real-world applications (Bai et al., 2019; Zhang et al., 2020b; Gao et al., 2021). Algorithm 2 only performs policy updates during synchronization episodes. Its switching cost is thus at most $mSAH\lfloor \log_2 K \rfloor$.*

6 BYZANTINE-ROBUST OFFLINE RL

In the offline setting, we assume the server has access to logged interaction data from many agents, among which some are adversarial. The goal of the server is to find a nearly optimal policy using this collection of offline datasets without further interaction with the environment. Specifically:

Definition 6.1 (Distributed offline RL with Byzantine corruption). *The server has access to an offline data set with m data batches $\bigcup_{j\in[m]} D_j$, including $(1-\alpha)m$ good batches $\mathcal{G}$ and αm bad batches $\mathcal{B}$, where $D_j := \bigcup_{h\in[H]} D_j^h := \bigcup_{h\in[H]}\left\{\left(s_h^{j,k}, a_h^{j,k}, r_h^{j,k}, {s'}_h^{j,k}\right)\right\}_{k=1}^{K_j}$. We make an assumption on the data generating process similar to (Wang et al., 2020). Specifically, for all $j \in \mathcal{G}$, D_j is drawn from an unknown distribution $\left\{\nu_h^j\right\}_{h=1}^{H}$, where for each $h \in [H]$, $\nu_h^j \in \Delta(\mathcal{S}\times\mathcal{A})$. For all h,j,k, $\left(s_h^{j,k}, a_h^{j,k}\right) \sim \nu_h^j$, ${s'}_h^{j,k} \sim P_h(\cdot|s_h^{j,k}, a_h^{j,k})$ and $r_h^{j,k}$ is an instantiation of $R_h\left(s_h^{j,k}, a_h^{j,k}\right)$. For any $j \in \mathcal{B}$ (i.e. bad batches), D_j can be arbitrary.*

The performance is measured by the suboptimality w.r.t. a deterministic comparator policy $\tilde{\pi}$ (not necessarily an optimal policy):

$$\mathrm{SubOpt}\left(\pi,\tilde{\pi}\right) := V_1^{\tilde{\pi}}(s_1) - V_1^{\pi}(s_1). \tag{15}$$

In the offline setting, the server cannot interact with the MDP. So our result relies heavily on the quality of the dataset. As we will see in the analysis, the suboptimality gap (15) can be upper bounded by the estimation error of the Bellman operator along the trajectory of $\tilde{\pi}$. As a result, we do not need full coverage over the whole state-action space. Instead, we only need the offline dataset to have proper coverage over $\{d_h^{\tilde{\pi}}\}_{h=1}^{H}$, the state distribution of policy $\tilde{\pi}$ at each step h. To characterize the data coverage, for any s,a,h, we define the counts on (s,a,h) tuples by:

$$N_h^j(s,a) := \sum_{k\in[K_j]} \mathbf{1}\left\{(s_h^{j,k}, a_h^{j,k}) = (s,a)\right\}, \quad \forall j \in [m]. \tag{16}$$

When calling Algorithm 1, the large data batches might be clipped in Line 4. By definition, the clipping threshold is bounded between $N_h^{\mathcal{G},\text{cut}_1}(s,a)$, the $(\alpha m+1)$-th largest of

$\left\{N_h^j(s,a)\right\}_{j\in\mathcal{G}}$ and $N_h^{\mathcal{G},\mathrm{cut}_2}(s,a)$, the $(2\alpha m+1)$-th largest of $\left\{N_h^j(s,a)\right\}_{j\in\mathcal{G}}$. We define three quantities $p^{\mathcal{G},0}$, κ, κ_{even} to characterize the quality of the offline dataset. The first quantity describes the density of $\tilde{\pi}$ trajectory that is not properly covered by the offline dataset:

Definition 6.2 (Measure of insufficient coverage)**.** *We define $p^{\mathcal{G},0}$ as the probability of $\tilde{\pi}$ visiting an (s,h,a) tuple that is insufficiently covered by the logged data, namely*

$$p^{\mathcal{G},0} := \sum_{h=1}^{H} \mathbb{E}_{d_h^{\tilde{\pi}}}\left[\mathbf{1}\left\{N_h^{\mathcal{G},\mathrm{cut}_2}(s,\tilde{\pi}(s))=0\right\}\right]. \tag{17}$$

Recall that Algorithm 1 requires there are at least $(2\alpha m+1)$ non-empty data batches to make an informed decision. $p^{\mathcal{G},0}$ measures an upper bound on the total probability under $d^{\tilde{\pi}}$ to encounter an (s,h,a) on which COW cannot return a good mean estimator.

We now introduce κ, the density ratio between the $d^{\tilde{\pi}}$, and the empirical distribution of the uncorrupted offline dataset. κ quantifies the portion of useful data in the whole dataset and is commonly used in the offline RL literature (Rashidinejad et al., 2021; Zhang et al., 2021a). We only focus on the (s,a,h) tuples excluded by $p^{\mathcal{G},0}$ in Definition 6.2:

Definition 6.3 (density ratio)**.** *We use $\{\mathcal{C}_h\}_{h=1}^H$ to denote the state space (in the support of $\{d_h^{\tilde{\pi}}\}_{h=1}^H$) that have proper clean agents coverage:*

$$\mathcal{C}_h = \left\{s | N_h^{\mathcal{G},\mathrm{cut}_2}(s,\tilde{\pi}(s))>0\right\}. \tag{18}$$

We use κ to denote the density ratio between the state distribution of policy $\tilde{\pi}$ and the empirical distribution over the uncorrupted offline dataset:

$$\kappa := \max_{h\in[H]}\max_{s\in\mathcal{C}_h} \frac{d_h^{\tilde{\pi}}(s)}{\sum_{j\in\mathcal{G}} N_h^j(s,\tilde{\pi}_h(s))/\sum_{j\in\mathcal{G}} K_j}. \tag{19}$$

As we can see in Theorem 3.2, the accuracy of Algorithm 1 heavily depends on the evenness of the batches. We define the following quantity to measure the information loss in the clipping step (Line 4 in Algorithm 1):

Definition 6.4 (Unevenness of good agents coverage)**.**

$$\kappa_{even} := \max_{h\in[H]}\max_{s\in\mathcal{C}_h} \frac{\sum_{j\in\mathcal{G}} N_h^j(s,\tilde{\pi}_h(s))}{\sum_{j\in\mathcal{G}} \tilde{N}_h^{j,\mathrm{cut}_2}(s,\tilde{\pi}_h(s))} \tag{20}$$

$$\cdot \frac{m(1-\alpha)N_h^{\mathcal{G},\mathrm{cut}_1}(s,\tilde{\pi}_h(s))}{\sum_{j\in\mathcal{G}} \tilde{N}_h^{j,\mathrm{cut}_2}(s,\tilde{\pi}_h(s))}, \tag{21}$$

where $\tilde{N}_h^{j,\mathrm{cut}_2}(s,\tilde{\pi}_h(s)) = \max\left(N_h^{\mathcal{G},\mathrm{cut}_2}(s,\tilde{\pi}_h(s)), N_h^j(s,\tilde{\pi}_h(s))\right)$.

Intuitively, κ_{even} describes the unevenness of good agent coverage. It takes into account both how much data in large batches are cut off by the clipping step and the unevenness of the batches after clipping. We include $N_h^{\mathcal{G},\mathrm{cut}_1}(s,\tilde{\pi}_h(s))$ and $N_h^{\mathcal{G},\mathrm{cut}_2}(s,\tilde{\pi}_h(s))$, instead of the true clipping threshold, meaning κ_{even} serves as an upper bound of the actual unevenness resulting from running the algorithm. For example, suppose $\alpha m>1$: if for any s,a,h,j, $N_h^j(s,a)=n$, then $\kappa_{\text{even}}=1$; if for any s,a,h, there is one good data batch with size Lm for some $L>1$ while the others have size 1, then $N_h^{\mathcal{G},\mathrm{cut}_1}(s,a)=N_h^{\mathcal{G},\mathrm{cut}_2}(s,a)=1$ and $\kappa_{\text{even}}=\frac{Lm+(1-\alpha)m-1}{(1-\alpha)m}\frac{(1-\alpha)m}{(1-\alpha)m}\approx L+1$, meaning κ_{even} increases as the batches become less even.

Remarkably, all three quantities defined above only depend on the (s,a,h) counts of the good data batches.

Given the above setup, we now present our second algorithm, BYZAN-PEVI, a Byzantine-Robust variant of pessimistic value iteration (Jin et al., 2021). Similar to the online setting, we use our COW (without perturbation) algorithm to approximate the Bellman operator and use the estimation error to design the PESSIMISTIC bonus for the value iteration. BYZAN-PEVI (Algorithm 3) runs pessimistic value iteration ((22)-(25)) and calls COW as a subroutine to robustly estimate the Bellman operator using offline dataset D:

$$\bar{Q}_h(\cdot,\cdot) = \left(\hat{\mathbb{B}}_h\hat{V}_{h+1}\right)(\cdot,\cdot) - \Gamma_h(\cdot,\cdot) \tag{22}$$

$$\hat{Q}_h(\cdot,\cdot) = \min\left\{\bar{Q}_h(\cdot,\cdot), H-h+1\right\}^+ \tag{23}$$

$$\hat{\pi}_h(\cdot) = \operatorname*{argmax}_a \hat{Q}_h(\cdot,a) \tag{24}$$

$$\hat{V}_h(\cdot) = \max_a \hat{Q}_h(\cdot,a). \tag{25}$$

Theorem 6.5. *Given any deterministic comparator policy $\tilde{\pi}$, under Definition 6.1, 6.2, 6.3 and 6.4: for any δ, $\alpha<\frac{1}{3}$, with probability at least $1-\delta$, Algorithm 3 outputs a policy $\hat{\pi}$ with:*

$$\begin{aligned}&\mathrm{SubOpt}(\hat{\pi},\tilde{\pi}) \le 2Hp^{\mathcal{G},0}\\ &+O\left(\sqrt{\kappa\kappa_{even}}H^2\sqrt{S}\frac{1+\sqrt{m}\alpha}{\sqrt{\sum_{j\in\mathcal{G}}K_j}}\sqrt{\log\frac{HSAm}{\delta}}\right).\end{aligned} \tag{26}$$

Remark 6.6 (Understanding the sub-optimality gap)**.** *The sub-optimality gap (26) depends on both the offline data distribution (characterized by $p^{\mathcal{G},0}$, κ and κ_{even}) and number of clear samples $\sum_{j\in\mathcal{G}}K_j$. The first term only depends on the coverage of the data distribution and will not shrink with a larger sample size. When for each (s,a,h), all agents visit the tuple for equal times, we have $\kappa_{even}=1$. Furthermore, let $K_j=K$ for all $j\in[m]$, RHS of (26) becomes:*

$$2Hp^{\mathcal{G},0}$$

Algorithm 3 BYZAN-PEVI

Require: $D := \bigcup_{j\in[m]} D_j := \bigcup_{h\in[H]} D_j^h := \bigcup_{h\in[H]} \left\{ \left(s_h^{j,k}, a_h^{j,k}, r_h^{j,k}, {s'}_h^{j,k} \right) \right\}_{k=1}^{K_j}$, α, δ

$\delta' \leftarrow \frac{\delta}{H|\mathcal{S}||\mathcal{A}|m}$
$\hat{V}_{H+1}(\cdot) \leftarrow 0$
for $h = H, H-1, \ldots, 1$ **do**
  $\sigma \leftarrow H - h + 1$
  for $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
    for $j \in \mathcal{G}$ **do**
      $n_j \leftarrow \sum_{k\in[K_j]} \mathbf{1}\left\{ (s_h^{j,k}, a_h^{j,k}) = (s,a) \right\}$
      $x_j \leftarrow \frac{1}{N_h^j(s,a)} \sum_{(s,a,r,s')\in D_h^j} \left(r + \hat{V}_{h+1}(s') \right)$
    if $|j \in [m] : n_j > 0| \geq 2\alpha m + 1$ **then**
      $\left(\hat{\mathbb{B}}_h \hat{V}_{h+1} \right)(s,a), \Gamma_h(s,a) \leftarrow$ COW $\left(x_{[m]}, n_{[m]}, \sigma, \alpha, \delta' \right)$
    else
      $\left(\hat{\mathbb{B}}_h \hat{V}_{h+1} \right)(s,a) \leftarrow 0, \Gamma_h(s,a) \leftarrow H - h + 1$
  Compute $\bar{Q}_h, \hat{Q}_h, \hat{\pi}_h, \hat{V}_h$ as in (22)-(25).
return $\{\hat{\pi}_h\}_{h=1}^H$

$$+O\left(\sqrt{\kappa} H^2 \sqrt{S} \frac{1}{\sqrt{mK}} \sqrt{\log \frac{HSAm}{\delta}} \right)$$
$$+O\left(\sqrt{\kappa} H^2 \sqrt{S} \frac{\alpha}{\sqrt{K}} \sqrt{\log \frac{HSAm}{\delta}} \right),$$

where the first term measures the effect of lack of coverage, the second term is the statistical error and the third term is the bias term due to the data corruption. Importantly, both the second and the third terms vanish as $K \to \infty$, whereas the first term is due to the lack of data coverage. On the contrary, (Zhang et al., 2021a) has a non-diminishing bias term due to data corruption. To the best of our knowledge, this is the first result for Byzantine-robust offline RL.

Remark 6.7 (Offline v.s. online RL)**.** *Our offline RL results are more involved and notation-heavy due to the nature of the problem. In the offline RL setting, the learner cannot control the data-generating process, and each data source can be arbitrarily different. The agent can only passively rely on the robust mean estimator we designed and the pessimism principle to learn as well as the data permits. In contrast, the learner has complete control over the clean agents' data collection process in the online setting. Our algorithm* BYZAN-UCBVI *enables the server to realize its full potential and obtain a tighter and cleaner sample complexity guarantee.*

7 CONCLUSION

To summarize, in this work, we start by presenting COW, a robust mean estimation algorithm for learning from uneven batches. Building upon COW, we propose byzantine-robust online (BYZAN-UCBVI) and the first byzantine-robust offline (BYZAN-PEVI) reinforcement learning algorithms in the distributed setting. Several questions remain open: (1) Can we provide a complete characterization of the information-theoretical lower bound for robust mean estimation from uneven batches? (2) Can we extend our RL algorithms to the function approximation setting? Allowing function approximation is essential to apply our algorithm to empirical evaluations. However, this would require a computationally efficient high-dimensional robust mean estimator from uneven batches, which is highly nontrivial. Therefore, we defer the generalization to the function approximation setting and empirical evaluation of our framework as an important direction for future research.

Acknowledgements

We thank Kirthevasan Kandasamy for the helpful discussions. Mengdi Wang acknowledges the support by NSF grants DMS-1953686, IIS-2107304, CMMI-1653435, ONR grant 1006977, and C3.AI. Kaiqing Zhang acknowledges the support by Simons-Berkeley Research Fellowship. This project is supported in part by NSF grants 1545481, 1704117, 1836978, 2023239, 2041428, 2202457, ARO MURI W911NF2110317, and AF CoE FA9550-18-1-0166.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Mridul Agarwal, Bhargav Ganguly, and Vaneet Aggarwal. Communication efficient parallel reinforcement learning. In *Uncertainty in Artificial Intelligence*, pages 247–256. PMLR, 2021.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

Yu Bai, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. Provably efficient q-learning with low switching cost. *Advances in Neural Information Processing Systems*, 32, 2019.

Ana LC Bazzan. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, 18(3):342–375, 2009.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Siu-On Chan, Ilias Diakonikolas, Paul Valiant, and Gregory Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1193–1203. SIAM, 2014.

Sitan Chen, Jerry Li, and Ankur Moitra. Efficiently learning structured distributions from untrusted batches. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 960–973, 2020.

Tianyi Chen, Kaiqing Zhang, Georgios B Giannakis, and Tamer Basar. Communication-efficient policy gradient methods for distributed reinforcement learning. *IEEE Transactions on Control of Network Systems*, 2021.

Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 655–664. IEEE, 2016.

Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, pages 999–1008. PMLR, 2017.

Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pages 1596–1606. PMLR, 2019.

Guohui Ding, Joewie J Koh, Kelly Merckaert, Bram Vanderborght, Marco M Nicotra, Christoffer Heckman, Alessandro Roncone, and Lijun Chen. Distributed reinforcement learning for cooperative multi-robot object manipulation. *arXiv preprint arXiv:2003.09540*, 2020.

Abhimanyu Dubey and Alex Pentland. Private and byzantine-proof cooperative decision-making. In *AAMAS*, pages 357–365, 2020.

Abhimanyu Dubey and Alex Pentland. Provably efficient cooperative multi-agent reinforcement learning with function approximation. *arXiv preprint arXiv:2103.04972*, 2021.

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.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Wei Jing, Cheston Tan, and Bryan Kian Hsiang Low. Fault-tolerant federated reinforcement learning with theoretical guarantee. *Advances in Neural Information Processing Systems*, 34, 2021.

David A Freedman. On tail probabilities for martingales. *the Annals of Probability*, pages 100–118, 1975.

Minbo Gao, Tianle Xie, Simon S Du, and Lin F Yang. A provably efficient algorithm for linear markov decision process with low switching cost. *arXiv preprint arXiv:2101.00494*, 2021.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

Ali Jadbabaie, Haochuan Li, Jian Qian, and Yi Tian. Byzantine-robust federated linear bandits. *arXiv preprint arXiv:2204.01155*, 2022.

Ayush Jain and Alon Orlitsky. Robust density estimation from batches: The best things in life are (nearly) free. In *International Conference on Machine Learning*, pages 4698–4708. PMLR, 2021.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

R Matthew Kretchmar. Parallel reinforcement learning. In *The 6th World Conference on Systemics, Cybernetics, and Informatics*. Citeseer, 2002.

Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016.

LESLIE LAMPORT, ROBERT SHOSTAK, and MARSHALL PEASE. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4 (3):382–401, 1982.

Ximeng Liu, Robert H Deng, Kim-Kwang Raymond Choo, and Yang Yang. Privacy-preserving reinforcement learning design for patient-centric dynamic treatment regimes. *IEEE Transactions on Emerging Topics in Computing*, 9 (1):456–470, 2019.

Thodoris Lykouris, Max Simchowitz, Alex Slivkins, and Wen Sun. Corruption-robust exploration in episodic reinforcement learning. In *Conference on Learning Theory*, pages 3242–3245. PMLR, 2021.

Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu. Policy poisoning in batch reinforcement learning and control. *Advances in Neural Information Processing Systems*, 32, 2019.

Yuzhe Ma, J Sharp, Ruizhe Wang, Earlence Fernandes, and Xiaojin Zhu. Adversarial attacks on kalman filter-based forward collision warning systems. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021.

Gina Neff and Peter Nagy. Automation, algorithms, and politics| talking to bots: Symbiotic agency and the case of tay. *International Journal of Communication*, 10:17, 2016.

Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 54(10):4750–4755, 2008.

Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

Mingda Qiao and Gregory Valiant. Learning discrete distributions from untrusted batches. *arXiv preprint arXiv:1711.08113*, 2017.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34, 2021.

Jun Sakuma, Shigenobu Kobayashi, and Rebecca N Wright. Privacy-preserving reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 864–871, 2008.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

John W Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485, 1960.

Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *ACM Computing Surveys (CSUR)*, 53(2):1–33, 2020.

Ruosong Wang, Dean P Foster, and Sham M Kakade. What are the statistical limits of offline rl with linear function approximation? *arXiv preprint arXiv:2010.11895*, 2020.

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.

Tao Yu, HZ Wang, Bin Zhou, Ka Wing Chan, and J Tang. Multi-agent correlated equilibrium q (λ) learning for coordinated smart generation control of interconnected power grids. *IEEE transactions on power systems*, 30(4):1669–1679, 2014.

Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 11225–11234. PMLR, 2020a.

Xuezhou Zhang, Yiding Chen, Jerry Zhu, and Wen Sun. Corruption-robust offline reinforcement learning. *arXiv preprint arXiv:2106.06630*, 2021a.

Xuezhou Zhang, Yiding Chen, Xiaojin Zhu, and Wen Sun. Robust policy gradient against strong data corruption. In *International Conference on Machine Learning*, pages 12391–12401. PMLR, 2021b.

Zihan Zhang, Yuan Zhou, and Xiangyang Ji. Almost optimal model-free reinforcement learningvia reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33:15198–15207, 2020b.

A MORE DISCUSSION ON Algorithm 1:COW

A.1 Impossibility Result

Theorem A.1 (impossibility result)**.** *There exists a distribution $\mathcal{D}$, s.t. given m data batches $\left\{\{x_j^i\}_{i=1}^{n_j}\right\}_{j\in[m]}$ generated under Definition 3.1, every robust mean estimation algorithm $\mathscr{A}$ suffers an error of at least*

$$\Omega\left(\frac{1}{\sqrt{N}}\right) \tag{27}$$

even $\mathscr{A}$ knows some of the batches are clean, where N is the sum of sizes of the smallest $(1-2\alpha)m$ good batches.

Proof of Theorem A.1. Let $\mathcal{D}$ be Bernoulli distribution with parameter $\frac{1}{2}$. W.l.o.g., assume $\mathcal{G} = [(1-\alpha)m]$, $n_1 \leq \cdots \leq n_{(1-\alpha)m}$ and $\mathcal{B} = \{(1-\alpha)m+1, \ldots, m\}$. We assume algorithm $\mathscr{A}$ knows $[(1-2\alpha)m]$ is a subset of the good batches. Let $\eta = \frac{1}{2\sqrt{N}} = \frac{1}{2\sqrt{\sum_{j=1}^{(1-2\alpha)m} n_j}}$. Let the bad batches $\mathcal{B}$ be i.i.d. samples from $\mathcal{D}'$, a Bernoulli distribution with parameter $\frac{1}{2}+\eta$. By Theorem 4 of (Paninski, 2008; Chan et al., 2014), no algorithm can distinguish if the batches $\{x_1^i\}_{i=1}^{n_1}, \ldots, \left\{x_{(1-2\alpha)m}^i\right\}_{i=1}^{n_{(1-2\alpha)m}}$ are sampled from $\mathcal{D}$ or $\mathcal{D}'$. I.e. no algorithm can distinguish if $\{(1-2\alpha)m+1, \ldots, (1-\alpha)m\}$ are good batches or $\mathcal{B}$ are good batches. This means, given m data batches $\left\{\{x_j^i\}_{i=1}^{n_j}\right\}_{j\in[m]}$, every robust mean estimation algorithm suffers an error at least $\Omega\left(\frac{1}{\sqrt{N}}\right)$. ☐

A.2 Adaption To Good Batch Perturbation And Distributed Learning

Compared to Algorithm 1, Algorithm 4 enlarges the confidence interval by ϵ on both endpoints due to the perturbation and only requires some sufficient statistics from the batches, instead of the whole dataset. When $n^{\text{cut}} > 0$, meaning there are at least $2\alpha m+1$ non-empty batches, Algorithm 4 runs a modified COW algorithm to calculate the mean estimation and the error upper bound . When $n^{\text{cut}} = 0$, Algorithm 4 returns 0 and a trivial error upper bound.

Algorithm 4 PERT-COW

Require: Batch empirical means: $\hat{\mu}_1, \ldots, \hat{\mu}_m$; batch sizes: $n_1, \ldots, n_m$; subGaussian parameter σ; corruption level α; confidence level δ

1: $n^{\text{cut}} \leftarrow$ the $(2\alpha m+1)$-th largest batch size
2: **if** $n^{\text{cut}} \leq 0$ **then**
3: Error $\leftarrow \infty$
4: **return** $\hat{\mu} \leftarrow 0$, Error
5: $I_j \leftarrow \left[\hat{\mu}_j - \frac{\sigma}{\sqrt{\tilde{n}_j}}\sqrt{2\log\frac{2m}{\delta}} - \epsilon, \hat{\mu}_j + \frac{\sigma}{\sqrt{\tilde{n}_j}}\sqrt{2\log\frac{2m}{\delta}} + \epsilon\right], \forall j \in [m]$
6: $C^* \leftarrow \operatorname{argmax}_{C\subseteq[m]:\bigcap_{j\in C} I_j \neq \emptyset} |C|$
7: $\tilde{n}_j \leftarrow \min(n_j, n^{\text{cut}}), \forall j \in [m]$
8: $\hat{\mu} \leftarrow \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in C^*}\tilde{n}_j\hat{\mu}_j,$
9: Error $\leftarrow$ RHS of (31)
10: **return** $\hat{\mu}$, Error

B PROOF OF Theorem 3.2

To prove Theorem 3.2, we show (1) holds under some concentration event while the event happens with high probability. We consider a slightly more general setting where there could be perturbations to even good batches:

Definition B.1 (Robust mean estimation from batches)**.** *There are m data providers indexed by: $\{1, 2, \ldots, m\} =: [m]$. Among these providers, we denote the indexes of uncorrupted providers by $\mathcal{G}$ and the indexes of corrupted providers by $\mathcal{B}$, where $\mathcal{B}\cup\mathcal{G} = [m]$, $\mathcal{B}\cap\mathcal{G} = \emptyset$, $|\mathcal{B}| = \alpha m$. Any uncorrupted providers have access to* ***perturbed*** *samples from a sub-Gaussian distribution $\mathcal{D}$ with mean μ and variance proxy σ^2 (i.e. $\mathbb{E}_{X\sim\mathcal{D}}[X] = \mu$ and $\mathbb{E}_{X\sim\mathcal{D}}\left[\exp\left(s\left(X-\mu\right)\right)\right] \leq \exp\left(\sigma^2 s^2/2\right)$, $\forall s \in \mathbb{R}$.).* ***For each $j \in \mathcal{G}$, a data batch $\{\tilde{x}_j^i\}_{i=1}^{n_j}$ is drawn from $\mathcal{D}$, while a perturbed version $\{x_j^i\}_{i=1}^{n_j}$ is sent to the learner, where n_j can be arbitrary and $\left|x_j^i - \tilde{x}_j^i\right| \leq \epsilon$ for some*** *$\epsilon \geq 0$. For $j \in \mathcal{B}$, $\{x_j^i\}_{i=1}^{n_j}$ can be arbitrary.*

One can easily recover Definition 3.1 by letting $\epsilon = 0$. Algorithm 4 only requires the empirical mean $\hat{\mu}_j := \frac{1}{n_j}\sum_{i=1}^{n_j} x_j^i$ and size n_j of each batch $j \in [m]$.. We first define the concentration event as follows:

Definition B.2 (Concentration event)**.** *For all $j \in \mathcal{G}$, define the event that the empirical mean of clean batches is close to the population mean as:*

$$\mathcal{E}_j := \left\{ |\hat{\mu}_j - \mu| \leq \frac{\sigma}{\sqrt{n_j}} \sqrt{2\log\frac{2m}{\delta}} + \epsilon \right\} \tag{28}$$

Define the event that the weighted average of empirical means of clean batches is close to the population mean as:

$$\mathcal{E}_{wa} := \left\{ \left| \frac{1}{\sum_{j\in\mathcal{G}} \tilde{n}_j} \sum_{j\in\mathcal{G}} \tilde{n}_j \hat{\mu}_j - \mu \right| \leq \frac{\sigma}{\sqrt{\sum_{j\in\mathcal{G}} \tilde{n}_j}} \sqrt{2\log\frac{2}{\delta}} + \epsilon \right\} \tag{29}$$

Let $\mathcal{E}_{conc}$ be the event that the events above happen together:

$$\mathcal{E}_{conc} := \mathcal{E}_{wa} \cap \bigcap_{j\in\mathcal{G}} \mathcal{E}_j \tag{30}$$

We can show $\mathcal{E}_{conc}$ happens with high probability using Hoeffding's inequality:

Lemma B.3. $\mathbb{P}\left(\mathcal{E}_{conc}\right) \geq 1 - 2\delta$.

Proof. See proof in Section B.1. □

Under event $\mathcal{E}_{conc}$, we can give an upper bound on the estimation error:

Lemma B.4. *Under event $\mathcal{E}_{conc}$, if $n^{\text{cut}} > 0$, Algorithm 4 outputs a $\hat{\mu}$ with*

$$|\hat{\mu} - \mu| \leq \frac{2}{\sqrt{\sum_{j\in[m]} \tilde{n}_j}} \sigma \sqrt{2\log\frac{2}{\delta}} + \frac{8\alpha m\sqrt{n^{\text{cut}}}}{\sum_{j\in[m]} \tilde{n}_j} \sigma \sqrt{2\log\frac{2m}{\delta}} + 5\epsilon \tag{31}$$

Proof. See proof in Section B.2. □

Proof of Theorem 3.2. Consider $\epsilon = 0$, i.e. no perturbation involved. By Lemma B.3 and Lemma B.4, with probability at least $1 - 2\delta$,

$$|\hat{\mu} - \mu| \leq \frac{2}{\sqrt{\sum_{j\in[m]} \tilde{n}_j}} \sigma \sqrt{2\log\frac{2}{\delta}} + \frac{8\alpha m\sqrt{n^{\text{cut}}}}{\sum_{j\in[m]} \tilde{n}_j} \sigma \sqrt{2\log\frac{2m}{\delta}} \tag{32}$$

□

B.1 Proof Of Lemma B.3

To prove Lemma B.3,

1. we first show that the perturbation changes the empirical mean of batches by at most ϵ;
2. we can show the concentration bound of empirical means and weighted means for the **unperturbed** samples;
3. we can conclude by using the two results above and triangular inequality.

The Probability Of Event $\bigcap_{j\in\mathcal{G}}\mathcal{E}_j$: For all $j \in \mathcal{G}$, let $\bar{x}_j$ be the empirical mean of **unperturbed** samples in batch j:

$$\bar{\mu}_j := \frac{1}{n_j}\sum_{i=1}^{n_j}\tilde{x}_j^i \tag{33}$$

By triangular inequality:

$$|\bar{\mu}_j - \hat{\mu}_j| = \left|\frac{1}{n_j}\sum_{i=1}^{n_j}(x_j^i - \tilde{x}_j^i)\right| \le \frac{1}{n_j}\sum_{i=1}^{n_j}\epsilon = \epsilon \tag{34}$$

Since $\mathcal{D}$ is sub-Gaussian distribution, we can show the concentration of unperturbed samples mean $\bar{\mu}_j$: for all good batch $j \in \mathcal{G}$,

$$\mathbb{P}\left(|\bar{\mu}_j - \mu| > t\right) \le 2\exp\left(-\frac{n_j t^2}{2\sigma^2}\right) \tag{35}$$

By union bound, with probability at least $1-\delta$, $\forall j \in \mathcal{G}$,

$$|\bar{\mu}_j - \mu| \le \frac{\sigma}{\sqrt{n_j}}\sqrt{2\log\frac{2|\mathcal{G}|}{\delta}} \le \frac{\sigma}{\sqrt{n_j}}\sqrt{2\log\frac{2m}{\delta}} \tag{36}$$

By triangular inequality, with probability at least $1-\delta$, $\forall j \in \mathcal{G}$,

$$|\hat{\mu}_j - \mu| \le |\hat{\mu}_j - \bar{\mu}_j| + |\bar{\mu}_j - \mu| \le \frac{\sigma}{\sqrt{n_j}}\sqrt{2\log\frac{2m}{\delta}} + \epsilon \tag{37}$$

I.e. $\mathbb{P}\left(\bigcap_{j\in\mathcal{G}}\mathcal{E}_j\right) \ge 1-\delta$.

The Probability Of Event $\mathcal{E}_{wa}$: We first show the weighted average of empirical mean of the **unperturbed** sample i.e., $\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j\in\mathcal{G}}\tilde{n}_j\bar{\mu}_j$ is a sub-Gaussian random variable: firstly, note that the mean of the weighted average is μ, i.e. $\mathbb{E}\left[\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j\in\mathcal{G}}\tilde{n}_j\bar{\mu}_j\right] = \mu$. By definition, we know for good batch $j \in \mathcal{G}$, $\tilde{x}_j^1, \ldots, \tilde{x}_j^{n_j}$ are i.i.d. sub-Gaussian random variable with mean μ and variance proxy σ^2, i.e.

$$\mathbb{E}\left[\exp\left(s\left(\tilde{x}_j^i - \mu\right)\right)\right] \le \exp\left(\frac{\sigma^2 s^2}{2}\right) \quad \forall s \in \mathbb{R}. \tag{38}$$

Since $\bar{\mu}_j = \frac{1}{n_j}\sum_{i=1}^{n_j}\tilde{x}_j^i$: for all $s \in \mathbb{R}$,

$$\mathbb{E}\left[\exp\left(s\left(\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j\in\mathcal{G}}\tilde{n}_j\bar{\mu}_j - \mu\right)\right)\right] = \prod_{j\in\mathcal{G}}\mathbb{E}\left[\exp\left(s\left(\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\tilde{n}_j(\bar{\mu}_j - \mu)\right)\right)\right] \tag{39}$$

$$= \prod_{j\in\mathcal{G}}\prod_{i\in[n_j]}\mathbb{E}\left[\exp\left(\frac{s}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\frac{\tilde{n}_j}{n_j}(\tilde{x}_j^i - \mu)\right)\right] \le \prod_{j\in\mathcal{G}}\prod_{i\in[n_j]}\exp\left(\frac{\sigma^2}{2}\left(\frac{s}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\frac{\tilde{n}_j}{n_j}\right)^2\right) \tag{40}$$

$$\le \exp\left(\frac{\sigma^2}{2}\left(\frac{s}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\right)^2\sum_{j\in\mathcal{G}}\sum_{i\in[n_j]}\left(\frac{\tilde{n}_j}{n_j}\right)^2\right) = \exp\left(\frac{\sigma^2}{2}\left(\frac{s}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\right)^2\sum_{j\in\mathcal{G}}\frac{\tilde{n}_j}{n_j}\tilde{n}_j\right) \tag{41}$$

$$\le \exp\left(\frac{\sigma^2}{2}\left(\frac{s}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\right)^2\sum_{j\in\mathcal{G}}\tilde{n}_j\right) = \exp\left(\frac{s^2}{2}\left(\frac{\sigma}{\sqrt{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}}\right)^2\right) \tag{42}$$

This means $\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}\bar{\mu}_j$ is a sub-Gaussian random variable with variance proxy $\frac{\sigma^2}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}$. Thus $\forall t > 0$,

$$\mathbb{P}\left(\left|\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j\in\mathcal{G}}\tilde{n}_j\bar{\mu}_j - \mu\right| > t\right) \le 2\exp\left(-\frac{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}t^2}{2\sigma^2}\right) \tag{43}$$

Thus with probability at least $1-\delta$:

$$\left|\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j\in\mathcal{G}}\tilde{n}_j\bar{\mu}_j-\mu\right|\le\frac{\sigma}{\sqrt{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}}\sqrt{2\log\frac{2}{\delta}} \tag{44}$$

This means:

$$\left|\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j\in\mathcal{G}}\tilde{n}_j\hat{\mu}_j-\mu\right| \tag{45}$$

$$\le\left|\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j\in\mathcal{G}}\tilde{n}_j\bar{\mu}_j-\mu\right|+\left|\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j\in\mathcal{G}}\tilde{n}_j\bar{\mu}_j-\frac{1}{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}\sum_{j\in\mathcal{G}}\tilde{n}_j\hat{\mu}_j\right| \tag{46}$$

$$\le\frac{\sigma}{\sqrt{\sum_{j'\in\mathcal{G}}\tilde{n}_{j'}}}\sqrt{2\log\frac{2}{\delta}}+\epsilon \tag{47}$$

I.e. $\mathbb{P}\left(\mathcal{E}_{wa}\right)\ge 1-\delta$.

By union bound $\mathbb{P}\left(\mathcal{E}_{conc}\right)=\mathbb{P}\left(\mathcal{E}_{wa}\cap\bigcap_{j\in\mathcal{G}}\mathcal{E}_j\right)\ge 1-2\delta$.

B.2 Proof Of Lemma B.4

By Lemma B.3, we know the weighted average of the empirical mean of good batches is a proper estimation for the population mean. Compared to $\mathcal{G}$, the C^* returned in Line 6 in Algorithm 4 may remove some good batches and include some bad batches. Even though, as long as we can show:

1. Line 6 will not remove too many good batches and will not include too many bad batches;
2. the bad batches included in C^* will not be significant

then we can show that the $\hat{x}$ returned in Line 8 is a reasonable estimation for μ.

The Structure Of C^*: C^* is the largest subset of batches with confidence interval intersection. The confidence intervals of all the good batches intersect under event $\bigcap_{j\in\mathcal{G}}\mathcal{E}_j$, thus C^* should be at least as large as $\mathcal{G}$, thus it is not possible to remove too many good batches. Furthermore, we can also show that we will significantly reduce the total number of samples. Later on, we can show that the statistical rate will not be affected too much. We make these ideas precise below.

Under event $\bigcap_{j\in\mathcal{G}}\mathcal{E}_j$,

$$\mu\in\bigcap_{j\in\mathcal{G}}I_j, \tag{48}$$

where I_j is the confidence interval defined in Line 5. Thus $\bigcap_{j\in\mathcal{G}}I_j\neq\emptyset$.

Because C^* maximizes

$$\max_{C\text{ s.t. }\emptyset\neq\bigcap_{j\in C}I_j}|C|, \tag{49}$$

we know $|C^*|\ge|\mathcal{G}|=(1-\alpha)m$. Furthermore, C^* can include at most αm batches, this means C^* includes at least $(1-2\alpha m)$ good batches. Formally:

$$|C^*\cap\mathcal{G}|=|C^*\setminus\mathcal{B}|\ge|C^*|-|\mathcal{B}|\ge(1-2\alpha)m. \tag{50}$$

Now we show C^* is not losing too much information, i.e. $\sum_{j\in C^*}\tilde{n}_j\ge\frac{1}{2}\sum_{j\in[m]}\tilde{n}_j$. By definition of n^{cut}, there are at least $2\alpha m+1$ batches in $[m]$ such that $\tilde{n}_j=n^{\text{cut}}$. Because C^* removes at more αm batches, there are at least $\alpha m+1$ batches in C^* such that $\tilde{n}_j=n^{\text{cut}}$. I.e.

$$\left|\left\{j\in C^*:\tilde{n}_j=n^{\text{cut}}\right\}\right|=\left|\left\{j\in[m]:\tilde{n}_j=n^{\text{cut}}\right\}\right|-\left|\left\{j\in[m]\setminus C^*:\tilde{n}_j=n^{\text{cut}}\right\}\right| \tag{51}$$

$$
\begin{aligned}
&\geq \left|\{j \in [m] : \tilde{n}_j = n^{\text{cut}}\}\right| - |[m] \setminus C^*| && (52)\\
&\geq 2\alpha m + 1 - \alpha m = \alpha m + 1 && (53)
\end{aligned}
$$

This means the information loss $\sum_{j\in[m]\setminus\mathcal{G}} \tilde{n}_j$ can be bounded by $\sum_{j\in C^*} \tilde{n}_j$, formally:

$$
\begin{aligned}
2\sum_{j\in C^*} \tilde{n}_j - \sum_{j\in[m]} \tilde{n}_j &= \sum_{j\in C^*} \tilde{n}_j + \sum_{j\in C^*} \tilde{n}_j - \sum_{j\in[m]\cap C^*} \tilde{n}_j - \sum_{j\in[m]\setminus C^*} \tilde{n}_j && (54)\\
&= \sum_{j\in C^*} \tilde{n}_j - \sum_{j\in[m]\setminus C^*} \tilde{n}_j \geq (\alpha m + 1)n^{\text{cut}} - \alpha m n^{\text{cut}} \geq 0 && (55)
\end{aligned}
$$

Thus we have:

$$
\sum_{j\in C^*} \tilde{n}_j \geq \frac{1}{2} \sum_{j\in[m]} \tilde{n}_j. \tag{56}
$$

Bad Batches In C^*: In order for a bad batch i to survive in C^*, its confidence interval I_i must intersect with each good batch's confidence interval in C^*. In particular, I_i must intersect with the good batch in C^* with the largest $\tilde{n}_j$. By definition, there are at least $\alpha m + 1$ good batches with $\tilde{n}_j = n^{\text{cut}}$. Because C^* excludes at most αm good batches, there is at least one good batch (denote by j^*), s.t. $\tilde{n}_{j^*} = n^{\text{cut}}$.

Thus $\forall j \in C^* \cap \mathcal{B}$, $I_i \cap I_{j^*} \neq \emptyset$. This means, there exists some point x, s.t. $x \in I_i \cap I_{j^*}$, thus

$$
\begin{aligned}
&|\hat{\mu}_i - \hat{\mu}_{j^*}| \leq |\hat{\mu}_i - x| + |x - \hat{\mu}_{j^*}| && (57)\\
\leq &\frac{\sigma}{\sqrt{n_i}}\sqrt{2\log\frac{2m}{\delta}} + \epsilon + \frac{\sigma}{\sqrt{n_{j^*}}}\sqrt{2\log\frac{2m}{\delta}} + \epsilon && (58)\\
\leq &\left(\frac{1}{\sqrt{\tilde{n}_i}} + \frac{1}{\sqrt{n^{\text{cut}}}}\right)\sigma\sqrt{2\log\frac{2m}{\delta}} + 2\epsilon. && (59)
\end{aligned}
$$

Furthermore, under event $\bigcap_{j\in\mathcal{G}} \mathcal{E}_j$,

$$
|\hat{\mu}_{j^*} - \mu| \leq \frac{\sigma}{\sqrt{n_{j^*}}}\sqrt{2\log\frac{2m}{\delta}} + \epsilon \leq \frac{\sigma}{\sqrt{n^{\text{cut}}}}\sqrt{2\log\frac{2m}{\delta}} + \epsilon \tag{60}
$$

By triangular inequality, $\hat{\mu}_i$ will not be too far away from μ:

$$
|\hat{\mu}_i - \mu| \leq |\hat{\mu}_i - \hat{\mu}_{j^*}| + |\hat{\mu}_{j^*} - \mu| = \left(\frac{1}{\sqrt{\tilde{n}_i}} + \frac{2}{\sqrt{n^{\text{cut}}}}\right)\sigma\sqrt{2\log\frac{2m}{\delta}} + 3\epsilon \tag{61}
$$

Error Decomposition: As mentioned earlier, we can decompose the estimation of $\hat{\mu}$ returned by Algorithm 4 by: statistical error (with potential information loss), term $\mathscr{A}_1$ in (66); error coming from including bad batches, term $\mathscr{A}_2$ in (66); error coming from removing good batches, term $\mathscr{A}_3$ in (66). Specifically:

$$
\begin{aligned}
&|\hat{\mu} - \mu| = \frac{1}{\sum_{j\in C^*} \tilde{n}_j}\left|\sum_{j\in C^*} \tilde{n}_j(\hat{\mu}_j - \mu)\right| && (62)\\
= &\frac{1}{\sum_{j\in C^*} \tilde{n}_j}\left|\left(\sum_{j\in\mathcal{G}} + \sum_{j\in C^*\cap\mathcal{B}} - \sum_{j\in\mathcal{G}\setminus C^*}\right)\tilde{n}_j(\hat{\mu}_j - \mu)\right| && (63)\\
\leq &\frac{1}{\sum_{j\in C^*} \tilde{n}_j}\left(\left|\sum_{j\in\mathcal{G}} \tilde{n}_j(\hat{\mu}_j - \mu)\right| + \left|\sum_{j\in C^*\cap\mathcal{B}} \tilde{n}_j(\hat{\mu}_j - \mu)\right| + \left|\sum_{j\in\mathcal{G}\setminus C^*} \tilde{n}_j(\hat{\mu}_j - \mu)\right|\right) && (64)\\
&\text{(this is by triangular inequality)} && (65)\\
=: &\mathscr{A}_1 + \mathscr{A}_2 + \mathscr{A}_3 && (66)
\end{aligned}
$$

We can bound the first term $\mathscr{A}_1$ by (56) under event $\mathcal{E}_{wa}$:

$$\mathscr{A}_1 = \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\left|\sum_{j\in\mathcal{G}}\tilde{n}_j(\hat{\mu}_j-\mu)\right| = \frac{\sum_{j\in\mathcal{G}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\left|\frac{1}{\sum_{j\in\mathcal{G}}\tilde{n}_j}\sum_{j\in\mathcal{G}}\tilde{n}_j(\hat{\mu}_j-\mu)\right| \tag{67}$$

$$= \frac{\sum_{j\in\mathcal{G}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\left|\frac{1}{\sum_{j\in\mathcal{G}}\tilde{n}_j}\sum_{j\in\mathcal{G}}\tilde{n}_j\hat{\mu}_j-\mu\right| \tag{68}$$

$$\le \frac{\sum_{j\in\mathcal{G}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\left(\frac{\sigma}{\sqrt{\sum_{j\in\mathcal{G}}\tilde{n}_j}}\sqrt{2\log\frac{2}{\delta}}+\epsilon\right) \quad \text{(By event } \mathcal{E}_{wa}\text{)} \tag{69}$$

$$= \frac{\sqrt{\sum_{j\in\mathcal{G}}\tilde{n}_j}}{\sum_{j\in C^*}\tilde{n}_j}\sigma\sqrt{2\log\frac{2}{\delta}} + \frac{\sum_{j\in\mathcal{G}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon \tag{70}$$

$$\le 2\frac{\sqrt{\sum_{j\in\mathcal{G}}\tilde{n}_j}}{\sum_{j\in[m]}\tilde{n}_j}\sigma\sqrt{2\log\frac{2}{\delta}} + \frac{\sum_{j\in\mathcal{G}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon \quad \text{(By (56))} \tag{71}$$

$$\le 2\frac{\sqrt{\sum_{j\in[m]}\tilde{n}_j}}{\sum_{j\in[m]}\tilde{n}_j}\sigma\sqrt{2\log\frac{2}{\delta}} + \frac{\sum_{j\in\mathcal{G}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon \quad \text{(By } \mathcal{G}\subseteq[m]\text{)} \tag{72}$$

$$= \frac{2}{\sqrt{\sum_{j\in[m]}\tilde{n}_j}}\sigma\sqrt{2\log\frac{2}{\delta}} + \frac{\sum_{j\in\mathcal{G}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon \tag{73}$$

By (61), we can bound the second term $\mathscr{A}_2$ by:

$$\mathscr{A}_2 = \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\left|\sum_{j\in C^*\cap\mathcal{B}}\tilde{n}_j(\hat{\mu}_j-\mu)\right| \le \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in C^*\cap\mathcal{B}}\tilde{n}_j\left|\hat{\mu}_j-\mu\right| \quad \text{(By triangular ineq)} \tag{74}$$

$$\le \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in C^*\cap\mathcal{B}}\tilde{n}_j\left(\left(\frac{1}{\sqrt{\tilde{n}_j}}+\frac{2}{\sqrt{n^{\text{cut}}}}\right)\sigma\sqrt{2\log\frac{2m}{\delta}}+3\epsilon\right) \quad \text{(By (61))} \tag{75}$$

$$\le \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in C^*\cap\mathcal{B}}\left(\sqrt{\tilde{n}_j}+\frac{2\tilde{n}_j}{\sqrt{n^{\text{cut}}}}\right)\sigma\sqrt{2\log\frac{2m}{\delta}} + \frac{\sum_{j\in C^*\cap\mathcal{B}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}3\epsilon \tag{76}$$

$$\le \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in C^*\cap\mathcal{B}}3\sqrt{n^{\text{cut}}}\sigma\sqrt{2\log\frac{2m}{\delta}} + \frac{\sum_{j\in C^*\cap\mathcal{B}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}3\epsilon \quad \text{(By } \tilde{n}_j\le n^{\text{cut}}\text{)} \tag{77}$$

$$\le \frac{3\alpha m\sqrt{n^{\text{cut}}}}{\sum_{j\in C^*}\tilde{n}_j}\sigma\sqrt{2\log\frac{2m}{\delta}} + \frac{\sum_{j\in C^*\cap\mathcal{B}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}3\epsilon \quad (C^* \text{ includes at most } \alpha m \text{ bad batches}) \tag{78}$$

We can bound the third term $\mathscr{A}_3$ by:

$$\mathscr{A}_3 = \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\left|\sum_{j\in\mathcal{G}\setminus C^*}\tilde{n}_j(\hat{\mu}_j-\mu)\right| \le \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in\mathcal{G}\setminus C^*}\tilde{n}_j\left|\hat{\mu}_j-\mu\right| \quad \text{(By triangular ineq)} \tag{79}$$

$$\le \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in\mathcal{G}\setminus C^*}\tilde{n}_j\left(\frac{\sigma}{\sqrt{n_j}}\sqrt{2\log\frac{2m}{\delta}}+\epsilon\right) \quad \left(\text{By event } \textstyle\bigcap_{j\in\mathcal{G}}\mathcal{E}_j\right) \tag{80}$$

$$\le \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in\mathcal{G}\setminus C^*}\tilde{n}_j\left(\frac{\sigma}{\sqrt{\tilde{n}_j}}\sqrt{2\log\frac{2m}{\delta}}+\epsilon\right) \tag{81}$$

$$= \frac{1}{\sum_{j\in C^*}\tilde{n}_j}\sum_{j\in\mathcal{G}\setminus C^*}\sigma\sqrt{\tilde{n}_j}\sqrt{2\log\frac{2m}{\delta}} + \frac{\sum_{j\in\mathcal{G}\setminus C^*}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon \tag{82}$$

$$\leq \frac{\alpha m\sqrt{n^{\text{cut}}}}{\sum_{j\in C^*}\tilde{n}_j}\sigma\sqrt{2\log\frac{2m}{\delta}} + \frac{\sum_{j\in\mathcal{G}\setminus C^*}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon \tag{83}$$

$$\text{(Because } C^* \text{ excludes at most } \alpha m \text{ good batches and } \tilde{n}_j \leq n^{\text{cut}}\text{)} \tag{84}$$

Note that the above upper bounds for $\mathscr{A}_2$ and $\mathscr{A}_3$ are still valid even if some of the $\tilde{n}_j$'s are zero.

In conclusion, we can bound the estimation error by:

$$|\hat{\mu} - \mu| \leq \mathscr{A}_1 + \mathscr{A}_2 + \mathscr{A}_3 \tag{85}$$

$$\leq \left(\frac{2}{\sqrt{\sum_{j\in[m]}\tilde{n}_j}}\sigma\sqrt{2\log\frac{2}{\delta}} + \frac{\sum_{j\in\mathcal{G}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon\right) \tag{86}$$

$$+ \left(\frac{3\alpha m\sqrt{n^{\text{cut}}}}{\sum_{j\in C^*}\tilde{n}_j}\sigma\sqrt{2\log\frac{2m}{\delta}} + \frac{\sum_{j\in C^*\cap\mathcal{B}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}3\epsilon\right) \tag{87}$$

$$+ \left(\frac{\alpha m\sqrt{n^{\text{cut}}}}{\sum_{j\in C^*}\tilde{n}_j}\sigma\sqrt{2\log\frac{2m}{\delta}} + \frac{\sum_{j\in\mathcal{G}\setminus C^*}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon\right) \tag{88}$$

$$= \frac{2}{\sqrt{\sum_{j\in[m]}\tilde{n}_j}}\sigma\sqrt{2\log\frac{2}{\delta}} + \frac{4\alpha m\sqrt{n^{\text{cut}}}}{\sum_{j\in C^*}\tilde{n}_j}\sigma\sqrt{2\log\frac{2m}{\delta}} \tag{89}$$

$$+ \frac{\left(\sum_{j\in\mathcal{G}} + \sum_{j\in C^*\cap\mathcal{B}}\right)\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon + \frac{\sum_{j\in C^*\cap\mathcal{B}}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}2\epsilon + \frac{\sum_{j\in\mathcal{G}\setminus C^*}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon \tag{90}$$

$$\leq \frac{2}{\sqrt{\sum_{j\in[m]}\tilde{n}_j}}\sigma\sqrt{2\log\frac{2}{\delta}} + \frac{4\alpha m\sqrt{n^{\text{cut}}}}{\sum_{j\in C^*}\tilde{n}_j}\sigma\sqrt{2\log\frac{2m}{\delta}} \tag{91}$$

$$+ \frac{\sum_{j\in[m]}\tilde{n}_j}{\sum_{j\in C^*}\tilde{n}_j}\epsilon + \frac{\alpha m n^{\text{cut}}}{\sum_{j\in C^*}\tilde{n}_j}2\epsilon + \frac{\alpha m n^{\text{cut}}}{\sum_{j\in C^*}\tilde{n}_j}\epsilon \tag{92}$$

$$\text{(By } \mathcal{G}\cup(C^*\cap\mathcal{B}) \subseteq [m],\ |C^*\cap\mathcal{B}| \leq \alpha m,\ |\mathcal{G}\setminus C^*| \leq \alpha m\text{)} \tag{93}$$

$$\leq \frac{2}{\sqrt{\sum_{j\in[m]}\tilde{n}_j}}\sigma\sqrt{2\log\frac{2}{\delta}} + \frac{8\alpha m\sqrt{n^{\text{cut}}}}{\sum_{j\in[m]}\tilde{n}_j}\sigma\sqrt{2\log\frac{2m}{\delta}} + 5\epsilon \tag{94}$$

$$\text{(By (53) and (56))} \tag{95}$$

C Proof Of Theorem 5.2

By following standard regret decomposition for UCB type of algorithm (see (Jin et al., 2020)), under the event that the estimation error of the Bellman operator is bounded by bonus terms, we can decompose the regret by:

1. the cumulative bonus term occurred in the trajectories of each good agent
2. a term that can be easier bounded by Azuma-Hoeffding's inequalities.

By Lemma B.4 and replacing Lemma B.3 with a variant for martingale, we can show the event mentioned above happens with high probability. Unlike standard regret bound for tabular settings, we cannot directly use the telescoping series to estimate the cumulative bonuses. Instead, we first need to show that because each good agent is using the same policy in every episode, their trajectories have a lot of overlaps, meaning the (s, a, h) counts of all good agents do not differ by too much. Given that, we can simplify the bound in Lemma B.4 and use the telescoping series.

We start by restating Theorem 5.2:

Theorem C.1 (Regret bound, Theorem 5.2). *If $\alpha \leq \frac{1}{3}\left(1 - \frac{1}{m}\right)$, for all $\delta < \frac{1}{4}$, with probability at least $1 - 3\delta$:*

$$\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\left(V_1^*(s_1) - V_1^{\hat{\pi}^k}(s_1)\right) = \tilde{O}\left((1 + \alpha\sqrt{m})SH^2\sqrt{AKm\log\frac{1}{\delta}}\right) \tag{96}$$

We first give the high-level idea of our proof:

1. We give an analysis under the intersection of three "good events":
   - event $\mathcal{E}$: the estimation error of Bellman operator is upper-bounded by bonus (See Section C.1, Lemma C.3);
   - event $\mathcal{E}_{\text{even}}$: if the total count $\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a,h)$ on some (s,a,h) is large, then the counts of each agent differ by at most 2 times (See Section C.3, Lemma C.9);
   - event $\mathcal{E}_{\text{Azmua}}$: an error term in the regret decomposition is bounded by Azmua-Hoeffding bound.
2. Under event $\mathcal{E}$, we can decompose the regret into two terms (see Section C.2, Lemma C.8):
   - a martingale with bounded difference which is controlled by Hoeffding bound under event $\mathcal{E}_{\text{Azmua}}$;
   - the cumulative bonus term, which can be bounded by telescoping series under event $\mathcal{E}_{\text{even}}$.

We use $\bar{Q}_h^k$, $\hat{Q}_h^k$, $\hat{\pi}_h^k$, $\hat{V}_h^k$, $\hat{\mathbb{B}}_h^k$, Γ_h^k to denote the variables used in the k-th episode. When synchronization happens in episode k, those variables are the updated ones after the synchronization; when there is no synchronization in episode k, those variables remain the same as in the last episode. Let $N_h^{j,k}(s,a)$ be the counts on (s,a,h) tuples in episode k after the counts update. Formally, We start by restating the data collection process and counts on (s,a,h) tuples of each good agent $j\in\mathcal{G}$: during the data collection process, we allow all of the agents to collect data together. In the k-th episode, agent j collects a multi-set of transition tuples using policy $d^{\hat{\pi}^k}$: $\left\{\left(s_h^{j,k}, a_h^{j,k}, r_h^{j,k}, s_{h+1}^{j,k}\right)\right\}_{h\in[H]}$.

$$D_{j,k} := \bigcup_{h\in[H]} D_{j,k}^h := \bigcup_{h\in[H]}\bigcup_{k'\le k}\left\{\left(s_h^{j,k'}, a_h^{j,k'}, r_h^{j,k'}, s_{h+1}^{j,k'}\right)\right\} \tag{97}$$

$N_h^{j,k}(s,a)$ is given by:

$$N_h^{j,k}(s,a) = \sum_{h=1}^{H}\sum_{(\tilde{s},\tilde{a},\tilde{r},\tilde{s}')\in D_{j,k}^h} \mathbf{1}\left\{(s,a)=(\tilde{s},\tilde{a})\right\} \tag{98}$$

We give the formal definition of good events below:

Definition C.2.

$$\mathcal{E}_{Azmua} := \left\{\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\sum_{h=1}^{H}\left(\mathbb{E}_{s'\sim P_h(\cdot|s_h^{j,k},a_h^{j,k})}\left[\hat{V}_{h+1}^k(s') - V_{h+1}^{\hat{\pi}^k}(s')\right]\right.\right. \tag{99}$$

$$\left.\left. - \left(\hat{V}_{h+1}^k(s_{h+1}^{j,k}) - V_{h+1}^{\hat{\pi}^k}(s_{h+1}^{j,k})\right)\right) \le \sqrt{8mKH^3\log\frac{2}{\delta}}\right\} \tag{100}$$

$$\mathcal{E} := \left\{\bigcap_{(s,a,h,k,f)\in\mathcal{S}\times\mathcal{A}\times H\times K\times[0,1]^{\mathcal{S}}}\left\{\left|\left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a)\right| \le \Gamma_h^k(s,a)\right\}\right\} \tag{101}$$

For any $(s,a,h,k)\in\mathcal{S}\times\mathcal{A}\times[H]\times[K]$, we define the following event:

$$\mathcal{E}_{even}(s,a,h,k) := \left\{\text{if } \textstyle\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) \ge 400m\log\frac{2mKSAH}{\delta}, \text{ then } \max_{i,j\in\mathcal{G}}\frac{N_h^{j,k}(s,a)}{N_h^{i,k}(s,a)} \le 2\right\} \tag{102}$$

And define

$$\mathcal{E}_{even} := \bigcap_{s,a,h,K}\mathcal{E}_{even}(s,a,h,k). \tag{103}$$

Proof of Theorem 5.2. By Azuma-Hoeffding inequality:

$$\mathbb{P}\left(\overline{\mathcal{E}_{\text{Azmua}}}\right) \le \delta \tag{104}$$

Then by union bound: Lemma C.3 and Lemma C.9 together implies for all $0<\delta<\frac{1}{4}$:

$$\mathbb{P}\left(\overline{\mathcal{E}}\cup\overline{\mathcal{E}_{\text{even}}}\cup\overline{\mathcal{E}_{\text{Azmua}}}\right) \le \mathbb{P}\left(\overline{\mathcal{E}}\right) + \mathbb{P}\left(\overline{\mathcal{E}_{\text{even}}}\right) + \mathbb{P}\left(\overline{\mathcal{E}_{\text{Azmua}}}\right) \le 3\delta \tag{105}$$

which means $\mathcal{E} \cap \mathcal{E}_{\text{even}} \cap \mathcal{E}_{\text{Azmua}}$ happens with probability at least $1 - 3\delta$.

We now upper bound the regret under event $\mathcal{E} \cap \mathcal{E}_{\text{even}} \cap \mathcal{E}_{\text{Azmua}}$. By Lemma C.8 we can decompose the regret by:

$$\sum_{k=1}^{K} \sum_{j \in \mathcal{G}} \left(V_1^*(s_1) - V_1^{\hat{\pi}^k}(s_1) \right) \tag{106}$$

$$\leq 2 \sum_{k=1}^{K} \sum_{j \in \mathcal{G}} \sum_{h=1}^{H} \Gamma_h^k(s_h^{j,k}, a_h^{j,k}) \tag{107}$$

$$+ \sum_{k=1}^{K} \sum_{j \in \mathcal{G}} \sum_{h=1}^{H} \left(\mathbb{E}_{s' \sim P_h(\cdot | s_h^k, a_h^k)} \left[\hat{V}_{h+1}^k(s') - V_{h+1}^{\hat{\pi}^k}(s') \right] - \left(\hat{V}_{h+1}^k(s_{h+1}^{j,k}) - V_{h+1}^{\hat{\pi}^k}(s_{h+1}^{j,k}) \right) \right) \tag{108}$$

$$(\text{Under event } \mathcal{E}) \tag{109}$$

$$\leq 2 \sum_{k=1}^{K} \sum_{j \in \mathcal{G}} \sum_{h=1}^{H} \Gamma_h^k(s_h^{j,k}, a_h^{j,k}) + \sqrt{8mKH^3 \log \frac{2}{\delta}} \tag{110}$$

$$\left(\text{Under event } \mathcal{E}_{\text{Azmua}}\right) \tag{111}$$

We only need to upper bound the cumulative bonus. Suppose the policy is updated at the beginning of $k_0 + 1, k_1 + 1, k_2 + 1, \ldots, k_l + 1$-th episodes, with the data collected in the first $k_0, k_1, k_2, \ldots, k_l$-th episodes, with $k_1 = 1$. To simplify the notation, we define $k_0 = 0$, $k_{l+1} = K$.

For convenience, in the following, we use $N_h^k(s, a)$ to denote the total count on (s, a, h) tuples up to episode k over all good agents:

$$N_h^k(s, a) := \sum_{j \in \mathcal{G}} N_h^{j,k}(s, a), \tag{112}$$

where $N_h^0(s, a) = 0$. We can rearrange the cumulative bonus by summing over (s, a) pairs:

$$\sum_{k=1}^{K} \sum_{j \in \mathcal{G}} \sum_{h=1}^{H} \Gamma_h^k(s_h^{j,k}, a_h^{j,k}) = \sum_{h=1}^{H} \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \sum_{t=1}^{l+1} \Gamma_h^{k_{t-1}+1}(s, a) \left(N_h^{k_t}(s, a) - N_h^{k_{t-1}}(s, a) \right) \tag{113}$$

When there are less than $(2\alpha m + 1)$ agents have coverage on some (s, a, h) tuple, the bonus term $\Gamma_h^k(s, a)$ is trivially set to be $H - h + 1$. In the following, we show that under the event $\mathcal{E}_{\text{even}}$, in (113), for each (s, a, h) tuple, there are at most $2N_0$ bonus term such that $\Gamma_h(s, a) = H - h + 1$, where

$$N_0 := 400m \log \frac{2mKSAH}{\delta}. \tag{114}$$

For any (s, a, h), let $l_0(s, a, h)$ be such that:

$$N_h^{k_{l_0(s,a,h)-1}}(s, a) < N_0 \leq N_h^{k_{l_0(s,a,h)}}(s, a). \tag{115}$$

This means when running the policy update at episode $k_{l_0(s,a,h)} + 1$, the total counts for (s, a, h), i.e. $N_h^{k_{l_0(s,a,h)}}(s, a)$, is larger than N_0. For any $k \geq k_{l_0(s,a,h)}$, we have

$$\sum_{j \in \mathcal{G}} N_h^{j,k}(s, a) = N_h^k(s, a) \geq N_h^{k_{l_0(s,a,h)}}(s, a) \geq N_0. \tag{116}$$

By definition of $\mathcal{E}_{\text{even}}$, for any $k \geq k_{l_0(s,a,h)}$

$$\max_{i,j \in \mathcal{G}} \frac{N_h^{j,k}(s, a)}{N_h^{i,k}(s, a)} \leq 2 \tag{117}$$

this means for any $k \geq k_{l_0(s,a,h)}$, $N_h^{j,k}(s, a) > 0, \forall j \in \mathcal{G}$, meaning all of the good agents have coverage on (s, a, h), this means there are at least $(1 - \alpha)m \geq 2\alpha m + 1$ agents have coverage, and thus:

- Trivial bonus can only happens at $k \leq k_{l_0(s,a,h)}$, i.e.

$$\Gamma_h^k(s,a) = H - h + 1 \text{ only if } k \leq k_{l_0(s,a,h)}. \tag{118}$$

Furthermore, in the algorithm, the agents synchronize and update their policy when or before any (s,a,h) counts for a good agent doubles. I.e.: for all $(s,a,h,j,i) \in \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{G} \times [l]$:

$$N_h^{k_t}(s,a) \leq 2N_h^{k_{t-1}}(s,a) \tag{119}$$

This means

$$N_h^{k_{l_0(s,a,h)}}(s,a) \leq 2N_h^{k_{l_0(s,a,h)-1}}(s,a) < 2N_0. \tag{120}$$

Thus for each (s,a,h) tuple, there are at most $2N_0$ bonus terms such that $\Gamma_h(s,a) = H - h + 1$.

- for any $k \geq k_{l_0(s,a,h)} + 1$

$$\Gamma_h^k(s,a) = \frac{6}{SAHKm} + \frac{2(H-h+1)}{\sqrt{\sum_{j\in[m]} \tilde{N}_h^{j,k-1}(s,a)}} \sqrt{2\log\frac{2(SAHKm)^{3S}}{\delta}} \tag{121}$$

$$+ \frac{8\alpha m\sqrt{N_h^{\text{cut},k-1}(s,a)}}{\sum_{j\in[m]} \tilde{N}_h^{j,k-1}(s,a)}(H-h+1)\sqrt{2\log\frac{2m(SAHKm)^{3S}}{\delta}} \tag{122}$$

Where $N_h^{\text{cut},k-1}(s,a)$ is the $(2\alpha m+1)$-largest among $\left\{N_h^{j,k-1}(s,a)\right\}$ and

$$\tilde{N}_h^{j,k-1}(s,a) = \max\left(N_h^{\text{cut},k-1}(s,a), N_h^{j,k-1}(s,a)\right); \tag{123}$$

For any $k-1 \geq k_{l_0(s,a,h)}$, $\max_{i,j\in\mathcal{G}} \frac{N_h^{j,k-1}(s,a)}{N_h^{i,k-1}(s,a)} \leq 2$ implies $\forall j$, $\tilde{N}_h^{j,k-1}(s,a) \geq \frac{1}{2}N_h^{j,k-1}(s,a)$ and $\tilde{N}_h^{j,k-1}(s,a) \geq \frac{1}{2}N_h^{\text{cut},k-1}(s,a)$.
This means for any $k \geq k_{l_0(s,a,h)} + 1$

$$\frac{1}{\sqrt{\sum_{j\in[m]} \tilde{N}_h^{j,k-1}(s,a)}} \leq \frac{\sqrt{2}}{\sqrt{\sum_{j\in[m]} N_h^{j,k-1}(s,a)}} = \frac{\sqrt{2}}{\sqrt{N_h^{k-1}(s,a)}} \tag{124}$$

$$\frac{m\sqrt{N_h^{\text{cut},k-1}(s,a)}}{\sum_{j\in[m]} \tilde{N}_h^{j,k-1}(s,a)} = \frac{\sqrt{m}\sqrt{\sum_{j\in[m]} N_h^{\text{cut},k-1}(s,a)}}{\sum_{j\in[m]} \tilde{N}_h^{j,k-1}(s,a)} \tag{125}$$

$$\leq \frac{\sqrt{m}\sqrt{2\sum_{j\in[m]} \tilde{N}_h^{j,k-1}(s,a)}}{\sum_{j\in[m]} \tilde{N}_h^{j,k-1}(s,a)} \leq \frac{2\sqrt{m}}{\sqrt{N_h^{k-1}(s,a)}} \tag{126}$$

Thus

$$\Gamma_h^k(s,a) \leq \frac{4 + 16\sqrt{2}\alpha\sqrt{m}}{\sqrt{N_h^{k-1}(s,a)}} H\sqrt{\log\frac{2m(SAHKm)^{3S}}{\delta}} + \frac{6}{SAHKm} \tag{127}$$

We are now ready to bound the cumulative bonus:

$$\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\sum_{h=1}^{H} \Gamma_h^k(s_h^{j,k}, a_h^{j,k}) = \sum_{h=1}^{H}\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\sum_{t=1}^{l+1} \Gamma_h^{k_{t-1}+1}(s,a)\left(N_h^{k_t}(s,a) - N_h^{k_{t-1}}(s,a)\right) \tag{128}$$

$$= \sum_{h=1}^{H}\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left(\sum_{t=1}^{l_0(s,a,h)} \Gamma_h^{k_{t-1}+1}(s,a)\left(N_h^{k_t}(s,a) - N_h^{k_{t-1}}(s,a)\right)\right. \tag{129}$$

$$+\sum_{t=l_0(s,a,h)+1}^{l+1}\Gamma_h^{k_{t-1}+1}(s,a)\left(N_h^{k_t}(s,a)-N_h^{k_{t-1}}(s,a)\right)\Bigg) \tag{130}$$

$$\leq\sum_{h=1}^{H}\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\left(\sum_{t=1}^{l_0(s,a,h)}\Gamma_h^{k_{t-1}+1}(s,a)\left(N_h^{k_t}(s,a)-N_h^{k_{t-1}}(s,a)\right)\right. \tag{131}$$

$$+\sum_{t=l_0(s,a,h)+1}^{l+1}\frac{4+16\sqrt{2}\alpha\sqrt{m}}{\sqrt{N_h^{k-1}(s,a)}}H\sqrt{\log\frac{2m(SAHKm)^{3S}}{\delta}}\left(N_h^{k_t}(s,a)-N_h^{k_{t-1}}(s,a)\right) \tag{132}$$

$$+\sum_{t=l_0(s,a,h)+1}^{l+1}\frac{6}{SAHKm}\left(N_h^{k_t}(s,a)-N_h^{k_{t-1}}(s,a)\right)\Bigg) \tag{133}$$

$$(\text{By (127)}) \tag{134}$$

$$=:\mathscr{A}_1+\mathscr{A}_2+\mathscr{A}_3. \tag{135}$$

By (118) and (120),

$$\mathscr{A}_1\leq SAH^2N_h^{k_{l_0(s,a,h)}}(s,a)\leq 2SAH^2N_0. \tag{136}$$

Because $k_{l+1}=K$,

$$\mathscr{A}_3\leq\frac{6}{SAHKm}\sum_{h=1}^{H}\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}N_h^K(s,a)=\frac{6}{SA} \tag{137}$$

By (119),

$$\sum_{t=l_0(s,a,h)+1}^{l+1}\frac{N_h^{k_t}(s,a)-N_h^{k_{t-1}}(s,a)}{\sqrt{N_h^{k_{t-1}}(s,a)}}\leq(\sqrt{2}+1)\sum_{t=l_0(s,a,h)+1}^{l+1}\frac{N_h^{k_t}(s,a)-N_h^{k_{t-1}}(s,a)}{\sqrt{N_h^{k_t}(s,a)}+\sqrt{N_h^{k_{t-1}}(s,a)}} \tag{138}$$

$$=(\sqrt{2}+1)\sum_{t=l_0(s,a,h)+1}^{l+1}\left(\sqrt{N_h^{k_t}(s,a)}-\sqrt{N_h^{k_{t-1}}(s,a)}\right)\leq(\sqrt{2}+1)\sqrt{N_h^K(s,a)} \tag{139}$$

By Cauchy–Schwarz inequality,

$$\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\sqrt{N_h^K(s,a)}\leq\sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}1\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}N_h^K(s,a)}=\sqrt{SAKm} \tag{140}$$

Thus

$$\mathscr{A}_2\leq(\sqrt{2}+1)(4+16\sqrt{2}\alpha\sqrt{m})H^2\sqrt{SAKm}\sqrt{\log\frac{2m(SAHKm)^{3S}}{\delta}} \tag{141}$$

$$=O\left((1+\alpha\sqrt{m})H^2S\sqrt{AKm}\sqrt{\log\frac{SAHKm}{\delta}}\right) \tag{142}$$

Thus

$$\mathscr{A}_1+\mathscr{A}_2+\mathscr{A}_3\leq O\left((1+\alpha\sqrt{m})H^2S\sqrt{AKm}\sqrt{\log\frac{SAHKm}{\delta}}\right) \tag{143}$$

$$+O\left(SAH^2m\log\frac{2mKSAH}{\delta}\right) \tag{144}$$

In conclusion:

$$\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\left(V_1^*(s_1)-V_1^{\hat{\pi}^k}(s_1)\right)\leq 2\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\sum_{h=1}^{H}\Gamma_h^k(s_h^{j,k},a_h^{j,k})+\sqrt{8mKH^3\log\frac{2}{\delta}} \tag{145}$$

$$=\tilde{O}\left((1+\alpha\sqrt{m})SH^2\sqrt{AKm\log\frac{1}{\delta}}\right) \tag{146}$$

□

C.1 The Good Event $\mathcal{E}$

We first show that our bonus is upper confidence bound for the estimated Bellman operator. Recall that our bonus term used in k-th episode is calculated based on the data collected in the first $k-1$-episodes. The bonus is given by:

- If $|j \in [m] : N_h^{j,k-1}(s,a) > 0| < 2\alpha m + 1$

$$\Gamma_h^k(s,a) = H - h + 1; \tag{147}$$

- If $|j \in [m] : N_h^{j,k-1}(s,a) > 0| \geq 2\alpha m + 1$

$$\Gamma_h^k(s,a) := \frac{6}{SAHKm} + \frac{2(H-h+1)}{\sqrt{\sum_{j\in[m]}\tilde{N}_h^{j,k-1}(s,a)}}\sqrt{2\log\frac{2(SAHKm)^{3S}}{\delta}} \tag{148}$$

$$+\frac{8\alpha m\sqrt{N_h^{\mathrm{cut},k-1}(s,a)}}{\sum_{j\in[m]}\tilde{N}_h^{j,k-1}(s,a)}(H-h+1)\sqrt{2\log\frac{2m(SAHKm)^{3S}}{\delta}} \tag{149}$$

Where $N_h^{\mathrm{cut},k-1}(s,a)$ is the $(2\alpha m+1)$-largest among $\left\{N_h^{j,k-1}(s,a)\right\}$ and

$$\tilde{N}_h^{j,k-1}(s,a) = \max\left(N_h^{\mathrm{cut},k-1}(s,a), N_h^{j,k-1}(s,a)\right). \tag{150}$$

To be precise:

Lemma C.3 (Valid bonus)**.** *Let $\mathcal{E}$ be the following event:*

$$\mathcal{E} = \left\{\bigcap_{(s,a,h,k,f)\in\mathcal{S}\times\mathcal{A}\times H\times K\times[0,1]^S}\left\{\left|\left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a)\right| \leq \Gamma_h^k(s,a)\right\}\right\} \tag{151}$$

Then, we have

$$\mathbb{P}\left(\mathcal{E}\right) \geq 1-\delta \tag{152}$$

To show that $\mathcal{E}$ is a high probability event, we seek to utilize the result of Theorem 3.2. Since there are two obstacles, we need to make some modifications:

1. Because the transition tuples are collected sequentially, they are no longer i.i.d., which means Lemma B.3 does not hold trivially. To resolve this, we use the concentration of martingale (see Lemma C.4);
2. Event $\mathcal{E}$ shows the concentration property of $\hat{\mathbb{B}}$ holds uniformly for infinitely many f's. Thus a direct union bound does not apply. Instead, we need to use a cover number argument for all possible f's, which is standard (see (Jin et al., 2020)).

Proof of Lemma C.3. Let $\mathcal{E}'$ be the following event:

$$\mathcal{E}' = \left\{N_h^{\mathrm{cut},k-1}(s,a) > 0\right\}. \tag{153}$$

In the following, we decompose $\mathcal{E}$ by:

$$\mathcal{E} = \left(\mathcal{E}\cap\overline{\mathcal{E}'}\right)\cup\left(\mathcal{E}\cap\mathcal{E}'\right) \tag{154}$$

and bound $\mathbb{P}\left(\mathcal{E}\right)$ by law of total probability.

If $N_h^{\mathrm{cut},k-1}(s,a) = 0$, because $\left(\hat{\mathbb{B}}_h^k f\right)(s,a) = 0$ and $\left(\mathbb{B}_h f\right)(s,a) \le H - h + 1$, with probability 1, $\forall (s,a,h,k,f) \in \mathcal{S} \times \mathcal{A} \times H \times K \times [0,1]^{\mathcal{S}}$,

$$\left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a) \right| \le \Gamma_h^k(s,a) \tag{155}$$

This means

$$\mathbb{P}\left(\mathcal{E} \cap \overline{\mathcal{E}'}\right) = \mathbb{P}\left(\mathcal{E} | \overline{\mathcal{E}'}\right) \mathbb{P}\left(\overline{\mathcal{E}'}\right) = \mathbb{P}\left(\overline{\mathcal{E}'}\right) \tag{156}$$

If $N_h^{\mathrm{cut},k-1}(s,a) > 0$, we use a covering number argument and union bound to bound the probability of event $\mathcal{E}$.

Consider $\mathcal{V}_\epsilon := \left\{ \frac{1}{\lceil 1/\epsilon \rceil}, \frac{2}{\lceil 1/\epsilon \rceil}, \ldots, \frac{H\lceil 1/\epsilon \rceil}{\lceil 1/\epsilon \rceil} \right\}^{\mathcal{S}}$, an ϵ cover of $[0,H]^{\mathcal{S}}$, in the sense of ∞-norm. We can bound the cover number by $|\mathcal{V}_\epsilon| \le \left(H\left(\frac{1}{\epsilon}+1\right)\right)^S$. This means $\forall f \in [0,H]^{\mathcal{S}}$, we can find an $V_f \in \mathcal{V}_\epsilon$, s.t. $\|f - V_f\|_\infty := \max_{x \in \mathcal{S}} |f(x) - V_f(x)| \le \epsilon$. In other words,

$$[0,H]^{\mathcal{S}} = \bigcup_{f_\epsilon \in \mathcal{V}_\epsilon} \left\{ f : \|f - f_\epsilon\|_\infty \le \epsilon \right\}. \tag{157}$$

Importantly, unlike the model-based method without bad agents, our $\hat{\mathbb{B}}$ is not a linear operator, meaning we cannot trivially upper bound $\left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\hat{\mathbb{B}}_h^k V_f\right)(s,a) \right|$ in the cover number argument. Instead, we need to use the continuity of error bound of our robust mean estimation Algorithm 4, meaning as long as each data point collected by each agent is not perturbed too much, then the estimation error bound does not increase too much.

Recall that in Algorithm 2, at episode k, if the agents decide to synchronize, then at each step h, given any function f, the clean agents will calculate the empirical mean for

$$\left\{ r + f(s') : (s,a,r,s') \in D_h^{j,k} \right\}. \tag{158}$$

Let f_ϵ be an element in $\mathcal{V}_\epsilon$, s.t. $\|f_\epsilon - f\|_\infty \le \epsilon$, this means set (158) is a perturbed version (by at most ϵ) of

$$\left\{ r + f_\epsilon(s') : (s,a,r,s') \in D_h^{j,k} \right\}. \tag{159}$$

This means given an $f_\epsilon \in \mathcal{V}_\epsilon$, for any f, s.t. $\|f - f_\epsilon\|_\infty \le \epsilon$, Algorithm 4 can be used to robustly estimate $\left(\mathbb{B}_h f_\epsilon\right)(s,a)$, given set (158). Furthermore, choosing $\epsilon = \frac{1}{SAHKm}$, by Lemma C.4, Lemma C.5 and Lemma B.4, given any s,a,h,k,f_ϵ, and any f, s.t. $\|f - f_\epsilon\|_\infty \le \epsilon$, with probability at least $1 - \frac{\delta}{(SAHKm)^{3S}/(2mK)}$,

$$\left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f_\epsilon\right)(s,a) \right| \le \Gamma_h^k(s,a) - \frac{1}{SAHKm}. \tag{160}$$

We can bound the $\left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a) \right|$ by:

$$\begin{aligned} \left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a) \right| &\le \left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f_\epsilon\right)(s,a) \right| + \left| \left(\mathbb{B}_h f_\epsilon\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a) \right| && (161) \\ &\le \left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f_\epsilon\right)(s,a) \right| + \frac{1}{SAHKm} && (162) \end{aligned}$$

Then

$$\begin{aligned} &\mathbb{P}\left(\bigcup_{s,a,h,k,f} \left\{ \left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a) \right| > \Gamma_h^k(s,a) \right\} \right) && (163) \\ &\le \sum_{s,a,h,k} \mathbb{P}\left(\bigcup_{f \in [0,H]^{\mathcal{S}}} \left\{ \left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a) \right| > \Gamma_h^k(s,a) \right\} \right) && (164) \\ &\le \sum_{s,a,h,k} \mathbb{P}\left(\bigcup_{f_\epsilon \in \mathcal{V}_\epsilon} \bigcup_{f : \|f - f_\epsilon\|_\infty \le \epsilon} \left\{ \left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f_\epsilon\right)(s,a) \right| + \frac{1}{SAHKm} > \Gamma_h^k(s,a) \right\} \right) && (165) \end{aligned}$$

$$\leq \sum_{s,a,h,k} \sum_{f_\epsilon \in \mathcal{V}_\epsilon} \mathbb{P}\left(\bigcup_{f: \|f - f_\epsilon\|_\infty \leq \epsilon} \left\{ \left| \left(\hat{\mathbb{B}}_h^k f\right)(s,a) - \left(\mathbb{B}_h f_\epsilon\right)(s,a) \right| + \frac{1}{SAHKm} > \Gamma_h^k(s,a) \right\} \right) \tag{166}$$

$$\leq SAHK(H(1+HSAKm))^S \frac{\delta}{(SAHKm)^{3S}/(2mK)} \leq \delta \tag{167}$$

This means

$$\mathbb{P}\left(\mathcal{E} \cap \mathcal{E}'\right) = \mathbb{P}\left(\mathcal{E} | \mathcal{E}'\right) \mathbb{P}\left(\mathcal{E}'\right) \geq (1-\delta)\, \mathbb{P}\left(\mathcal{E}'\right) \geq \mathbb{P}\left(\mathcal{E}'\right) - \delta \tag{168}$$

In conclusion,

$$\mathbb{P}\left(\mathcal{E}\right) = \mathbb{P}\left(\mathcal{E} \cap \mathcal{E}'\right) + \mathbb{P}\left(\mathcal{E} \cap \overline{\mathcal{E}'}\right) \geq \mathbb{P}\left(\mathcal{E}\right) + \mathbb{P}\left(\mathcal{E}'\right) - \delta = 1 - \delta. \tag{169}$$

□

C.1.1 Concentration Of Estimation From Good Agents

Lemma C.4. *Let:*

$$\left(\hat{\mathbb{B}}_h^{j,k} f\right)(s,a) := \frac{1}{N_h^{j,k}(s,a)} \sum_{(s,a,r,s') \in D_h^{j,k}} r + f(s'), \tag{170}$$

where we define $\frac{0}{0} = 0$. *For any* $f : \mathcal{S} \mapsto [H]$, *and for any* $(s,a,h,k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$ *with probability at least* $1 - \delta/2$, $\mathcal{E}_{conc-seq}(s,a,h,k)$ *happens, where*

$$\mathcal{E}_{conc-seq}(s,a,h,k) = \bigcap_{j \in \mathcal{G}} \mathcal{E}_{c-seq}(s,a,h,j,k), \tag{171}$$

and

$$\mathcal{E}_{c-seq}(s,a,h,j,k) := \left\{ \left| \left(\hat{\mathbb{B}}_h^{j,k} f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a) \right| \leq \frac{H-h+1}{\sqrt{\tilde{N}_h^{j,k}(s,a)}} \sqrt{2 \log \frac{4Km}{\delta}} \right\} \tag{172}$$

Proof of Lemma C.4. We use the martingale stopping time argument in Lemma 4.3 of (Jin et al., 2018).

For each fixed $(s,a,h,j) \in \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{G}$: for all $t \in [K]$, define

$$\mathcal{F}_t := \sigma\left(\bigcup_{t' \leq t} \bigcup_{j \in [m]} \left\{ \left(s_h^{j,t'}, a_h^{j,t'}, r_h^{j,t'}, s_{h+1}^{j,t'} \right) \right\}_{h=1}^H \right). \tag{173}$$

Let

$$X_t = \sum_{(s,a,r,s') \in D_h^{j,t}} \left(r + f(s') - \left(\mathbb{B}_h f\right)(s,a) \right) \tag{174}$$

Then $\{(\mathcal{F}_t, X_t)\}_{t=1}^K$ is a martingale. One observation is $X_{t_1} = X_{t_2}$ if agent j did not visit (s,a,h) in $t_1+1, t_1+2, \ldots, t_2$-th episodes. Thus we can use the stopping time idea to shorten the martingale sequence.

Define the following sequence of t_i's: $t_0 := 0$,

$$t_i := \min\left(\left\{ t' \in [K] : t' > t_{i-1} \text{ and } (s_h^{j,t'}, a_h^{j,t'}) = (s,a) \right\} \cup \{K+1\} \right). \tag{175}$$

Intuitively, t_i is the episode when (s,a,h) is visited by agent j for the i-th time. If agent j visit (s,a,h) for less than i times, then $t_i = K+1$. By definition, t_i is a stopping time w.r.t. $\{\mathcal{F}_t\}_{t=1}^K$.

By optional sampling theorem, $\{(\mathcal{F}_{t_i}, X_{t_i})\}_{i=1}^K$ is a martingale.

By Azuma-Hoeffding's inequality: for any $\tau \leq K$

$$\mathbb{P}\left(|X_{t_\tau}| \geq \beta\right) \leq 2 \exp\left(-\frac{2\beta^2}{4\tau(H-h+1)^2} \right) \tag{176}$$

Let $\frac{\delta}{2mK} = 2\exp\left(-\frac{2\beta^2}{4\tau(H-h+1)^2}\right)$, we get: for any (s,a,h,j), for any $\tau \le K$, with probability at least $1-\frac{\delta}{2mK}$:

$$\left|\sum_{(s,a,r,s')\in D_h^{j,t_\tau}} \left(r + f(s') - \left(\mathbb{B}_h f\right)(s,a)\right)\right| < \sqrt{\tau}(H-h+1)\sqrt{2\log\frac{4mK}{\delta}}. \tag{177}$$

By union bound, for any (s,a,h,j), with probability at least $1-\frac{\delta}{mK}$, for any $\tau \le K$:

$$\left|\sum_{(s,a,r,s')\in D_h^{j,t_\tau}} \left(r + f(s') - \left(\mathbb{B}_h f\right)(s,a)\right)\right| < \sqrt{\tau}(H-h+1)\sqrt{2\log\frac{4mK}{\delta}}. \tag{178}$$

This means for any (s,a,h,j,k) and any $\tau \le k$

$$\begin{aligned}
&\mathbb{P}\left(\overline{\mathcal{E}_{c-seq}(s,a,h,j,k)}\middle| N_h^{j,k}(s,a)=\tau\right) && (179)\\
\le&\mathbb{P}\left(\left|\left(\hat{\mathbb{B}}_h^{j,k} f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a)\right| \ge \frac{H-h+1}{\sqrt{N_h^{j,k}(s,a)}}\sqrt{2\log\frac{4Km}{\delta}}\middle| N_h^{j,k}(s,a)=\tau\right) && (180)\\
\le&\frac{\delta}{mK} && (181)
\end{aligned}$$

Thus

$$\begin{aligned}
\mathbb{P}\left(\overline{\mathcal{E}_{c-seq}(s,a,h,j,k)}\right) =& \sum_{\tau=0}^{k} \mathbb{P}\left(\overline{\mathcal{E}_{c-seq}(s,a,h,j,k)}\middle| N_h^{j,k}(s,a)=\tau\right)\mathbb{P}\left(N_h^{j,k}(s,a)=\tau\right) && (182)\\
\le&\frac{\delta}{2mK} && (183)
\end{aligned}$$

By union bound

$$\mathbb{P}\left(\mathcal{E}_{conc-seq}(s,a,h,k)\right) \ge 1-\frac{\delta}{2}. \tag{184}$$

□

Lemma C.5. *Let:*

$$\left(\hat{\mathbb{B}}_h^{j,k} f\right)(s,a) := \frac{1}{N_h^{j,k}(s,a)} \sum_{(s,a,r,s')\in D_h^{j,k}} r + f(s'), \tag{185}$$

$$\left(\hat{\mathbb{B}}_h^{\mathcal{G},k} f\right)(s,a) := \frac{1}{\sum_{j\in\mathcal{G}} \tilde{N}_h^{j,k}(s,a)} \sum_{j\in\mathcal{G}} \tilde{N}_h^{j,k}(s,a)\left(\hat{\mathbb{B}}_h^{j,k} f\right)(s,a), \tag{186}$$

where we define $\frac{0}{0}=0$*. For any* $f:\mathcal{S}\mapsto[H]$*, with probability at least* $1-\delta/2$*,* $\mathcal{E}_{ct}(s,a,h,k)$ *happens, where*

$$\mathcal{E}_{ct}(s,a,h,k) := \left\{\left|\left(\hat{\mathbb{B}}_h^{\mathcal{G},k} f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a)\right| \le \frac{H-h+1}{\sqrt{\sum_{j\in\mathcal{G}} \tilde{N}_h^{j,k}(s,a)}}\sqrt{2\log\frac{4mK}{\delta}}\right\} \tag{187}$$

Proof *Lemma C.5*. During the data-collecting process, the agents are allowed to collect data simultaneously. For analysis purposes, we artificially order the data in the following sequence:

$$E^{1,1}, E^{2,1}, \ldots, E^{m,1}, E^{1,2}, \ldots, E^{m,2}, \ldots, E^{1,K}, \ldots, E^{m,K} \tag{188}$$

where $E^{j,k} := \left\{\left(s_h^{j,k}, a_h^{j,k}, r_h^{j,k}, s_{h+1}^{j,k}\right)\right\}_{h=1}^{H}$. Let

$$\mathcal{F}_t = \sigma\left(\bigcup_{j,k \text{ s.t. } m(k-1)+j\le t} E^{j,k}\right). \tag{189}$$

Then $\{\mathcal{F}_t\}_{t=0}^{mK}$ forms a valid filtration. Let $\left\{\{\gamma_{j,k}\}_{j\in[m]}\right\}_{k\in[K]}$ be a fixed set of scalar, s.t. $0 \le \gamma_{j,k} \le 1$, for all j, k.

For each fixed $(s,a,h) \in \mathcal{S}\times\mathcal{A}\times[H]$: for all $t \in [mK]$, Let

$$X_t = \sum_{(s,a,r,s')\in\bigcup_{(j,k)\in\mathcal{G}\times[K] \text{ s.t. } m(k-1)+j\le t} E^{j,k}} \gamma_{j,k}\left(r + f(s') - \left(\mathbb{B}_h f\right)(s,a)\right) \tag{190}$$

Then $\{(\mathcal{F}_t, X_t)\}_{t=1}^{mK}$ is a martingale. As we can see, if good agent j did not visit (s,a,h) in episode k, then $X_{m(k-1)+j} = X_{m(k-1)+j-1}$ a.s. Thus we can use the stopping time idea to shorten the martingale sequence.

Define the following functions to map from sequence index to agent index and episode index:

$$\mathcal{J}(t) := t - m\left(\lceil t/m\rceil - 1\right), \quad \mathcal{K}(t) := \lceil t/m\rceil \tag{191}$$

For any $n_1, \ldots, n_m$, define the following sequence of t_i's: $t_0 := 0$,

$$t_i := \min\left(\left\{t' \in [mK] : t' > t_{i-1} \text{ and } (s_h^{\mathcal{J}(t'),\mathcal{K}(t')}, a_h^{\mathcal{J}(t'),\mathcal{K}(t')}) = (s,a)\right.\right. \tag{192}$$

$$\left.\left.\text{and for all } j \le \mathcal{J}(t'), N_h^{j,\mathcal{K}(t')} \le n_j; j > \mathcal{J}(t'), N_h^{j,\mathcal{K}(t')-1} \le n_j\right\} \cup \{K+1\}\right). \tag{193}$$

Intuitively, t_i is the episode when (s,a,h) is visited in sequence (188) for the i-th time. And for all j, agent j have not collected n_j (s,a,h) tuples. If (s,a,h) is visited for less than i times or there exists agent j visiting (s,a,h) more than n_j times, then $t_i = K+1$. By definition, t_i is a stopping time w.r.t. $\{\mathcal{F}_t\}_{t=1}^{mK}$.

In particular, let n_{cut} be the $(2\alpha m+1)$th-largest of all n_j's and $\tilde{n}_j = \min(n_{\text{cut}}, n_j)$. We choose $\gamma_{j,k} := \frac{\tilde{n}_j}{n_j} \le 1$.

By optional sampling theorem, $\{(\mathcal{F}_{t_i}, X_{t_i})\}_{i=1}^{mK}$ is a martingale.

By Azuma-Hoeffding's inequality: for any $\tau := \sum_{j\in[m]} n_j \le mK$

$$\mathbb{P}\left(|X_{t_\tau}| \ge \beta\right) \le 2\exp\left(-\frac{2\beta^2}{4(H-h+1)^2\sum_{t=1}^{\tau}\gamma_{\mathcal{J}(t),\mathcal{K}(t)}^2}\right) \tag{194}$$

Let $\frac{\delta}{2mK} = 2\exp\left(-\frac{2\beta^2}{4(H-h+1)^2\sum_{t=1}^{\tau}\gamma_{\mathcal{J}(t),\mathcal{K}(t)}^2}\right)$, we get: for any (s,a,h), for any $\tau \le mK$, with probability at least $1-\frac{\delta}{2mK}$:

$$|X_{t_\tau}| < \sqrt{\sum_{t=1}^{\tau}\gamma_{\mathcal{J}(t),\mathcal{K}(t)}^2}(H-h+1)\sqrt{2\log\frac{4mK}{\delta}}. \tag{195}$$

By union bound, for any (s,a,h), with probability at least $1-\frac{\delta}{2}$, for any $\tau \le mK$:

$$|X_{t_\tau}| < \sqrt{\sum_{t=1}^{\tau}\gamma_{\mathcal{J}(t),\mathcal{K}(t)}^2}(H-h+1)\sqrt{2\log\frac{4mK}{\delta}}. \tag{196}$$

This means for any $(s,a,h,k) \in \mathcal{S}\times\mathcal{A}\times[H]\times[K]$ and any $\tau \le mk$

$$\mathbb{P}\left(\overline{\mathcal{E}_{ct}(s,a,h,k)}\,\middle|\, N_h^{j,k}(s,a) = n_j, \forall j\right) \tag{197}$$

$$\le\mathbb{P}\left(\left|\left(\hat{\mathbb{B}}_h^{\mathcal{G},k} f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a)\right| \ge \frac{(H-h+1)\sqrt{\sum_{t=1}^{\tau}\frac{\tilde{N}_h^{\mathcal{J}(t),\mathcal{K}(t)}(s,a)}{N_h^{\mathcal{J}(t),\mathcal{K}(t)}(s,a)}}}{\sum_{j\in\mathcal{G}}\tilde{N}_h^{j,k}(s,a)}\right. \tag{198}$$

$$\left.\cdot\sqrt{2\log\frac{4SAHmK^2}{\delta}}\,\middle|\, N_h^{j,k}(s,a) = n_j, \forall j\right) \tag{199}$$

$$\leq \mathbb{P}\left(\left| \left(\hat{\mathbb{B}}_h^{\mathcal{G},k} f\right)(s,a) - \left(\mathbb{B}_h f\right)(s,a) \right| \geq \frac{(H-h+1)\sqrt{\sum_{t=1}^{\tau}\left(\frac{\tilde{N}_h^{\mathcal{J}(t),\mathcal{K}(t)}(s,a)}{N_h^{\mathcal{J}(t),\mathcal{K}(t)}(s,a)}\right)^2}}{\sum_{j\in\mathcal{G}} \tilde{N}_h^{j,k}(s,a)} \right. \tag{200}$$

$$\left. \cdot\sqrt{2\log\frac{4SAHmK^2}{\delta}} \right| N_h^{j,k}(s,a) = n_j, \forall j \Bigg) \tag{201}$$

$$\leq \frac{\delta}{2} \quad \left(\text{By } \gamma_{j,k} = \frac{\tilde{N}_h^{j,k}(s,a)}{N_h^{j,k}(s,a)}\right) \tag{202}$$

Thus

$$\mathbb{P}\left(\overline{\mathcal{E}_{ct}(s,a,h,k)}\right) = \sum_{(n_1,\ldots,n_m)\in[K]^m} \mathbb{P}\left(\overline{\mathcal{E}_{ct}(s,a,h,k)} \middle| N_h^{j,k}(s,a) = n_j, \forall j\right) \tag{203}$$

$$\mathbb{P}\left(N_h^{j,k}(s,a) = n_j, \forall j\right) \tag{204}$$

$$\leq \frac{\delta}{2} \tag{205}$$

□

C.2 The Regret Decomposition For UCB Style Algorithm

We follow the regret decomposition strategy in (Jin et al., 2020) under event $\mathcal{E}$, i.e. the estimation error for the Bellman operator is bounded by the bonus term.

The estimated Bellman operator can be used to approximate the Q function:

Lemma C.6. *Under event $\mathcal{E}$, for any $(s,a,h,k) \in \mathcal{S}\times\mathcal{A}\times H\times K$, and any policy π'*

$$\left| \left(\hat{\mathbb{B}}_h^k \hat{V}_{h+1}^k\right)(s,a) - Q_h^{\pi'}(s,a) - \mathbb{E}_{s'\sim P_h(\cdot|s,a)}\left[\hat{V}_{h+1}^k(s') - V_{h+1}^{\pi'}(s')\right] \right| \leq \Gamma_h^k(s,a) \tag{206}$$

Proof of Lemma C.6.

$$\left| \left(\hat{\mathbb{B}}_h^k \hat{V}_{h+1}^k\right)(s,a) - Q_h^{\pi'}(s,a) - \mathbb{E}_{s'\sim P_h(\cdot|s,a)}\left[\hat{V}_{h+1}^k(s') - V_{h+1}^{\pi'}(s')\right] \right| \tag{207}$$

$$\leq \left| \left(\hat{\mathbb{B}}_h^k \hat{V}_{h+1}^k\right)(s,a) - \left(\mathbb{B}_h \hat{V}_{h+1}^k\right)(s,a) \right| \tag{208}$$

$$+ \left| \left(\mathbb{B}_h \hat{V}_{h+1}^k\right)(s,a) - \left(\mathbb{B}_h V_{h+1}^{\pi'}\right)(s,a) - \mathbb{E}_{s'\sim P_h(\cdot|s,a)}\left[\hat{V}_{h+1}^k(s') - V_{h+1}^{\pi'}(s')\right] \right| \tag{209}$$

$$\left(\text{By triangular inequality and the fact that } \left(\mathbb{B}_h V_{h+1}^{\pi'}\right)(s,a) = Q_h^{\pi'}(s,a).\right) \tag{210}$$

$$\leq \Gamma_h^k(s,a) \tag{211}$$

$$(\text{We can bound the first term by the definition of event } \mathcal{E}, \tag{212}$$

$$\text{and the second term is zero by the definition of Bellman operator.}) \tag{213}$$

□

Under event $\mathcal{E}$ we can upper bound the value function and Q function of the optimal policy by the estimated value function and Q function of policy $\hat{\pi}^k$:

Lemma C.7 (Optimism)**.** *Under event $\mathcal{E}$, $\forall s,a,h,k$:*

$$\hat{Q}_h^k(s,a) \geq Q_h^*(s,a), \quad \hat{V}_h^k(s) \geq V_h^*(s) \tag{214}$$

Proof of Lemma C.7. We prove this by induction on h. Before that, note that, for any h,k,s, if

$$\hat{Q}_h^k(s,a) \geq Q_h^*(s,a), \quad \forall a \tag{215}$$

then because $\hat{\pi}^k$ is chosen by maximizing $\hat{Q}_h^k(s,a)$, we know

$$\hat{V}_h^k(s) = \max_a \hat{Q}_h^k(s,a) \geq \hat{Q}_h^k(s,\pi_h^*(a)) \geq Q_h^*(s,\pi_h^*(a)) = V_h^*(s) \tag{216}$$

This means for any h, k, s:

$$\left\{\forall a, \hat{Q}_h^k(s,a) \geq Q_h^*(s,a)\right\} \implies \left\{\hat{V}_h^k(s) \geq V_h^*(s)\right\} \tag{217}$$

We now begin our induction:

- For the base case, our goal is to show for any s, a, k, in the last step H,

$$\hat{Q}_H^k(s,a) \geq Q_H^*(s,a), \quad \hat{V}_H^k(s) \geq V_H^*(s) \tag{218}$$

  First note that $\hat{V}_{H+1} = V_{H+1}^* = 0$. By Lemma C.6 and choose $\pi' = \pi^*$,

$$\left|\left(\hat{\mathbb{B}}_H^k \hat{V}_{H+1}^k\right)(s,a) - Q_h^*(s,a)\right| \leq \Gamma_H^k(s,a) \tag{219}$$

  By definition of $\hat{Q}_H^k(s,a)$, and the fact that $Q_h^*(s,a)$ only contains the reward at step H, which is bounded by 1:

$$\hat{Q}_H^k(s,a) = \min\left(\left(\hat{\mathbb{B}}_H^k \hat{V}_{H+1}^k\right)(s,a) + \Gamma_H^k(s,a), 1\right) \geq Q_H^*(s,a) \tag{220}$$

  By (217), $\hat{V}_H^k(s) \geq V_H^*(s), \forall s$.

- Suppose for any s, a, k, the statement holds for step $h+1$, i.e.

$$\hat{Q}_{h+1}^k(s,a) \geq Q_{h+1}^*(s,a), \quad \hat{V}_{h+1}^k(s) \geq V_{h+1}^*(s) \tag{221}$$

  our goal is to show $\forall s, a, k$:

$$\hat{Q}_h^k(s,a) \geq Q_h^*(s,a), \quad \hat{V}_h^k(s) \geq V_h^*(s) \tag{222}$$

$$\begin{aligned}
&\left(\hat{\mathbb{B}}_h^k \hat{V}_{h+1}^k\right)(s,a) + \Gamma_h^k(s,a) && (223)\\
\geq &\left(\hat{\mathbb{B}}_h^k \hat{V}_{h+1}^k\right)(s,a) + \left|\left(\hat{\mathbb{B}}_h^k \hat{V}_{h+1}^k\right)(s,a) - Q_h^*(s,a) - \mathbb{E}_{s'\sim P_h(\cdot|s,a)}\left[\hat{V}_{h+1}^k(s') - V_{h+1}^*(s')\right]\right| && (224)\\
&(\text{By Lemma C.6 and let } \pi' = \pi^*) && (225)\\
\geq &Q_h^*(s,a) + \mathbb{E}_{s'\sim P_h(\cdot|s,a)}\left[\hat{V}_{h+1}^k(s') - V_{h+1}^*(s')\right] && (226)\\
&(\text{By triangular inequality}) && (227)\\
\geq &Q_h^*(s,a) && (228)\\
&\left(\forall s, \hat{V}_{h+1}^k(s') \geq V_{h+1}^*(s') \text{ by } (221)\right) && (229)
\end{aligned}$$

  By definition of Q function $Q_h^*(s,a) \leq H - h + 1$. Thus

$$\hat{Q}_h^k(s,a) = \min\left(\left(\hat{\mathbb{B}}_h^k \hat{V}_{h+1}^k\right)(s,a) + \Gamma_h^k(s,a), H - h + 1\right) \geq Q_h^*(s,a) \tag{230}$$

  By (217), $\hat{V}_h^k(s) \geq V_h^*(s), \forall s$.

□

We are now ready to prove the regret decomposition lemma:

Lemma C.8. *Under good event $\mathcal{E}$:*

$$\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\left(V_1^*(s_1)-V_1^{\hat{\pi}^k}(s_1)\right) \tag{231}$$
$$\leq 2\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\sum_{h=1}^{H}\Gamma_h^k(s_h^{j,k},a_h^{j,k}) \tag{232}$$
$$+\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\sum_{h=1}^{H}\left(\mathbb{E}_{s'\sim P_h(\cdot|s_h^k,a_h^k)}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right]-\left(\hat{V}_{h+1}^k(s_{h+1}^{j,k})-V_{h+1}^{\hat{\pi}^k}(s_{h+1}^{j,k})\right)\right) \tag{233}$$

Proof of Lemma C.8. We start by showing the decomposition of regret after step h in one episode of a single agent: by Lemma C.6 and Lemma C.7, under event $\mathcal{E}$, for any s,k,h

$$V_h^*(s)-V_h^{\hat{\pi}^k}(s)\leq \hat{V}_h^{\hat{\pi}^k}(s)-V_h^{\hat{\pi}^k}(s)\quad \text{(By Lemma C.7)} \tag{234}$$
$$=\hat{Q}_h^k(s,\hat{\pi}_h^k(s))-Q_h^{\hat{\pi}^k}(s,\hat{\pi}_h^k(s)) \tag{235}$$
$$\leq\left(\hat{\mathbb{B}}_h^k\hat{V}_{h+1}^k\right)(s,a)+\Gamma_h^k(s,a)-Q_h^{\hat{\pi}^k}(s,\hat{\pi}_h^k(s))\quad\left(\text{By definition of }\hat{Q}_h^k\right) \tag{236}$$
$$\leq\left|\left(\hat{\mathbb{B}}_h^k\hat{V}_{h+1}^k\right)(s,\hat{\pi}_h^k(s))-Q_h^{\hat{\pi}^k}(s,\hat{\pi}_h^k(s))-\mathbb{E}_{s'\sim P_h(\cdot|s,\hat{\pi}_h^k(s))}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right]\right| \tag{237}$$
$$+\left|Q_h^{\hat{\pi}^k}(s,\hat{\pi}_h^k(s))+\mathbb{E}_{s'\sim P_h(\cdot|s,\hat{\pi}_h^k(s))}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right]\right| \tag{238}$$
$$+\Gamma_h^k(s,a)-Q_h^{\hat{\pi}^k}(s,\hat{\pi}_h^k(s)) \tag{239}$$
$$\text{(By using triangular inequality on the first term)} \tag{240}$$
$$\leq\Gamma_h^k(s,\hat{\pi}_h^k(s))+Q_h^{\hat{\pi}^k}(s,\hat{\pi}_h^k(s))+\mathbb{E}_{s'\sim P_h(\cdot|s,\hat{\pi}_h^k(s))}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right] \tag{241}$$
$$+\Gamma_h^k(s,a)-Q_h^{\hat{\pi}^k}(s,\hat{\pi}_h^k(s)) \tag{242}$$
$$\big(\text{The first term is by using Lemma C.6 with }\pi'=\hat{\pi}^k, \tag{243}$$
$$\text{the term inside the absolute in the second is non-negative by Lemma C.7}\big) \tag{244}$$
$$=2\Gamma_h^k(s,\hat{\pi}_h^k(s))+\mathbb{E}_{s'\sim P_h(\cdot|s,\hat{\pi}_h^k(s))}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right] \tag{245}$$

This indeed gives a recursive formula: for any trajectory $\left\{(s_h^k,a_h^k,r_h^k,s_{h+1}^k)\right\}_{h\in[H]}$

$$\hat{V}_h^{\hat{\pi}^k}(s_h^k)-V_h^{\hat{\pi}^k}(s_h^k) \tag{246}$$
$$\leq 2\Gamma_h^k(s_h^k,\hat{\pi}_h^k(s_h^k))+\mathbb{E}_{s'\sim P_h(\cdot|s_h^k,\hat{\pi}_h^k(s_h^k))}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right] \tag{247}$$
$$=\hat{V}_{h+1}^k(s_h^k)-V_{h+1}^{\hat{\pi}^k}(s_h^k)+2\Gamma_h^k(s_h^k,\hat{\pi}_h^k(s_h^k)) \tag{248}$$
$$+\left(\mathbb{E}_{s'\sim P_h(\cdot|s_h^k,\hat{\pi}_h^k(s_h^k))}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right]-\left(\hat{V}_{h+1}^k(s_h^k)-V_{h+1}^{\hat{\pi}^k}(s_h^k)\right)\right) \tag{249}$$

Then, we can show the regret decomposition in one episode of a single agent by recursion:
for any trajectory $\left\{(s_h^k,a_h^k,r_h^k,s_{h+1}^k)\right\}_{h\in[H]}$ collected by a clean agent under policy $\hat{\pi}^k$:

$$V_1^*(s_1^k)-V_1^{\hat{\pi}^k}(s_1^k)\leq\hat{V}_1^k(s_1^k)-V_1^{\hat{\pi}^k}(s_1^k) \tag{250}$$
$$\leq\left(\hat{V}_2^k(s_2^k)-V_2^{\hat{\pi}^k}(s_2^k)\right)+2\Gamma_1^k(s_1^k,a_1^k) \tag{251}$$
$$+\left(\mathbb{E}_{s'\sim P_1(\cdot|s_1^k,a_1^k)}\left[\hat{V}_2^k(s')-V_2^{\hat{\pi}^k}(s')\right]-\left(\hat{V}_2^k(s_2^k)-V_2^{\hat{\pi}^k}(s_2^k)\right)\right) \tag{252}$$
$$\leq\left(\hat{V}_3^k(s_3^k)-V_3^{\hat{\pi}^k}(s_3^k)\right)+\sum_{h=1}^{2}2\Gamma_h^k(s_h^k,a_h^k) \tag{253}$$

$$+\sum_{h=1}^{2}\left(\mathbb{E}_{s'\sim P_h(\cdot|s_h^k,a_h^k)}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right]-\left(\hat{V}_{h+1}^k(s_{h+1}^k)-V_{h+1}^{\hat{\pi}^k}(s_{h+1}^k)\right)\right) \tag{254}$$

$$\leq\cdots \tag{255}$$

$$\leq\sum_{h=1}^{H}2\Gamma_h^k(s_h^k,a_h^k) \tag{256}$$

$$+\sum_{h=1}^{H}\left(\mathbb{E}_{s'\sim P_h(\cdot|s_h^k,a_h^k)}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right]-\left(\hat{V}_{h+1}^k(s_{h+1}^k)-V_{h+1}^{\hat{\pi}^k}(s_{h+1}^k)\right)\right) \tag{257}$$

Now we are ready to show the total regret decomposition. For each episode, we can make the regret decomposition w.r.t. any trajectory collected by a clean agent following policy $\hat{\pi}^k$. For convenience, we specialize the trajectories to be exactly the ones that are collected by the good agents and are used to calculate the bonus terms. The purpose is, in the future, when we bound the regret, we need to bound the cumulative bonus used in the trajectory. By decomposing the regret w.r.t. the trajectory collected in the algorithm, it is naturally guaranteed that the (s, a, h) tuples that are collected a lot by the good agents have a lower bonus. This is because, with more data collected, we can narrow down the confidence interval and design small but still valid bonus terms.

Because in our MDP definition, the MDP has a deterministic initial distribution, meaning the good agents always have the same starting state:

$$\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\left(V_1^*(s_1)-V_1^{\hat{\pi}^k}(s_1)\right)=\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\left(V_1^*(s_1^{j,k})-V_1^{\hat{\pi}^k}(s_1^{j,k})\right) \tag{258}$$

$$\leq 2\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\sum_{h=1}^{H}\Gamma_h^k(s_h^{j,k},a_h^{j,k}) \tag{259}$$

$$+\sum_{k=1}^{K}\sum_{j\in\mathcal{G}}\sum_{h=1}^{H}\left(\mathbb{E}_{s'\sim P_h(\cdot|s_h^{j,k},a_h^{j,k})}\left[\hat{V}_{h+1}^k(s')-V_{h+1}^{\hat{\pi}^k}(s')\right]-\left(\hat{V}_{h+1}^k(s_{h+1}^{j,k})-V_{h+1}^{\hat{\pi}^k}(s_{h+1}^{j,k})\right)\right) \tag{260}$$

□

C.3 Evenness Of Clean Agents

We need at least $(2\alpha m+1)$-agents to cover (s, a, h) in order to learn the Bellman operator properly. In this section, we show that the agents have "even" coverage on the visited (s, a, h) tuples in each (except a relatively small number) of the episodes. In the following we use $\tilde{m} := (1-\alpha)m = |\mathcal{G}|$ to denote the number of good agents.

Formally, we have:

Lemma C.9 (Even coverage of good agent). *For any $(s, a, h, k) \in \mathcal{S}\times\mathcal{A}\times[H]\times[K]$, we define the following event:*

$$\mathcal{E}_{even}(s,a,h,k) := \left\{\textit{if} \sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) \geq 400m\log\frac{2mKSAH}{\delta}, \textit{ then } \max_{i,j\in\mathcal{G}}\frac{N_h^{j,k}(s,a)}{N_h^{i,k}(s,a)}\leq 2\right\} \tag{261}$$

then, we have: for all $0<\delta<\frac{1}{4}$

$$\mathbb{P}\left(\bigcap_{(s,a,h,k)\in\mathcal{S}\times\mathcal{A}\times[H]\times[K]}\mathcal{E}_{even}(s,a,h,k)\right)\geq 1-2\delta \tag{262}$$

Remark C.10 (Intuition of the good event). *The event $\mathcal{E}_{even}(s, a, h, k)$ characterizes that: if in any episode k, a (s, a, h) tuple gets enough coverage from the clean agents, then the coverage of each agent are very close.*

See proof of Lemma C.9 in Section C.3.1.

C.3.1 Proof Of Lemma C.9

Proof of Lemma C.9 depends on the concentration of $N_h^{j,k}(s,a)$:

Lemma C.11 (Concentration of counts around empirical mean)**.** *For all* $0 < \delta < \frac{1}{4}$

$$\mathbb{P}\left(\bigcup_{s,a,h,k,j}\left\{\left|N_h^{j,k}(s,a) - \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right|\right.\right. \tag{263}$$

$$\left.\left. > \; 18\log\frac{2SAHmK}{\delta} + 4\sqrt{\log\frac{2SAHmK}{\delta}}\sqrt{\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)}\right\}\right) < 2\delta \tag{264}$$

Proof of Lemma C.11. See Section C.3.2. □

Proof of Lemma C.9. Let

$$N_0 := 400m\log\frac{2mKSAH}{\delta} \tag{265}$$

For any $(s,a,h,k) \in \mathcal{S}\times\mathcal{A}\times[H]\times[K]$, define events:

$$\mathcal{E}_1(s,a,h,k) := \left\{\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) \geq N_0\right\} \tag{266}$$

$$\mathcal{E}_2(s,a,h,k) := \left\{\max_{i,j\in\mathcal{G}}\frac{N_h^{j,k}(s,a)}{N_h^{i,k}(s,a)} \leq 2\right\} \tag{267}$$

Recall:

$$\mathcal{E}_{\text{even}}(s,a,h,k) := \left\{\text{if } \textstyle\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) \geq 400m\log\frac{2mKSAH}{\delta}\text{, then } \max_{i,j\in\mathcal{G}}\frac{N_h^{j,k}(s,a)}{N_h^{i,k}(s,a)} \leq 2\right\} \tag{268}$$

Then we can rewrite even $\mathcal{E}_{\text{even}}(s,a,h,k)$ as:

$$\mathcal{E}_{\text{even}}(s,a,h,k) = \overline{\mathcal{E}_1(s,a,h,k)} \cup \mathcal{E}_2(s,a,h,k) \tag{269}$$

We first show that if there are two $N_h^{j,k}$'s, whose ratio exceeds 2, then there must be some $N_h^{j,k}$ that deviates a lot from the empirical mean of $N_h^{j,k}$'s:

$$\overline{\mathcal{E}_2(s,a,h,k)} = \left\{\max_{i,j\in\mathcal{G}}\frac{N_h^{j,k}(s,a)}{N_h^{i,k}(s,a)} > 2\right\} \tag{270}$$

$$\subseteq \bigcup_{i\in\mathcal{G}}\left\{N_h^{i,k}(s,a) > \frac{498}{400}\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right\} \cup \bigcup_{i\in\mathcal{G}}\left\{N_h^{i,k}(s,a) < \frac{302}{400}\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right\} \tag{271}$$

$$= \bigcup_{i\in\mathcal{G}}\left\{N_h^{i,k}(s,a) - \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) > \frac{98}{400}\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right\} \tag{272}$$

$$\cup \bigcup_{i\in\mathcal{G}}\left\{N_h^{i,k}(s,a) - \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) < -\frac{98}{400}\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right\} \tag{273}$$

$$= \bigcup_{i\in\mathcal{G}}\left\{\left|N_h^{i,k}(s,a) - \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right| > \frac{98}{400}\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right\} \tag{274}$$

To show that $\mathcal{E}_{\text{even}}(s,a,h,k)$ happens w.h.p.:

$$\mathbb{P}\left(\bigcup_{s,a,h,k}\overline{\mathcal{E}_{\text{even}}(s,a,h,k)}\right) = \mathbb{P}\left(\bigcup_{s,a,h,k}\overline{\overline{\mathcal{E}_1(s,a,h,k)}\cup\mathcal{E}_2(s,a,h,k)}\right) \tag{275}$$

$$=\mathbb{P}\left(\bigcup_{s,a,h,k}\mathcal{E}_1(s,a,h,k)\cap\overline{\mathcal{E}_2(s,a,h,k)}\right) \tag{276}$$

$$\leq \mathbb{P}\left(\exists s,a,h,k, \sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) \geq N_0,\right. \tag{277}$$

$$\left. \exists i \in \mathcal{G}, \left| N_h^{i,k}(s,a) - \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right| > \frac{98}{400}\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right) \tag{278}$$

$$\text{(By (274))} \tag{279}$$

$$\leq \mathbb{P}\left(\exists s,a,h,k,i \left| N_h^{i,k}(s,a) - \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right|\right. \tag{280}$$

$$\left. > \frac{18}{400}\frac{1}{|\mathcal{G}|}N_0 + 4\sqrt{\frac{1}{400}\frac{1}{|\mathcal{G}|}N_0}\sqrt{\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)}\right) \tag{281}$$

$$= \mathbb{P}\left(\exists s,a,h,k,i \left| N_h^{i,k}(s,a) - \frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)\right|\right. \tag{282}$$

$$\left. > 18\log\frac{2mKSAH}{\delta} + 4\sqrt{\log\frac{2mKSAH}{\delta}}\sqrt{\frac{1}{|\mathcal{G}|}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)}\right) \tag{283}$$

$$< 2\delta \quad \text{(By Lemma C.11)} \tag{284}$$

□

C.3.2 Proof Of Lemma C.11

The high-level ideas are:

1. For each s, a, h,
   - for each $j \in \mathcal{G}$, define centered $N_h^{j,k}(s,a)$ as a martingale;
   - define centered $\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)$ as a martingale;
2. apply a modified Bernstein type of martingale concentration bound for both centered $N_h^{j,k}(s,a)$'s and centered $\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)$ (see Lemma C.12 and Lemma C.13);
3. because $N_h^{j,k}(s,a)$ and $\frac{1}{\bar{m}}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)$ have the same mean, we can use triangular inequality to show these two terms are close, and the distance is bounded by the variance term in Bernstein inequality.
4. Bernstein on $\frac{1}{\bar{m}}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)$ also allow us to bound its variance in terms of itself.
5. We can get our result by combining Step 3 and Step 4.

Lemma C.12 (Concentration of each $N_h^{j,k}(s,a)$)**.** *For all $0<\delta\leq 1/4$, with probability at least $1-\delta$, for all $(s,a,h,j,k)\in \mathcal{S}\times\mathcal{A}\times[H]\times\mathcal{G}\times[K]$:*

$$\left| N_h^{j,k}(s,a) - \sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\right| < 3\log\frac{2SAHmK}{\delta} + \sqrt{2\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\log\frac{2SAHmK}{\delta}} \tag{285}$$

Proof of Lemma C.12. See Section C.3.3 □

Lemma C.13 (Concentration of each $\frac{1}{\bar{m}}\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a)$)**.** *For all $0<\delta\leq 1/4$, with probability at least $1-\delta$, for all $(s,a,h,k)\in \mathcal{S}\times\mathcal{A}\times[H]\times[K]$:*

$$\left| \sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) - |\mathcal{G}|\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\right| < 3\log\frac{2SAHmK}{\delta} + \sqrt{2|\mathcal{G}|\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\log\frac{2SAHmK}{\delta}} \tag{286}$$

Proof of Lemma C.13. See Section C.3.4 □

Proof of Lemma C.11. Let $\mathcal{E}_N$ the intersection of the events in Lemma C.12 and Lemma C.13. Then by Lemma C.12 and Lemma C.13, $\mathcal{E}_N$ happens with probability at least $1 - 2\delta$. By (286),

$$\sqrt{\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)} \leq 4\sqrt{\log \frac{2SAHmK}{\delta}} + \sqrt{\frac{1}{|\mathcal{G}|}\sum_{j \in \mathcal{G}} N_h^{j,k}(s,a)} \tag{287}$$

By (285) and (286), for all s, a, h, j, k

$$\left|\frac{1}{|\mathcal{G}|}\sum_{j' \in \mathcal{G}} N_h^{j',k}(s,a) - N_h^{j,k}(s,a)\right| \tag{288}$$

$$\leq \left|N_h^{j,k}(s,a) - \sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\right| + \left|\frac{1}{|\mathcal{G}|}\sum_{j' \in \mathcal{G}} N_h^{j',k}(s,a) - \sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\right| \tag{289}$$

$$\leq 6\log\frac{2SAHmK}{\delta} + 2\sqrt{2\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\log\frac{2SAHmK}{\delta}} \tag{290}$$

$$\leq 6\log\frac{2SAHmK}{\delta} + 2\sqrt{2\log\frac{2SAHmK}{\delta}}\left(4\sqrt{\log\frac{2SAHmK}{\delta}} + \sqrt{\frac{1}{|\mathcal{G}|}\sum_{j \in \mathcal{G}} N_h^{j,k}(s,a)}\right) \tag{291}$$

$$\leq 18\log\frac{2SAHmK}{\delta} + 4\sqrt{\log\frac{2SAHmK}{\delta}}\sqrt{\frac{1}{|\mathcal{G}|}\sum_{j \in \mathcal{G}} N_h^{j,k}(s,a)} \tag{292}$$

□

C.3.3 Proof Of Lemma C.12

Proof of Lemma C.12. For each fixed $(s, a, h, j) \in \mathcal{S}\times\mathcal{A}\times[H]\times\mathcal{G}$: for all $t \in [K]$, define

$$\mathcal{F}_k := \sigma\left(\bigcup_{t \leq k}\bigcup_{j \in [m]}\left\{\left(s_h^{j,t}, a_h^{j,t}, r_h^{j,t}, s_{h+1}^{j,t}\right)\right\}_{h=1}^{H}\right). \tag{293}$$

Let

$$S_h^{j,k}(s,a) = N_h^{j,k}(s,a) - \sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a) \tag{294}$$

$$T_h^{j,k}(s,a) = \sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\left(1 - d_h^{\hat{\pi}^t}(s,a)\right) \tag{295}$$

Then $\left\{\left(\mathcal{F}_k, S_h^{j,k}(s,a)\right)\right\}_{t=k}^{K}$ is a martingale. Since $d_h^{\hat{\pi}^k}(s,a)$ depends on $\hat{\pi}^k$, which is calculated use data in the first $k-1$ episodes, then $d_h^{\hat{\pi}^k}(s,a) \in \mathcal{F}_{k-1}$. By Corollary E.3,

$$\mathbb{P}\left(\bigcup_{k=1}^{K}\left\{|S_h^{j,k}(s,a)| \geq 3\log\frac{2SAHmK}{\delta} + \sqrt{2\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\log\frac{2SAHmK}{\delta}}\right\}\right) \tag{296}$$

$$\leq \mathbb{P}\left(\bigcup_{k=1}^{K}\left\{|S_h^{j,k}(s,a)| \geq 3\log\frac{2SAHmK}{\delta} + \sqrt{2T_h^{j,k}(s,a)\log\frac{2SAHmK}{\delta}}\right\}\right) \leq \frac{\delta}{SAHm} \tag{297}$$

By union bound, with probability at least $1-\delta$, for all $(s,a,h,j,k) \in \mathcal{S}\times\mathcal{A}\times[H]\times\mathcal{G}\times[K]$:

$$|S_h^{j,k}(s,a)| < 3\log\frac{2SAHmK}{\delta} + \sqrt{2\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\log\frac{2SAHmK}{\delta}} \tag{298}$$

□

C.3.4 Proof Of Lemma C.13

Proof of Lemma C.13. During the data-collecting process, the agents are allowed to collect data simultaneously. For analysis purposes, we artificially order the data in the following sequence:

$$E^{1,1}, E^{2,1}, \ldots, E^{m,1}, E^{1,2}, \ldots, E^{m,2}, \ldots, E^{1,K}, \ldots, E^{m,K} \tag{299}$$

where $E^{j,k} := \left\{\left(s_h^{j,k}, a_h^{j,k}, r_h^{j,k}, s_{h+1}^{j,k}\right)\right\}_{h=1}^{H}$. Let

$$\mathcal{F}_t = \sigma\left(\bigcup_{j,k \text{ s.t. } m(k-1)+j\le t} E^{j,k}\right). \tag{300}$$

Then $\{\mathcal{F}_t\}_{t=0}^{mK}$ forms a valid filtration. Define the following functions to map from sequence index to agent index and episode index:

$$\mathcal{J}(t) := t - m\left(\lceil t/m\rceil - 1\right), \quad \mathcal{K}(t) := \lceil t/m\rceil \tag{301}$$

For each fixed $(s,a,h) \in \mathcal{S}\times\mathcal{A}\times[H]$, for all $t \in [mK]$, we define $S_h^{\mathcal{G},t}(s,a)$ as the (centered) total counts of (s,a,h) collected by all good agents up to time t. The t-th term in (299) could be in the center of an episode, meaning some agents have not collected their trajectories yet. So we need to treat the agents differently: Let

$$S_h^{\mathcal{G},t}(s,a) = \sum_{j\in\mathcal{G}, j\le\mathcal{J}(t)}\left(N_h^{j,\mathcal{K}(t)}(s,a) - \sum_{t=1}^{\mathcal{K}(t)} d_h^{\hat{\pi}^t}(s,a)\right) \tag{302}$$

$$+ \sum_{j\in\mathcal{G}, j>\mathcal{J}(t)}\left(N_h^{j,\mathcal{K}(t)-1}(s,a) - \sum_{t=1}^{\mathcal{K}(t)-1} d_h^{\hat{\pi}^t}(s,a)\right) \tag{303}$$

Then $\left\{\left(\mathcal{F}_t, S_h^{\mathcal{G},t}(s,a)\right)\right\}_{t=1}^{mK}$ is a martingale. Similar to Lemma C.12, define

$$T_h^{\mathcal{G},t}(s,a) = \sum_{j\in\mathcal{G}, j\le\mathcal{J}(t)}\sum_{t=1}^{\mathcal{K}(t)} d_h^{\hat{\pi}^t}(s,a)\left(1 - d_h^{\hat{\pi}^t}(s,a)\right) \tag{304}$$

$$+ \sum_{j\in\mathcal{G}, j>\mathcal{J}(t)}\sum_{t=1}^{\mathcal{K}(t)-1} d_h^{\hat{\pi}^t}(s,a)\left(1 - d_h^{\hat{\pi}^t}(s,a)\right) \tag{305}$$

Then by Corollary E.3,

$$\mathbb{P}\left(\bigcup_{k\in[K]}\left\{\left|\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) - |\mathcal{G}|\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\right| \ge 3\log\frac{2SAHmK}{\delta}\right.\right. \tag{306}$$

$$\left.\left. + \sqrt{2|\mathcal{G}|\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\log\frac{2SAHmK}{\delta}}\right\}\right) \tag{307}$$

$$\leq \mathbb{P}\left(\bigcup_{k\in[K]}\left\{\left|\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) - |\mathcal{G}|\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\right| \geq 3\log\frac{2SAHmK}{\delta}\right.\right. \tag{308}$$

$$\left.\left. + \sqrt{2|\mathcal{G}|\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\left(1 - d_h^{\hat{\pi}^t}(s,a)\right)\log\frac{2SAHmK}{\delta}}\right\}\right) \tag{309}$$

$$\leq \mathbb{P}\left(\bigcup_{k=1}^{mK}\left\{|S_h^{\mathcal{G},k}(s,a)| \geq 3\log\frac{2SAHmK}{\delta} + \sqrt{2T_h^{\mathcal{G},k}(s,a)\log\frac{2SAHmK}{\delta}}\right\}\right) \tag{310}$$

$$\leq \frac{\delta}{SAH} \tag{311}$$

By union bound, with probability at least $1-\delta$, for all $(s,a,h,k) \in \mathcal{S}\times\mathcal{A}\times[H]\times[K]$:

$$\left|\sum_{j\in\mathcal{G}} N_h^{j,k}(s,a) - |\mathcal{G}|\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\right| < 3\log\frac{2SAHmK}{\delta} + \sqrt{2|\mathcal{G}|\sum_{t=1}^{k} d_h^{\hat{\pi}^t}(s,a)\log\frac{2SAHmK}{\delta}} \tag{312}$$

□

D Proof Of Theorem 6.5

By the following lemma, we can upper bound the suboptimality by the cumulative bonuses:

Lemma D.1. *[Suboptimality for Pessimistic Value Iteration, Lemma 3.2 in (Zhang et al., 2021a) and Theorem 4.2 in (Jin et al., 2021)] Under the event $\mathcal{E}$ that the $\Gamma_h(s,a)$ satisfies the required property of bounding the Bellman error, i.e. $|\hat{Q}_h(s,a) - (\mathbb{B}\hat{V}_{h+1})(s,a)| \leq \Gamma_h(s,a), \forall h \in [H], (s,a) \in \mathcal{S}\times\mathcal{A}$ then against any comparator policy $\tilde{\pi}$, it achieves*

$$\mathrm{SubOpt}(\hat{\pi}, \tilde{\pi}) \leq 2\sum_{h=1}^{H} \mathbb{E}_{d^{\tilde{\pi}}}[\Gamma_h(s_h, a_h)] \tag{313}$$

Recall that for all $(s,a,h) \in \mathcal{S}\times\mathcal{A}\times[H]$,

$$N_h^j(s,a) := \sum_{k\in[K_j]} \mathbf{1}\left\{(s_h^{j,k}, a_h^{j,k}) = (s,a)\right\}, \quad \forall j \in [m]. \tag{314}$$

and $N_h^{\mathrm{cut}}(s,a)$ is the $(2\alpha m+1)$-largest among $\left\{N_h^j(s,a)\right\}_{j\in[m]}$. $N_h^{\mathcal{G},\mathrm{cut}_1}(s,a)$ is the $(\alpha m+1)$-th largest of $\left\{N_h^j(s,a)\right\}_{j\in\mathcal{G}}$ and $N_h^{\mathcal{G},\mathrm{cut}_2}(s,a)$ is the $(2\alpha m+1)$-th largest of $\left\{N_h^j(s,a)\right\}_{j\in\mathcal{G}}$. The bonuses are given by:

- If $N_h^{\mathrm{cut}}(s,a) = 0$

$$\Gamma_h(s,a) = H - h + 1; \tag{315}$$

- If $N_h^{\mathrm{cut}}(s,a) > 0$

$$\Gamma_h(s,a) := \frac{2(H-h+1)}{\sqrt{\sum_{j\in[m]} \tilde{N}_h^j(s,a)}}\sqrt{2\log\frac{2SAH}{\delta}} \tag{316}$$

$$+ \frac{8\alpha m\sqrt{N_h^{\mathrm{cut}}(s,a)}}{\sum_{j\in[m]} \tilde{N}_h^j(s,a)}(H-h+1)\sqrt{2\log\frac{2mSAH}{\delta}} \tag{317}$$

Where

$$\tilde{N}_h^j(s,a) = \max\left(N_h^{\mathrm{cut}}(s,a), N_h^j(s,a)\right). \tag{318}$$

Proof of Theorem 6.5. We first show that with probability at least $1-\delta$,

$$|(\hat{\mathbb{B}}_h\hat{V}_{h+1})(s,a)-(\mathbb{B}_h\hat{V}_{h+1})(s,a)|\leq\Gamma_h(s,a),\quad\forall(s,a)\in\mathcal{S}\times\mathcal{A},\forall h\in[H]\tag{319}$$

where $\Gamma_h(s,a)$ is defined in (313).

- if $N_h^{\text{cut}}(s,a)=0$, by definition, $(\hat{\mathbb{B}}_h\hat{V}_{h+1})(s,a)=0$. By definition of $\hat{V}_h$ and $\mathbb{B}_h$, $(\mathbb{B}_h\hat{V}_{h+1})(s,a)\in[0,H-h+1]$, thus (319) holds;
- if $N_h^{\text{cut}}(s,a)>0$, for any fixed $h\in[H]$, $(s,a)\in\mathcal{S}\times\mathcal{A}$, $f:\mathcal{S}\to[0,H]$. Because $(\hat{\mathbb{B}}_h f)(s,a)$ is bounded and thus sub-Gaussian, we can use Theorem 3.2 to upper bound $|(\hat{\mathbb{B}}_h f)(s,a)-(\mathbb{B}_h f)(s,a)|$:

$$\mathbb{P}\left(\left|(\hat{\mathbb{B}}_h f)(s,a)-(\mathbb{B}_h f)(s,a)\right|\geq\Gamma_h(s,a)\right)\leq\frac{\delta}{HSA}\tag{320}$$

Thus

$$\begin{aligned}
&\mathbb{P}\left(|(\hat{\mathbb{B}}_h\hat{V}_{h+1})(s,a)-(\mathbb{B}_h\hat{V}_{h+1})(s,a)|\geq\Gamma_h(s,a)\right) &(321)\\
=&\int_{[0,H]^{\mathcal{S}}}\mathbb{P}\left(|(\hat{\mathbb{B}}_h\hat{V}_{h+1})(s,a)-(\mathbb{B}_h\hat{V}_{h+1})(s,a)|\geq\Gamma_h(s,a)\middle|\hat{V}_{h+1}(\cdot)\right)d\mathbb{P}(\hat{V}_{h+1}(\cdot)) &(322)\\
\leq&\frac{\delta}{HSA} &(323)
\end{aligned}$$

By union bound, with probability at least $1-\delta$,

$$|(\hat{\mathbb{B}}_h\hat{V}_{h+1})(s,a)-(\mathbb{B}_h\hat{V}_{h+1})(s,a)|\leq\Gamma_h(s,a),\quad\forall(s,a)\in\mathcal{S}\times\mathcal{A},\forall h\in[H]\tag{324}$$

Then, by Lemma D.1, with probability at least $1-\delta$,

$$\begin{aligned}
\text{SubOpt}(\hat{\pi},\tilde{\pi})\leq&2\sum_{h=1}^H\mathbb{E}_{d^{\tilde{\pi}}}[\Gamma_h(s_h,a_h)] &(325)\\
=&2\sum_{h=1}^H\mathbb{E}_{d^{\tilde{\pi}}}\left[\Gamma_h(s_h,a_h)\mathbf{1}\left\{N_h^{\mathcal{G},\text{cut}_2}(s_h,a_h)=0\right\}\right] &(326)\\
&+2\sum_{h=1}^H\mathbb{E}_{d^{\tilde{\pi}}}\left[\Gamma_h(s_h,a_h)\mathbf{1}\left\{N_h^{\mathcal{G},\text{cut}_2}(s_h,a_h)>0\right\}\right] &(327)\\
=:&\mathscr{A}_1+\mathscr{A}_2. &(328)
\end{aligned}$$

By definition of $p^{\mathcal{G},0}$ in Definition 6.2,

$$\mathscr{A}_1\leq 2Hp^{\mathcal{G},0}\tag{329}$$

$$\begin{aligned}
\mathscr{A}_2=&2\sum_{h=1}^H\mathbb{E}_{d^{\tilde{\pi}}}\left[\Gamma_h(s_h,a_h)\mathbf{1}\left\{N_h^{\mathcal{G},\text{cut}_2}(s_h,a_h)>0\right\}\right] &(330)\\
\leq&2\sum_{h=1}^H\mathbb{E}_{d^{\tilde{\pi}}}\left[\left(\frac{2(H-h+1)}{\sqrt{\sum_{j\in\mathcal{G}}\tilde{N}_h^j(s,a)}}\sqrt{2\log\frac{2SAH}{\delta}}\right.\right. &(331)\\
&\left.\left.+\frac{8\alpha m\sqrt{N_h^{\text{cut}}(s,a)}}{\sum_{j\in\mathcal{G}}\tilde{N}_h^j(s,a)}(H-h+1)\sqrt{2\log\frac{2mSAH}{\delta}}\right)\mathbf{1}\left\{N_h^{\mathcal{G},\text{cut}_2}(s_h,a_h)>0\right\}\right]. &(332)
\end{aligned}$$

By the definition of κ_{even} in Definition 6.4: for $a=\tilde{\pi}(s)$,

$$\frac{1}{\sqrt{\sum_{j\in\mathcal{G}}\tilde{N}_h^j(s,a)}}=\frac{\sqrt{\sum_{j\in\mathcal{G}}N_h^j(s,a)}}{\sqrt{\sum_{j\in\mathcal{G}}\tilde{N}_h^j(s,a)}}\frac{1}{\sqrt{\sum_{j\in\mathcal{G}}N_h^j(s,a)}}\tag{333}$$

$$\leq \frac{\sqrt{\sum_{j\in\mathcal{G}} N_h^j(s,a)}}{\sqrt{\sum_{j\in\mathcal{G}} \tilde{N}_h^{j,\mathrm{cut}_2}(s,a)}} \frac{1}{\sqrt{\sum_{j\in\mathcal{G}} N_h^j(s,a)}} \tag{334}$$

$$\leq \frac{\sqrt{\kappa_{\text{even}}}}{\sqrt{\sum_{j\in\mathcal{G}} N_h^j(s,a)}} \tag{335}$$

and

$$\frac{m\sqrt{N_h^{\mathrm{cut}}(s,a)}}{\sum_{j\in\mathcal{G}} \tilde{N}_h^j(s,a)} \leq \frac{1}{\sqrt{1-\alpha}} \sqrt{\frac{\sum_{j\in\mathcal{G}} N_h^j(s,a)}{\sum_{j\in\mathcal{G}} \tilde{N}_h^j(s,a)} \frac{m(1-\alpha)N_h^{\mathrm{cut}}(s,a)}{\sum_{j\in\mathcal{G}} \tilde{N}_h^j(s,a)}} \frac{\sqrt{m}}{\sqrt{\sum_{j\in\mathcal{G}} N_h^j(s,a)}} \tag{336}$$

$$\leq \sqrt{\frac{\sum_{j\in\mathcal{G}} N_h^j(s,a)}{\sum_{j\in\mathcal{G}} \tilde{N}_h^{j,\mathrm{cut}_2}(s,a)} \frac{m(1-\alpha)N_h^{\mathcal{G},\mathrm{cut}_1}(s,a)}{\sum_{j\in\mathcal{G}} \tilde{N}_h^{j,\mathrm{cut}_2}(s,a)}} \frac{\sqrt{2m}}{\sqrt{\sum_{j\in\mathcal{G}} N_h^j(s,a)}} \tag{337}$$

$$\leq \frac{\sqrt{2\kappa_{\text{even}} m}}{\sqrt{\sum_{j\in\mathcal{G}} N_h^j(s,a)}} \tag{338}$$

Thus

$$\mathscr{A}_2 \leq 2\sum_{h=1}^{H} \mathbb{E}_{d^{\tilde{\pi}}} \Bigg[\left(\frac{2}{\sqrt{\sum_{j\in\mathcal{G}} \tilde{N}_h^j(s,a)}} + \frac{8\alpha m\sqrt{N_h^{\mathrm{cut}}(s,a)}}{\sum_{j\in\mathcal{G}} \tilde{N}_h^j(s,a)} \right) H\sqrt{2\log\frac{2mSAH}{\delta}} \tag{339}$$

$$\mathbf{1}\left\{N_h^{\mathcal{G},\mathrm{cut}_2}(s_h,a_h) > 0\right\} \Bigg] \tag{340}$$

$$\leq 2\sum_{h=1}^{H} \mathbb{E}_{d^{\tilde{\pi}}} \left[\frac{\left(2+8\alpha\sqrt{2m}\right)\sqrt{\kappa_{\text{even}}}}{\sqrt{\sum_{j\in\mathcal{G}} N_h^j(s,a)}} H\sqrt{2\log\frac{2mSAH}{\delta}} \mathbf{1}\left\{N_h^{\mathcal{G},\mathrm{cut}_2}(s_h,a_h) > 0\right\} \right] \tag{341}$$

$$\leq 2\left(2+8\alpha\sqrt{2m}\right)\sqrt{\kappa_{\text{even}}} H\sqrt{2\log\frac{2mSAH}{\delta}} \sum_{h=1}^{H} \mathbb{E}_{d^{\tilde{\pi}}} \left[\frac{\mathbf{1}\left\{N_h^{\mathcal{G},\mathrm{cut}_2}(s_h,a_h) > 0\right\}}{\sqrt{\sum_{j\in\mathcal{G}} N_h^j(s,a)}} \right] \tag{342}$$

Recall that $\mathcal{C}_h = \left\{s | N_h^{\mathcal{G},\mathrm{cut}_2}(s,\tilde{\pi}(s)) > 0\right\}$. By Cauchy–Schwarz inequality and the definition of κ in Definition 6.3,

$$\mathbb{E}_{d^{\tilde{\pi}}} \left[\frac{\mathbf{1}\left\{N_h^{\mathcal{G},\mathrm{cut}_2}(s_h,a_h) > 0\right\}}{\sqrt{\sum_{j\in\mathcal{G}} N_h^j(s,a)}} \right] \leq \sqrt{\mathbb{E}_{d^{\tilde{\pi}}} \left[\frac{\mathbf{1}\left\{N_h^{\mathcal{G},\mathrm{cut}_2}(s_h,a_h) > 0\right\}}{\sum_{j\in\mathcal{G}} N_h^j(s,a)} \right]} \tag{343}$$

$$= \sqrt{\sum_{s\in\mathcal{C}_h} \frac{d_h^{\tilde{\pi}}(s)}{\sum_{j\in\mathcal{G}} N_h^j(s,a)}} \tag{344}$$

$$= \sqrt{\sum_{s\in\mathcal{C}_h} \frac{d_h^{\tilde{\pi}}(s)}{\sum_{j\in\mathcal{G}} N_h^j(s,a)/\sum_{j\in\mathcal{G}} K_j} \frac{1}{\sum_{j\in\mathcal{G}} K_j}} \tag{345}$$

$$\leq \sqrt{\sum_{s\in\mathcal{C}_h} \frac{\kappa}{\sum_{j\in\mathcal{G}} K_j}} \leq \sqrt{\frac{\kappa S}{\sum_{j\in\mathcal{G}} K_j}} \tag{346}$$

In conclusion,

$$\mathrm{SubOpt}(\hat{\pi},\tilde{\pi}) \leq \mathscr{A}_1 + \mathscr{A}_2 \tag{347}$$

$$\leq 2Hp^{\mathcal{G},0} + 2\left(2+8\alpha\sqrt{2m}\right) \frac{\sqrt{\kappa\kappa_{\text{even}} S}}{\sqrt{\sum_{j\in\mathcal{G}} K_j}} H^2 \sqrt{2\log\frac{2mSAH}{\delta}} \tag{348}$$

$$=2Hp^{\mathcal{G},0}+O\left(\sqrt{\kappa\kappa_{\text{even}}}H^2\sqrt{S}\frac{1+\alpha\sqrt{m}}{\sqrt{\sum_{j\in\mathcal{G}}K_j}}\sqrt{\log\frac{mSAH}{\delta}}\right) \tag{349}$$

□

E USEFUL INEQUALITIES

Theorem E.1 (Bernstein type of bound for martingale, Theorem 1.6 of (Freedman, 1975))**.** *Let $(\Omega,\mathcal{F},P)$ be a probability triple. Let $\mathcal{F}_0\subset\mathcal{F}_1\subset\cdots$ be an increasing sequence of sub-σ-fields of $\mathcal{F}$. Let $X_1,X_2,\ldots$ be random variables on $(\Omega,\mathcal{F},P)$, such that X_n is $\mathcal{F}_n$ measurable. Let $V_n=\mathbb{V}\left[X_n|\mathcal{F}_{n-1}\right]$. Assume $|X_n|\leq 1$ and $\mathbb{E}[X_n|\mathcal{F}_{n-1}]=0$. Let*

$$S_n=X_1+\cdots+X_n \tag{350}$$
$$T_n=V_1+\cdots+V_n, \tag{351}$$

where $S_0=T_0=0$. Then, for any $a>0$, $b>0$,

$$\mathbb{P}\left(|S_n|\geq a \text{ and } T_n\leq b \text{ for some } n\right)\leq 2\exp\left(-\frac{a^2}{2(a+b)}\right). \tag{352}$$

By union bound and partition, we can get a more useful version of Theorem E.1.

We first present a result, which shows: given,

$$\mathbb{P}\left(X\geq t, Y\leq t\right)\leq\delta(t) \tag{353}$$

We can bound $\mathbb{P}\left(X\geq Y\right)$ up to some error.

Lemma E.2. *Let $\{A_n\}_{n=1}^N$ and $\{B_n\}_{n=1}^N$ be two sequences of random variables. We don't make any assumptions about independence. Assume*

- *$\forall n$, $0\leq B_n\leq nM$ almost surely;*
- *$\forall\delta>0$, $f_\delta:\mathbb{R}_+\mapsto\mathbb{R}_+$, $f_\delta(\cdot)$ monotonic increasing,*

If for all $t>0$,

$$\mathrm{P}\left(\bigcup_{n=1}^N\{|A_n|\geq f_\delta(t), B_n\leq t\}\right)\leq\delta \tag{354}$$

Then for any $\epsilon>0$,

$$\mathrm{P}\left(\bigcup_{n=1}^N\{|A_n|\geq f_\delta(B_n+\epsilon)\}\right)\leq NM\lceil 1/\epsilon\rceil\delta \tag{355}$$

Proof. See proof in Section E.1. □

Corollary E.3. *Under the assumption of Theorem E.1, suppose X_n terminate at $n=N$. Then, for all $0<\delta<2\exp(-2)$,*

$$\mathbb{P}\left(\bigcup_{n=1}^N\left\{|S_n|\geq 3\log\frac{2N}{\delta}+\sqrt{2T_n\log\frac{2N}{\delta}}\right\}\right)\leq\delta \tag{356}$$

Proof of Corollary E.3. Let $\frac{\delta}{N}=2\exp\left(-\frac{a^2}{2(a+b)}\right)$ then

$$a=\log\frac{2N}{\delta}+\sqrt{\log^2\frac{2N}{\delta}+2b\log\frac{2N}{\delta}} \tag{357}$$

by Theorem E.1, For all $b > 0$,

$$\mathbb{P}\left(|S_n| \geq \log\frac{2N}{\delta} + \sqrt{\log^2\frac{2N}{\delta} + 2b\log\frac{2N}{\delta}},\text{ and } T_n \leq b \text{ for some } n\right) \leq \delta/N \tag{358}$$

In Lemma E.2, let:

- $A_n = S_n$, $B_n = T_n$, $M = 1$
- $\epsilon = \frac{1}{2}\log\frac{2N}{\delta}$
- $f_\delta(x) = \log\frac{2N}{\delta} + \sqrt{\log^2\frac{2N}{\delta} + 2x\log\frac{2N}{\delta}}$

Because $0 < \delta < 2\exp(-2)$, $\epsilon \geq 1$. then, we get:

$$\mathbb{P}\left(\bigcup_{n=1}^{N}\left\{|S_n| \geq 3\log\frac{2N}{\delta} + \sqrt{2T_n\log\frac{2N}{\delta}}\right\}\right) \tag{359}$$

$$\leq\mathbb{P}\left(\bigcup_{n=1}^{N}\left\{|S_n| \geq \log\frac{2N}{\delta} + \sqrt{2\log^2\frac{2N}{\delta} + 2T_n\log\frac{2N}{\delta}}\right\}\right) \tag{360}$$

$$\leq N\lceil 1/\epsilon\rceil\frac{\delta}{N} \leq \delta \tag{361}$$

□

E.1 Proof For Lemma E.2

Proof of Lemma E.2. For discrete random variables, we can just condition on each possible value of B_n and use a union bound. Here, because B_n can be a continuous random variable, we divide the range of B_n into intervals and upper bound the target by the law of total probability.

For all n, let:

$$0 < \frac{1}{\lceil 1/\epsilon\rceil} < \frac{2}{\lceil 1/\epsilon\rceil} < \cdots < \frac{nM\lceil 1/\epsilon\rceil}{\lceil 1/\epsilon\rceil} = nM \tag{362}$$

Be a partition of interval $[0, nM]$. Let $I_i := \left[\frac{i-1}{\lceil 1/\epsilon\rceil}, \frac{i}{\lceil 1/\epsilon\rceil}\right], i = 1, \ldots, nM\lceil 1/\epsilon\rceil$ be a set of intervals. Note that, $\bigcup_{i=1}^{nM\lceil 1/\epsilon\rceil} I_i = [0, nM]$.

Then

$$\bigcup_{n=1}^{N}\{|A_n| \geq f_\delta(B_n+\epsilon)\} = \bigcup_{n=1}^{N}\bigcup_{i=1}^{nM\lceil 1/\epsilon\rceil}\{|A_n| \geq f_\delta(B_n+\epsilon), B_n \in I_i\} \tag{363}$$

$$= \bigcup_{n=1}^{N}\bigcup_{i=1}^{nM\lceil 1/\epsilon\rceil}\left\{|A_n| \geq f_\delta(B_n+\epsilon), \frac{i-1}{\lceil 1/\epsilon\rceil} \leq B_n \leq \frac{i}{\lceil 1/\epsilon\rceil}\right\} \tag{364}$$

$$\subseteq \bigcup_{n=1}^{N}\bigcup_{i=1}^{nM\lceil 1/\epsilon\rceil}\left\{|A_n| \geq f_\delta(\frac{i}{\lceil 1/\epsilon\rceil}), B_n \leq \frac{i}{\lceil 1/\epsilon\rceil}\right\} \tag{365}$$

$$\subseteq \bigcup_{n=1}^{N}\bigcup_{i=1}^{NM\lceil 1/\epsilon\rceil}\left\{|A_n| \geq f_\delta(\frac{i}{\lceil 1/\epsilon\rceil}), B_n \leq \frac{i}{\lceil 1/\epsilon\rceil}\right\} \tag{366}$$

$$= \bigcup_{i=1}^{NM\lceil 1/\epsilon\rceil}\bigcup_{n=1}^{N}\left\{|A_n| \geq f_\delta(\frac{i}{\lceil 1/\epsilon\rceil}), B_n \leq \frac{i}{\lceil 1/\epsilon\rceil}\right\} \tag{367}$$

Thus

$$\mathrm{P}\left(\bigcup_{n=1}^{N}\{|A_n| \geq f_\delta(B_n+\epsilon)\}\right) \leq \sum_{i=1}^{NM\lceil 1/\epsilon\rceil} \mathbb{P}\left(\bigcup_{n=1}^{N}\left\{|A_n| \geq f_\delta(\frac{i}{\lceil 1/\epsilon\rceil}), B_n \leq \frac{i}{\lceil 1/\epsilon\rceil}\right\}\right) \tag{368}$$

$$\leq NM\lceil 1/\epsilon\rceil\delta \quad \text{(By (354))} \tag{369}$$

□