Agnostic Interactive Imitation Learning: New Theory and Practical Algorithms

Yichen Li [1] **Chicheng Zhang** [1]

Abstract

We study interactive imitation learning, where a learner interactively queries a demonstrating expert for action annotations, aiming to learn a policy that has performance competitive with the expert, using as few annotations as possible. We focus on the general agnostic setting where the expert demonstration policy may not be contained in the policy class used by the learner. We propose a new oracle-efficient algorithm MFTPL-P (abbreviation for Mixed Follow the Perturbed Leader with Poisson perturbations) with provable finite-sample guarantees, under the assumption that the learner is given access to samples from some "explorative" distribution over states. Our guarantees hold for any policy class, which is considerably broader than prior state of the art. We further propose BOOTSTRAP-DAGGER, a more practical variant that does not require additional sample access. Empirically, MFTPL-P and BOOTSTRAP-DAGGER notably surpass online and offline imitation learning baselines in continuous control tasks.

1. Introduction

Imitation learning (IL) is a learning paradigm for training sequential decision making agents using expert demonstrations (Bagnell, 2015). It seeks to learn a policy whose performance is on par with the expert, with as few expert demonstrations as possible. Compared to reinforcement learning which potentially requires intricate reward engineering, imitation learning sidesteps this challenge, making it apt for complex decision making problems (Osa et al., 2018).

The format of expert demonstrations often comprises of (state, action) pairs, each representing the expert's action taken under respective state. At first sight, imitation learning seems similar to supervised learning, where one would like to train a policy that maps states to actions recommended by the expert. However, it is now well-known that these two problems are different in nature (Pomerleau, 1988): compared to supervised learning, imitation learning agents are faced with a new fundamental challenge of *covariate shift*: the state distribution the learner sees at the test stage can be very different from that at the training stage. To elaborate, upon deploying the trained policy, its imperfection in mimicking the expert can result in *compounding error*: the learner may make an initial mistake and enter a state not covered by the distribution of expert demonstrations, and takes another incorrect action due to lack of training data coverage in this state. This may lead to sequences of consecutive states that were unseen in the expert demonstrations, ultimately resulting in poor performance. Addressing such problem calls for better data collection methods beyond naively collecting expert trajectories; as summarized by (Bojarski et al., 2016) in the context of imitation learning for autonomous driving, "Training with data from only the human driver is not sufficient. The network must learn how to recover from mistakes".

To cope with the covariate shift challenge, the interactive imitation learning paradigm has been proposed and used in practice (Ross and Bagnell, 2010; Ross et al., 2011; Ross and Bagnell, 2014; Sun et al., 2017; Pan et al., 2020; Celemin et al., 2022). Instead of having only access to offline expert trajectories, in interactive imitation learning, the learner has the freedom to select states to ask for expert annotations. This allows more targeted feedback and gives the learning agent opportunities to learn to "recover from mistakes", and thus achieve better performance.

Recent years have seen many exciting developments on designing provably efficient interactive imitation learning algorithms, using new algorithmic approaches such as distribution / moment matching (Ke et al., 2020; Swamy et al., 2021) and online classification / regression (Rajaraman et al., 2021; Sekhari et al., 2023). Most of these works rely on some realizability assumption: they either assume that the learner is given a policy class that contains the expert policy (Rajaraman et al., 2021; Sekhari et al., 2023; Sun et al., 2017), or that the learner is given some function

[1] Department of Computer Science, University of Arizona, Tucson, AZ, USA. Correspondence to: Yichen Li <yichenl@arizona.edu>, Chicheng Zhang <chichengz@cs.arizona.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

class that contains reward or value functions of the underlying environment MDP (Swamy et al., 2021; 2022; Sun et al., 2019).

Although realizability assumptions are useful in driving the development of theory and practical algorithms, misspecified model settings are common in practice (e.g. in autonomous driving (Pan et al., 2020)). Thus, it is important to design policy search-based imitation learning algorithms that can work in general agnostic (i.e., nonrealizable) settings. However, the fundamental computational and statistical limits of imitation learning in the general agnostic setting are not well-understood. An important line of work (Ross et al., 2011; Ross and Bagnell, 2014; Sun et al., 2017) tackles this question by establishing a general reduction from agnostic interactive imitation learning to no-regret online learning (Shalev-Shwartz et al., 2011) (see Section 3 below for a recap). Its key intuition is that, with access to a learner that can perform online prediction by swiftly adapting to nonstationary data, one can use it to find a policy that mostly agrees with the expert on its own state visitation distribution. In other words, online learning can be utilized to learn a policy that recovers from its own mistakes. Utilizing this general reduction framework, agnostic imitation learning algorithms with provable computational and finite-sample statistical efficiency guarantees have been proposed (Cheng et al., 2019b;a; Li and Zhang, 2022). In the discrete-action setting, the development so far has been limited: state-of-the-art efficient algorithms (Li and Zhang, 2022) rely on a strong technical "small separator" assumption on the policy class, which is only known to hold for a few policy classes (such as disjunctions and linear classes) (Syrgkanis et al., 2016; Dudík et al., 2020). This motivates our main question: can we design computationally and statistically efficient imitation learning algorithms in the general agnostic setting for *general policy classes*?

Our contributions are twofold:

- Theoretically, we design a provably efficient online imitation learning algorithm that enjoys no-regret guarantees for discrete action spaces for general policy classes. The no-regret property guarantees the learning agent's swift adaptation to data distributions it encounters, ensuring competitiveness to the expert policy (Ross et al., 2011). Specifically, our algorithm, *Mixed Follow the Perturbed Leader with Poisson Perturbations* (abbrev. MFTPL-P), assumes sample access to a distribution d_0 that "covers" the state visitation distributions of all policies in the policy class of interest. Algorithmically, MFTPL-P can be viewed as maintaining an ensemble of policies, each member of which is trained using historical expert demonstration data combined with noisy perturbation examples drawn from d_0.

  Inspired by recent analysis of efficient smoothed online learning algorithms (Haghtalab et al., 2022a; Block et al., 2022), we prove that MFTPL-P: (1) has a sublinear regret guarantee, which can be easily converted into a guarantee of its output policy's suboptimality; (2) is computationally efficient, assuming access to an offline learning oracle. Our computational efficiency result relies on arguably much weaker assumptions than previous state-of-the-art efficient learning algorithms, whose guarantees require strong assumptions on the policy class (Li and Zhang, 2022) or convexity of loss function (Cheng and Boots, 2018; Cheng et al., 2019b;a; Sun et al., 2017).

- Empirically, we verify the utility of using sample-based perturbations in MFTPL-P. Furthermore, we evaluate MFTPL-P and show that it outperforms the popular DAGGER and Behavior Cloning baselines across several continuous control benchmarks. Inspired by the ensemble nature of MFTPL-P, we also propose and evaluate a practical approximation of it, namely BOOTSTRAP-DAGGER, that avoids sample access to d_0 and achieves competitive performance. We also investigate the expert demonstration data collected by BOOTSTRAP-DAGGER and show that it gathers pertinent expert demonstration data more efficiently than DAGGER.

2. Related Work

Imitation Learning with Interactive Expert. Existing works in interactive imitation learning established a solid foundation to tackle covariate shift, with the help of an interactive demonstration expert. Early works reduced interactive imitation learning to offline learning (Ross and Bagnell, 2010; Daumé et al., 2009) and (Ross et al., 2011) proposed a general reduction from interactive imitation learning to online learning. This major line of research (Ross et al., 2011; Ross and Bagnell, 2014; Sun et al., 2017; Cheng and Boots, 2018; Cheng et al., 2019a;b; Rajaraman et al., 2021; Li and Zhang, 2022) provided provably efficient online regret guarantees, which can be translated to guarantees of learned policy's competitiveness with expert policy. It has been shown in (Rajaraman et al., 2021) that interactive imitation learning can be significantly more sample efficient in favorable environments than its offline counterpart. Recently, (Sekhari et al., 2023) reduced interactive imitation learning to online regression, which assumes that the expected value of expert annotation as a function of state lies in a real-valued regressor class.

As discussed in the introduction section, most of these works make some realizability assumptions. In the absence of such realizability assumption (i.e., the agnostic setting), existing guarantees require convexity of the loss functions with respect to policy's pre-specified parameters (which may not hold for general policy classes) (Cheng and Boots, 2018; Cheng et al., 2019b;a; Sun et al., 2017) or are only applicable to a few special policy classes (Li and Zhang, 2022).

Another line of work studies interactive imitation learning with *online expert intervention feedback* (Zhang and Cho, 2016; Menda et al., 2017; 2019; Cui et al., 2019; Kelly et al., 2019; Hoque et al., 2021; Spencer et al., 2022); here it is assumed that the learner has access to an expert in real time; the learner may cede control to the expert to request for demonstrations (machine-gated setting) or the expert can actively intervene (human-gated setting). Different from our learning objective, these works have a focus on ensuring training-time safety.

Imitation Learning without Interactive Expert. Given only offline expert demonstrations, Behavior Cloning (BC) naively reduced imitation learning to offline classification (Ross and Bagnell, 2010; Syed and Schapire, 2010). By further assuming the ability to interact with the environment, generative adversarial imitation learning (GAIL) (Ho and Ermon, 2016) and following line of works (Sun et al., 2019; Spencer et al., 2021) applied moment matching to tackle covariate shift. (Chang et al., 2021) replaced the requirement of environment access by combining model estimation and pessimism. DART (Laskey et al., 2017) accessed neighborhood states of the expert trajectories by injecting noise during the expert's demonstrations, achieving performance comparable to DAGGER. DRIL (Brantley et al., 2019) trained an ensemble of policies using expert demonstrations, then leveraged the variance among ensemble predictions as a cost, which was optimized through reinforcement learning together with the classification loss on expert set. These works are different from ours, in that they do not assume access to an expert that can provide interactive demonstrations in the training process. (Ke et al., 2020; Swamy et al., 2021; 2022) formulates imitation learning as a distribution matching problem, and further reduce it to solving two-player zero-sum games, which can be solved either interactively or offline, however their guarantees only hold under realizability assumptions.

Concentrability and Smoothness. One key assumption we make is that the learning agent has sample access of some covering distribution, so that all policy's visitation distributions are "smooth" with respect to it (Assumption 2). This allows us to design provably-efficient imitation learning algorithms using techniques for smoothed online learning (Haghtalab et al., 2022a; Block et al., 2022). Our assumption is related to the boundedness of the concentrability coefficient commonly used in offline reinforcement learning (Munos, 2007; Munos and Szepesvári, 2008; Chen and Jiang, 2019; Xie and Jiang, 2020; 2021), which was first introduced by Munos (Munos, 2003). Concentrability in general holds for MDPs with "noisy" transitions (i.e. nontrivial probability of transitioning to all potential next states) but can also hold for deterministic transitions (Szepesvári and Munos, 2005). The concentrability assumptions most related to ours are in (Xie and Jiang, 2020; Xie et al., 2022). However, note that our Assumption 2 is solely with respect to distributions over states instead of (state, action) pairs. After all, unlike standard offline reinforcement learning, in IL, we neither seek to learn the optimal value function nor assume access to a candidate value function class.

3. Preliminaries

Basic notations. Denote by $[N] = \{1, \ldots, N\}$. Given a set $\mathcal{E}$, denote by $\Delta(\mathcal{E})$ the set of all probability distributions over it; when $\mathcal{E}$ is finite, elements in $\Delta(\mathcal{E})$ can be represented by probability vectors in $\mathbb{R}^{|\mathcal{E}|}$: $\left\{(w_e)_{e\in\mathcal{E}} : \sum_{e\in\mathcal{E}} w_e = 1, w_e \geq 0, \forall e \in \mathcal{E}\right\}$. Given a function $f : \mathcal{E} \to \mathbb{R}$, denote by $\|f\|_\infty := \max_{e\in\mathcal{E}} |f(e)|$.

Markov decision process. We study imitation learning for sequential decision making, where we model the environment as a Markov decision process (MDP). An episodic MDP $\mathcal{M}$ is a tuple $(\mathcal{S}, \mathcal{A}, \rho, P, C, H)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, ρ is the initial state distribution, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition probability distribution, $C : \mathcal{S} \times \mathcal{A} \to \Delta([0, 1])$ is the cost distribution, $H \in \mathbb{N}$ is the length of the time horizon. Without loss of generality, we assume that $\mathcal{M}$ is layered, in that $\mathcal{S}$ can be partitioned to $\{\mathcal{S}_t\}_{t=1}^H$, where for every step t, $s \in \mathcal{S}_t$ and action $a \in \mathcal{A}$, $P(\cdot \mid s, a)$ is supported on $\mathcal{S}_{t+1}$.

Agent-environment interaction and policy. An agent interacts with MDP in one episode by first observing initial state $s_1 \sim \rho$, and for every step $t = 1, \ldots, H$, takes an action a_t, receives cost $c_t \sim C(\cdot \mid s, a)$, and transitions to next state $s_{t+1} \sim \mathbb{P}(\cdot \mid s_t, a_t)$. A stationary (history-independent) policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps a state to a distribution over actions, which can be used by the agent to take actions, i.e. $a_t \sim \pi(\cdot \mid s_t)$ for all t.

Value Functions. Given policy π, for every $t \in [H]$ and $s \in \mathcal{S}_t$, denote by $V_\pi^t(s) = \mathbb{E}\left[\sum_{t'=t}^H c_{t'} \mid s_t = s, \pi\right]$ and $Q_\pi^t(s, a) = \mathbb{E}\left[\sum_{t'=t}^H c_{t'} \mid (s_t, a_t) = (s, a), \pi\right]$ the state-value function and action-value function of π, respectively. When it is clear from context, we will abbreviate $V_\pi^t(s)$ and $Q_\pi^t(s, a)$ as $V_\pi(s)$ and $Q_\pi(s, a)$, respectively. Denote by $J(\pi) = \mathbb{E}_{s_1\sim\rho}\left[V_\pi(s_1)\right]$ the expected cost of policy π. Given policy π, denote by $d_\pi^{\mathcal{M}}(\cdot) = \frac{1}{H}\sum_{t=1}^H \mathbb{P}(s_t = \cdot)$ the state visitation distribution of π under $\mathcal{M}$; when it is clear from context, we will abbreviate $d_\pi^{\mathcal{M}}$ as d_π.

To help the learner make decisions, the learner is given a policy class $\mathcal{B}$, which is a structured set of deterministic policies $\pi : \mathcal{S} \to \mathcal{A}$. Denote by $\Pi_{\mathcal{B}} = \left\{\pi_w(a \mid s) := \sum_{h \in \mathcal{B}} w_h I(a = h(s)) : w \in \Delta(\mathcal{B})\right\}$ the mixed policy class induced by $\mathcal{B}$. Denote by $\pi^{\exp}$ the expert's policy. The learning setting is said to be *realizable* if it is known apriori that $\pi^{\exp} \in \mathcal{B}$; otherwise, the learning setting is said to be *agnostic*. We will use S, A, B to denote $|\mathcal{S}|$, $|\mathcal{A}|$, and $|\mathcal{B}|$ respectively.*

Definition 1. *A (MDP, expert policy) pair $(\mathcal{M}, \pi^{\exp})$ is said to be μ-recoverable with respect to loss ℓ, if for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $Q_{\pi^{\exp}}(s, a) - V_{\pi^{\exp}}(s) \le \mu \cdot \ell(a, \pi^{\exp}(s))$.*

Two notable examples are:

- $\ell(a, a') = I(a \ne a')$ is the 0-1 loss. Then μ-recoverability implies that for all $a \ne \pi^{\exp}(s)$, $Q_{\pi^{\exp}}(s, a) - V_{\pi^{\exp}}(s) \le \mu$, which is suitable for discrete-action settings (Ross et al., 2011).
- $\ell(a, a') = \|a - a'\|$ is the absolute loss. Then a sufficient condition for μ-recoverability is that $Q_{\pi^{\exp}}(s, a)$ is μ-Lipschitz in a with respect to $\|\cdot\|$. This is suitable for continuous-action settings (Pan et al., 2020).

In prior works (Ross and Bagnell, 2010; Rajaraman et al., 2021), it has been demonstrated that for cases where $(M, \pi^{\exp})$ is μ-recoverable with $\mu \ll H$, i.e., the expert can recover from mistakes with little extra cost, interactive imitation learning can achieve a much lower sample complexity than offline imitation learning.

Reduction from Interactive Imitation Learning to Online Learning. Ross et al. (2011) proposes a general reduction from interactive imitation learning to no-regret online learning, which we will frequently refer to as the *online imitation learning framework* throughout the paper. As our algorithm design and performance guarantees will be under this framework, we briefly recap it below. Its key insight is to simulate an N-round online learning game between the learner and the environment: at round n, the learner chooses a policy π_n, and the environment responds with loss function F_n. The learner then incurs a loss of $F_n(\pi_n)$, and observes as feedback a sample-based approximation of $F_n(\cdot)$. Here, $F_n(\pi) := \mathbb{E}_{s \sim d_{\pi_n}, a \sim \pi(\cdot|s)} \ell(a, \pi^{\exp}(s))$ is carefully chosen as the expected loss of policy π with respect to the expert policy $\pi^{\exp}$ under the state visitation distribution of π_n. $F_n(\cdot)$ are naturally approximated by the average loss on supervised learning examples $(s, \pi^E(s))$, whose feature and label parts are sampled from d_{π_n} and queried from expert π^E, respectively. Ross et al. (2011) shows that, if $\{\pi_n\}_{n=1}^N$ has a low online regret, a policy returned uniformly at random from $\{\pi_n\}_{n=1}^N$ has an expected cost competitive with the expert. Formally:

Theorem 2 (Ross et al. (2011))**.** *Suppose $(\mathcal{M}, \pi^{\exp})$ is μ-recoverable with respect to ℓ. Define the regret of the sequence of policies $\{\pi_n\}_{n=1}^N$ w.r.t. policy class $\mathcal{B}$ as:*

$$\mathrm{Reg}(N) := \sum_{n=1}^{N} F_n(\pi_n) - \min_{\pi \in \mathcal{B}} \sum_{n=1}^{N} F_n(\pi). \qquad (1)$$

Then $\hat{\pi}$, which is by choosing a policy uniformly at random from $\{\pi_n\}_{n=1}^N$ and adhering to it satisfies:

$$J(\hat{\pi}) - J(\pi^{\exp}) \le \mu H \left(\min_{\pi \in \mathcal{B}} \frac{1}{N} \sum_{n=1}^{N} F_n(\pi) + \frac{\mathrm{Reg}(N)}{N} \right).$$

We will denote $\mathrm{EstGap} := \mu H \frac{\mathrm{Reg}(N)}{N}$, which can be viewed as the "estimation gap" that bounds the performance gap between $\hat{\pi}$ and the best policy in hindsight. Meanwhile, $\min_{\pi \in \mathcal{B}} \frac{1}{N} \sum_{n=1}^{N} F_n(\pi)$ measures the expressiveness of policy class $\mathcal{B}$ with respect to the expert policy $\pi^{\exp}$: it is smaller with a larger $\mathcal{B}$. In the special case that $\pi^{\exp} \in \mathcal{B}$ (the realizable setting) and $\pi^{\exp}$ is deterministic, $\min_{\pi \in \mathcal{B}} \frac{1}{N} \sum_{n=1}^{N} F_n(\pi) = 0$. For completeness, we provide Theorem 2's proof in Appendix A and incorporate discussions on limitations of the reduction-based framework.

4. Oracle-efficient Imitation Learning: Algorithm and Analysis for General Policy Classes

In this section, we present an interactive imitation learning algorithm for discrete action space settings with provable computational and statistical efficiency guarantees for general policy classes. It is based on the aforementioned online IL framework and aims to guarantee sublinear regret under the 0-1 loss ($\ell(a, a') = I(a \ne a')$) in the general agnostic setting.

A line of works design practical interactive learning algorithms by assuming access to offline learning oracles (Beygelzimer et al., 2010; Dann et al., 2018; Simchi-Levi and Xu, 2022; Dudík et al., 2020; Syrgkanis et al., 2016; Rakhlin and Sridharan, 2016). Following this, in designing computationally efficient IL algorithms, we also assume that our learner has access to an offline learning oracle that can output a policy that minimizes 0-1 classification loss given a input dataset of (state, action) pairs.

Assumption 1 (Offline learning oracle)**.** *There is an offline classification oracle $\mathcal{O}$ for policy class $\mathcal{B}$; specifically, i.e. given any input multiset of classification examples $D = \{(s, a)\}$, $\mathcal{O}$ returns $\mathcal{O}(D) = \mathrm{argmin}_{h \in \mathcal{B}} \sum_{(s,a) \in D} I(h(s) \ne a)$.*

[*] Although we assume S and B to be finite, as we will see, our sample complexity results only scale with $\ln S$ and $\ln B$.

We argue that this is a reasonable assumption – without a offline learning oracle, one may not even efficiently solve supervised learning, a special case of imitation learning.

Note that without further assumptions, oracle efficient online algorithms are impossible (Hazan and Koren, 2016). To allow the design of efficient online algorithms, we also make an assumption that the learner has sampling access to an explorative state distribution that can "cover" the state visitation distribution of any policy in the policy class to learn from:

Assumption 2 (Sampling access to covering distribution)**.** *The learner has sampling access to a covering distribution $d_0 \in \Delta(\mathcal{S})$, such that there exists $\sigma \leq 1$, for any $\pi \in \Pi_{\mathcal{B}}$, d_π is σ-smooth with respect to d_0; formally, $\left\|\frac{d_\pi}{d_0}\right\|_\infty \leq \frac{1}{\sigma}$.*

Assumption 2 is closely related to the smoothed online learning problem setup (Haghtalab et al., 2022a; Block et al., 2022): under this assumption, in the N-round online learning game induced in the online imitation learning framework, d_{π_n}, the distribution that induces the loss function F_n at round n, is σ-smooth with respect to covering distribution d_0 for all $n \in [N]$. The larger σ is, the less variability d_{π_n}'s can have, indicating that the underlying online learning problem may be more stationary and easier to learn. The special case $d_0 = d_{\pi^E}$ has also been studied in Spencer et al. (2021), although they do not provide a finite-sample analysis.

Challenges in applying existing approaches. Based on the connection between imitation learning and no-regret online learning mentioned in Section 3, it may be tempting to directly apply existing oracle-efficient smoothed online learning algorithms (Haghtalab et al., 2022a; Block et al., 2022) and establish regret guarantees. However, several fundamental challenges still remain. First and most fundamentally, existing smoothed online learning formulations assume that the sampling distribution at round n is chosen *before* the learner commits to its decision π_n (Haghtalab et al., 2022b;a; Block et al., 2022). Unfortunately, this assumption does not hold in the online imitation laerning framework – specifically, d_{π_n} can depend on π_n. Second, at each round of online imitation learning, the learner may collect and learn from a batch of examples, while (Haghtalab et al., 2022a; Block et al., 2022) only addresses the setting when the batch size is 1. Lastly, we consider general action set size A, meaning the learner needs to perform online multiclass classification, while (Haghtalab et al., 2022a; Block et al., 2022) only address binary classification and regression settings.

We address these challenges by proposing the *Mixed Follow the Perturbed Leader with Poission perturbations* algorithm (MFTPL-P, Algorithm 1). Specifically, we address the first challenge by making the following key observation: even though in the online IL framework, the sampling distribution at round n can now directly depend on π_n, as long as the sequence of policies $\{\pi_n\}$ has a *deterministic regret guarantee* in the original smoothed online learning problem, the same regret guarantee will carry over to the new online imitation learning problem. Such deterministic regret guarantee property, to the best of our knowledge, is not known to hold for randomized online learning algorithms such as Follow the Perturbed Leader (FTPL) (Kalai and Vempala, 2005), but holds for deterministic online learning algorithms such as an in-expectation version of FTPL (Abernethy et al., 2014) or Follow the Regularized Leader (FTRL) (Shalev-Shwartz et al., 2011).

Using this observation, MFTPL-P aims to approximate an in-expectation of FTPL to guarantee a sublinear regret. It follows the online IL framework: at round n, in the data collection step (line 8) , MFTPL-P rolls out the currently learned policy π_n in the MDP multiple times to sample K states from d_{π_n}. It then requests expert's demonstrations on them, obtaining a dataset D_n of (state, action) pairs. In the policy update step (line 4 to line 7), MFTPL-P calls the TRAIN-BASE function E times on the accumulated dataset D, to train a new policy π_{n+1}, which is an average of E base policies $\{\pi_{n+1,e}\}_{e=1}^{E}$. Each $\pi_{n+1,e}$ can be seen as a freshly-at-random output of the FTPL algorithm with *Poisson sample-based perturbations* (Haghtalab et al., 2022a): first drawing a Poisson random variable X representing perturbation sample size, then drawing Q, a set of X iid examples from covering distribution d_0 with uniform-at-random labels from $\mathcal{A}$; finally calling the offline oracle $\mathcal{O}$ on the perturbed dataset $D \cup Q$. It can now be seen that π_{n+1} approximates the output of an in-expectation version of FTPL: a larger E yields a better approximation, which ensures high-probability regret guarantees. Finally, MFTPL-P returns a policy $\hat{\pi}$ uniformly at random from the historical policies $\{\pi_n\}$.

Algorithmically, MFTPL-P can be viewed as maintaining an ensemble of E policies in an online fashion and use it to perform strategic collection of expert demonstration data. For this reason, we will refer to E as the ensemble size parameter. Similar algorithmic approaches have been proposed in imitation learning with expert intervention feedback (Menda et al., 2019); however, as discussed in Section 2, these works focus on ensuring safety in training and do not provide finite-sample analysis.

We show the following theorem regarding the regret guarantee of MFTPL-P; we defer its full version (Theorem 22), along with proofs to Appendix B.

Theorem 3. *For any $\delta \in (0, 1]$, for large enough N,* MFTPL-P *with appropriate setting of its parameters E, λ outputs $\{\pi_n\}_{n=1}^N$ that satisfies that with probability at least*

Algorithm 1 MFTPL-P

```
Input: MDP $\mathcal{M}$, expert $\pi^{\exp}$, policy class $\mathcal{B}$, oracle $\mathcal{O}$, covering distribution $d_0$, sample size per round $K$, ensemble size $E$, perturbation budget $\lambda$.
Initialize $D = \emptyset$.
for $n = 1, 2, \ldots, N$ do
    for $e = 1, 2, \ldots, E$ do
        $\pi_{n,e} \leftarrow$ TRAIN-BASE$(D, d_0, \lambda)$
    end for
    Set $\pi_n(a \mid s) := \frac{1}{E}\sum_{e=1}^{E} I(\pi_{n,e}(s) = a)$.
    $D_n = \left\{(s_{n,k}, \pi^{\exp}(s_{n,k}))\right\}_{k=1}^{K} \leftarrow$ sample $K$ states i.i.d. from $d_{\pi_n}$ by rolling out $\pi_n$ in $\mathcal{M}$, and query expert $\pi^{\exp}$ on these states.
    Aggregate datasets $D \leftarrow D \cup D_n$.
end for
Return $\hat{\pi} \leftarrow$ AGGREGATE-POLICIES$(\{\pi_{n,e}\}_{n=1,e=1}^{N+1,E})$
function TRAIN-BASE $(D, d_0, \lambda)$:
    Sample $X \sim \mathrm{Poi}(\lambda)$
    Sample $Q \leftarrow$ draw i.i.d. perturbation samples $\{(\tilde{s}_{n,x}, \tilde{a}_{n,x})\}_{x=1}^{X}$ from $\mathcal{D}_0 = d_0 \otimes \mathrm{Unif}(\mathcal{A})$.
    Return $h \leftarrow \mathcal{O}(D \cup Q)$.
function AGGREGATE-POLICIES $(\{\pi_{n,e}\}_{n=1,e=1}^{N+1,E})$:
    Sample $\hat{n} \sim \mathrm{Unif}([N])$
    Return $\pi_{\hat{n}}(a \mid s) := \frac{1}{E}\sum_{e=1}^{E} I(\pi_{\hat{n},e}(s) = a)$.
```

$1 - \delta$, $\mathrm{Reg}(N)$ *is at most*

$$\tilde{O}\left(\sqrt{N}\left(\frac{A(\ln B)^2}{\sigma K^2}\right)^{\frac{1}{4}} + \sqrt{N}\left(\frac{A \ln B}{\sigma}\right)^{\frac{1}{4}} + \sqrt{N \ln \frac{1}{\delta}}\right),$$

where we recall that B denotes the size of the base policy class $\mathcal{B}$.

Specialized to $A = 2$ and $K = 1$, this result is consistent with Haghtalab et al. (2022a) in the regret bound is dominated by $\sqrt{N}\left(\frac{(\ln B)^2}{\sigma}\right)^{\frac{1}{4}}$; we remark again though, that our regret analysis needs to get around the three additional challenges mentioned above.

In view of Theorem 2, Theorem 3 translates to the following result on the sample complexity of expert demonstrations and the number of calls to the classification oracle, for $\mathrm{EstGap} = \frac{\mu H \mathrm{Reg}(N)}{N}$ to be at most ϵ:

Corollary 4. *For any small enough* $\epsilon > 0$, MFTPL-P, *by setting its parameters as in Theorem 3 and number of rounds* $N = \tilde{O}\left(\frac{\mu^2 H^2 \sqrt{A \ln(B)}}{\epsilon^2 \sqrt{\sigma}}\right)$ *and batch size* $K = \sqrt{\ln B}$, *guarantees that* $\frac{\mu H \mathrm{Reg}(N)}{N} \leq \epsilon$ *with high probability, using* $\tilde{O}\left(\frac{\mu^2 H^2 \sqrt{A} \ln B}{\epsilon^2 \sqrt{\sigma}}\right)$ *expert demonstrations, and* $\tilde{O}\left(\frac{\mu^4 H^4 A^2 (\ln B)^2}{\epsilon^4 \sigma}\right)$ *calls to* $\mathcal{O}$.

In practice, the batch size K may be considered as part of problem specification and chosen ahead of time; motivated by this, we provide a version of this corollary with general K, Corollary 23, in Appendix B.

Table 1 compares MFTPL-P with (Li and Zhang, 2022) and Behavior Cloning in terms of the number of expert demonstrations for $\mathrm{EstGap} \leq \epsilon$, with a focus on comparing their dependences on μ, H and ϵ. Both MFTPL-P and (Li and Zhang, 2022) have a coefficient of $\mu^2 H^2$, much smaller than H^4 for Behavior Cloning, while (Li and Zhang, 2022) requires that the policy class $\mathcal{B}$ has a small separator set (Syrgkanis et al., 2016; Dudík et al., 2020; Li and Zhang, 2022), which is only known to hold for a few $\mathcal{B}$'s.

Table 1: Number of expert demonstrations $C(\epsilon)$ for $\mathrm{EstGap} \leq \epsilon$.

Algorithms	$C(\epsilon)$	Remarks
MFTPL-P (this work)	$\frac{\mu^2 H^2}{\epsilon^2}$	General $\mathcal{B}$, Access d_0
Li and Zhang (2022)	$\frac{\mu^2 H^2}{\epsilon^2}$	$\mathcal{B}$ small separator set
Behavior Cloning	$\frac{H^4}{\epsilon^2}$	General $\mathcal{B}$

5. Experiments

In this section, we evaluate MFTPL-P and its variant, comparing them with online and offline IL baselines in 4 continuous control tasks from OpenAI Gym (Brockman et al., 2016). Our experiments are designed to answer the following questions: **Q1:** Does sample-based perturbation provide any benefit in MFTPL-P? **Q2:** How does the choice of covering distribution d_0 affect the performance of MFTPL-P? **Q3:** Does MFTPL-P outperform online and offline IL baselines? **Q4:** Can we find a practical variant of MFTPL-P that achieves similar performance to MFTPL-P without additional sample access to some covering distribution? **Q5:** If Q3 and Q4 are true, which component of our algorithms confers this advantage?

Our experiment sections are organized as follows: Section 5.1 provides an introduction to our experimental settings. Section 5.2 presents positive results for **Q1** and **Q2**, evaluating MFTPL-P on two continuous control tasks using a linear policy class $\mathcal{B}$ with nonrealizable experts. Subsequently, Section 5.3 affirmatively answers **Q3** and **Q4** and introduces BOOTSTRAP-DAGGER (abbreviated as BD), a practical variant of MFTPL-P, and demonstrates the efficacy of our algorithms through neural network-based experiments. Across 4 continuous control tasks that encompass realizable and nonrealizable settings, BD and MFTPL-P outperform both online and offline IL baselines. Finally, for **Q5**, Section 5.4 investigates the underlying reasons for BD's improvement over the DAGGER baseline.

5.1. Experiment Settings

Environment: Following Brantley et al. (2019), we use normalized states. The name and {state dimension, action dimension} for each continuous control task are: Ant $\{28, 8\}$, Half-Cheetah $\{18, 6\}$, Hopper $\{11, 3\}$, and Walker2D $\{18, 6\}$.

Expert: For each task, the expert policy π^{exp} is a multilayer perceptron (MLP) with 2 hidden layers of size 64 and corresponding state, action dimension, pretrained by TRPO (Schulman et al., 2015). We employ TRPO's stochastic policy, sampling expert actions from MLP output with Gaussian noise.

Offline Learning Oracle: Our MFTPL-P relies on offline learning oracle $\mathcal{O}$; we describe their implementations below. In continuous control tasks, the training loss $\tilde{\ell}(a, a')$ is calculated by clipping input actions to the range $[-1, 1]$ and computing the MSE loss (Brantley et al., 2019). In Section 5.2, we first work on linear models and implement a deterministic offline learning oracle by outputting the Ordinary Least Squares (OLS) solution using the Moore-Penrose pseudoinverse (Moore, 1920). Later, in Section 5.3 and 5.4, we use MLP for base policies and implement $\mathcal{O}$ by conducting 2000 SGD iterations over its input dataset with batch size 200. See Appendix C.2 for results of 500 and 10000 iterations.

Sampling Oracle: We define the covering distribution d_0 as the uniform distribution over states obtained from 10 independent runs collected by DAGGER. Note that this gives MFTPL-P some unfair advantage over the baselines; we will subsequently propose practical variants of our algorithms that do not require knowledge of d_0. Additionally, in Section 5.2, we consider an alternative d_0, defined as the uniform distribution over state space.

Algorithms: Due to the sample-efficient nature of IL, we make the tasks more challenging by setting the sample size per round $K = 50$ for all algorithms (Ho and Ermon, 2016; Laskey et al., 2017). All policies in the first round π_1 are initialized at 0 for linear policy and at random for MLPs. We choose DAGGER and Behavior Cloning (BC) as online and offline IL baselines. At round n, BC receives K additional (state, action) pairs sampled from expert's trajectories and calls the offline learning oracle on the accumulated dataset. In contrast, all other algorithms sample K states from their current policy π_n's trajectories and query the expert's action on them, while following dataset aggregation and calling the offline learning oracle to compute policies π_{n+1} for the next round. As a practical implementation of MFTPL-P, we choose ensemble size $E = 25$; in addition, to facilitate parallel training of the ensembles (Brantley et al., 2019), instead of drawing sample sizes X from a Poisson distribution, we choose X as fixed numbers[†] – we abbreviate this algorithm as MP-25(X).

Evaluation: We run each algorithm 10 times with different seeds, treating each round n as the final one and only returning the last trained policy π_{n+1} for evaluation (Cheng et al., 2019a;b). As in common practice (Menda et al., 2019; Hoque et al., 2021; Menda et al., 2017), we return the ensemble mean $\bar{\pi}_n(s) := \frac{1}{E}\sum_{e=1}^{E} \pi_{n+1,e}(s)$, which is also known as Bagging (Breiman, 1996). Given a returned policy π, we roll out $T = 25$ trajectories (denote by $\left\{\tau_i^\pi\right\}_{i=1}^T$) and compute their average reward as an estimate of π's expected reward.

5.2. Utility of Sample-based Perturbation

We use linear policy classes along with OLS offline learning oracle for our first experiment. We study the impact of perturbation size X and the choice of d_0 on the performance of MP-25(X). Here, we choose DAGGER as the baseline; note that this is equivalent to MP-25(0) given that the offline learning oracle returns OLS solutions deterministically. We consider two settings of d_0 in Section 5.1.

The average reward of the trained policies as a function of the number of expert annotations for Ant and Hopper are shown in Figure 1 with 80% bootstrap confidence bands (DiCiccio and Efron, 1996). Surprisingly, though the expert (an MLP policy) is not contained in the policy class, MFTPL-P still learns policies with nontrivial performance. The overall performance of MP-25(X) initially increases with the perturbation size and then decreases, matching our intuition. For **Q1**, since MP-25(7) and MP-25(15) outperform DAGGER (MP-25(0)) in most cases, we have strong evidence that sample-based perturbation benefits performance with proper choices of perturbation sample size. For **Q2**, by comparing the performance of the same MP-25(X) on the left and right, it is evident that using states collected by DAGGER for perturbation results in better performance than uniform samples over state spaces. Based on our observations, we focus on evaluating MP-25(15) for the following sections. Please see Appendix C.3 for performance of other algorithms under this setting.

5.3. Performance Evaluation of MFTPL-P and Its Practical Variant BOOTSTRAP-DAGGER

Though MFTPL-P is provably efficient, it requires additional sample access to d_0 and proper choice of the perturbation sample size. We propose BOOTSTRAP-DAGGER

[†] It is well-known that Poission distribution has good concentration properties (e.g. Canonne, 2017). so we do not expect this to deviate too much from the original algorithm.

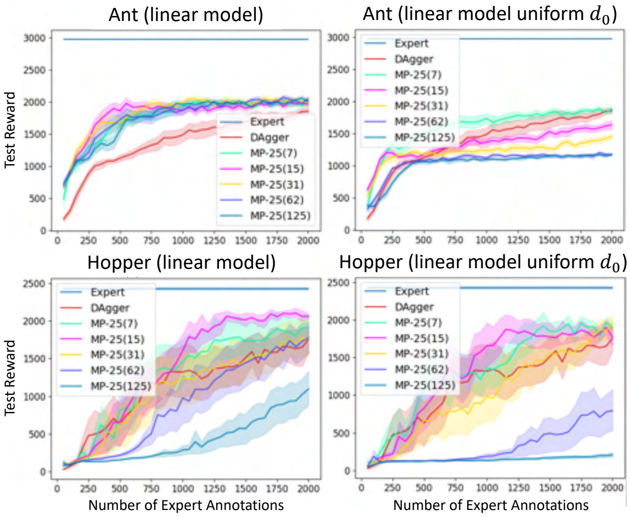

Figure 1: Comparative performance of MFTPL-P using linear models with nonrealizable MLP experts: variation across different perturbation state sources and set sizes in Ant and Hopper. Shaded region represents range between 10th and 90th quantiles of bootstrap confidence interval (DiCiccio and Efron, 1996), computed over 10 runs. On the left, the perturbation example sources are states collected by DAGGER on each task, while the right side uses uniform distribution over $[-2, 2]^{28}$ (Ant) and $[-2, 2]^{11}$ (Hopper). Overall, MP-25s on the left exceed their counterparts on the right. Meanwhile, MP-25(15) leads in performance, except in the Ant with uniform d_0 (upper right).

(abbrev. BD), a variant of MFTPL-P without sample access to d_0, and evaluate our algorithms in the 4 continuous control tasks in Section 5.1. BD shares the same data collection scheme as MFTPL-P and only differs on the TRAIN-BASE function.

As can be seen in the BD's TRAIN-BASE function in Algorithm 2 (see Appendix C.1 for the full BD algorithm), BD trains on different bootstrap subsamples of the accumulated dataset D to obtain a diverse ensemble of policies, instead of training on the accumulated dataset union with different sample-based perturbations. BD is fundamentally different from (Menda et al., 2019), where the diversity of ensembles are attributed solely to the stochasticity of SGD. In the following, we study the performance of BD with increasing size of ensembles, choosing $E = 1, 5, 25$ (abbreviated as BD-1,BD-5,BD-25).

We perform evaluations in realizable and non-realizable settings using MLPs as base policy classes. In the realizable setting, the base policy class contains the conditional mean function of the expert policy. Meanwhile, the non-realizable setting considers the base policy class to be

Algorithm 2 BOOTSTRAP-DAGGER

```
function TRAIN-BASE ($D$):
    $\tilde{D} \leftarrow$ Sample $|D|$ i.i.d. samples $\sim \mathrm{Unif}(D)$ with replacement.
    Return $h \leftarrow \mathcal{O}(\tilde{D})$.
```

MLPs with one hidden layer and limited numbers of nodes (see Appendix C.2 and C.4 for details).

As shown in Figure 2, MP-25(15) consistently outperforms others in most cases. Overall, BD shows a notable improvement in performance as the ensemble size grows, with BD-25 achieving performance on par with MP-25(15). Perhaps unsurprisingly, the naive BD-1 falls short of matching DAGGER's performance. This is attributed to the inherent limitations of bootstrapping, which omits a significant portion of the original sample. However, it is important to highlight the consistent and significant improvements from BD-1 to BD-5 across 4 tasks, as they demonstrate the effectiveness of using model ensembles to mitigate the sample underutilization from bootstrapping. Notice that the increase in performance from BD-5 to BD-25 is marginal, with BD-5 outperforming the baselines in all cases except in the realizable Hopper, where DAGGER achieves a similar level of performance. Interestingly, as shown in the lower part of Figure 2, MP-25(15), BD-25 and BD-5 not only learn faster than the baselines, but also converge to policies with higher performance.

For running time and space requirements, under realizable settings, all algorithms consume similar memory (1400 MB) on GPU, while BD-25 and MP-25(15) run 5 times longer than BD-5, BC, and DAGGER (see Appendix C.2 for details). Notably, BD-5 maintains strong performance without imposing significant computational overhead, taking just twice the running time of DAGGER. Therefore, we recommend using BD-5 for practical applications.

5.4. Explaining the benefit of BOOTSTRAP-DAGGER

Though BD-5 outperforms DAGGER, the underlying reason of this improvement demands further investigation. We hypothesize two possible factors behind BD-5's success: (1) BD-5 collects data of higher quality during the training stage; (2) Given the same expert demonstration dataset, BD-5 returns a better policy via ensemble averaging, similar to the benefit of Bagging in supervised learning (Breiman, 1996).

To test these, we evaluate two additional approaches: (a) naive supervised learning (abbreviated as SL) on data collected by BD-5; (b) Bagging (bootstrap and return 5-ensemble average) on data collected by DAgger. As shown in Figure 3, switching the final policy training between

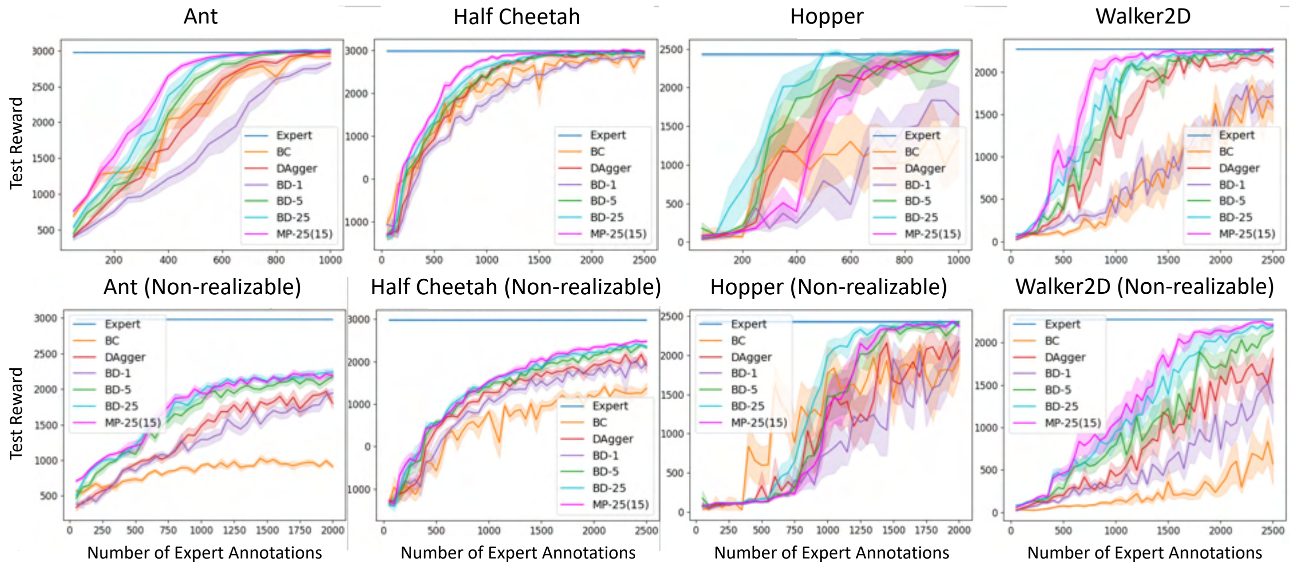

Figure 2: Results on continuous control tasks with realizable and non-realizable experts. Remarkably, MP-25(15) (magenta), BD-25 (blue-green) and BD-5 (green) surpass baselines under both settings, with distinct performance gaps particularly evident in the non-realizable setting between MP-25(15), BD-25, BD-5, and the baselines.

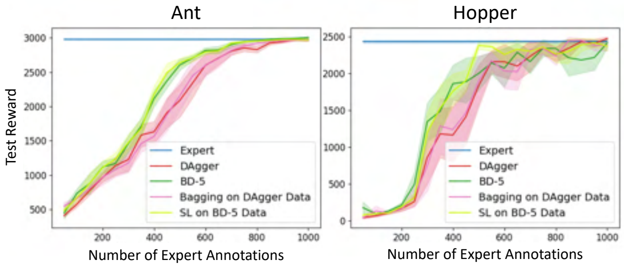

Figure 3: Results on comparing BD-5 and DAGGER, along with the two additional approaches in Section 5.4, over Ant and Hopper. Bagging on data collected by DAGGER yields pink learning curves that align closely with DAGGER's performance (red). Meanwhile, naive supervised learning on data collected by BD-5 produce lime green learning curves that match the performance of BD-5 (green). Overall the two methods (red and pink) that uses ensembles to perform data collection has better performance than those two that does not (green and lime green). This suggests that BD-5 improves over DAGGER by collecting better data.

Bagging and naive supervised learning does not change the policy performance significantly. In contrast, using ensemble for data collection significantly increases the trained policy's performance. This verifies hypothesis (1) and invalidates hypothesis (2). We further visualize states queried by different algorithms in Appendix C.5, which implies more efficient exploration by ensembles.

6. Conclusion

We propose and evaluate MFTPL-P, a computationally and statistically efficient IL algorithm for general policy classes. We also propose a practical variant, BOOTSTRAP-DAGGER that we recommend for practical applications.

Our work is built on the online imitation learning reduction framework (Ross et al., 2011; Ross and Bagnell, 2014). As we discuss in Appendix A, in the agnostic setting, this framework has the drawback of only providing runtime-dependent guarantees, as well as not ensuring global optimality. We leave overcoming these drawbacks as important open problems.

Impact Statement

This paper presents work whose goal is to advance the field of Imitation Learning. To the best of our knowledge, we are not aware of negative social impacts by our work.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback, and thank Yao Qin for helpful discussions on visualizing the observation distributions encountered in imitation learning data collection. This research is partially supported by the University of Arizona FY23 Eighteenth Mile TRIF Funding.

References

Jacob Abernethy, Chansoo Lee, Abhinav Sinha, and Ambuj Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory*, pages 807–823. PMLR, 2014.

J Andrew Bagnell. An invitation to imitation. *Robotics Inst., Carnegie-Mellon Univ., Pittsburgh, PA, USA, Tech. Rep*, 2015.

Normand J Beaudry and Renato Renner. An intuitive proof of the data processing inequality. *arXiv preprint arXiv:1107.0740*, 2011.

Alina Beygelzimer, Daniel J Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. *Advances in neural information processing systems*, 23, 2010.

Adam Block, Yuval Dagan, Noah Golowich, and Alexander Rakhlin. Smoothed online learning is as easy as statistical learning. In *Conference on Learning Theory*, pages 1716–1786. PMLR, 2022.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Kiante Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized imitation learning. In *International Conference on Learning Representations*, 2019.

Leo Breiman. Bagging predictors. *Machine learning*, 24: 123–140, 1996.

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

Clément Canonne. A short note on poisson tail bounds. *Retrieved from the website: http://www. cs. columbia. edu/ccanonne*, 2017.

Carlos Celemin, Rodrigo Pérez-Dattari, Eugenio Chisari, Giovanni Franzese, Leandro de Souza Rosa, Ravi Prakash, Zlatan Ajanović, Marta Ferraz, Abhinav Valada, Jens Kober, et al. Interactive imitation learning in robotics: A survey. *Foundations and Trends® in Robotics*, 10(1-2):1–197, 2022.

Jonathan Chang, Masatoshi Uehara, Dhruv Sreenivas, Rahul Kidambi, and Wen Sun. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems*, 34:965–979, 2021.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

Ching-An Cheng and Byron Boots. Convergence of value aggregation for imitation learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1801–1809. PMLR, 2018.

Ching-An Cheng, Xinyan Yan, Nathan Ratliff, and Byron Boots. Predictor-corrector policy optimization. In *International Conference on Machine Learning*, pages 1151–1161. PMLR, 2019a.

Ching-An Cheng, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Accelerating imitation learning with predictive models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3187–3196. PMLR, 2019b.

Sanjiban Choudhury. Cs 4756 spring 2024 assignment 1, 2024.

Yuchen Cui, David Isele, Scott Niekum, and Kikuo Fujimura. Uncertainty-aware data aggregation for deep imitation learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 761–767. IEEE, 2019.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient pac rl with rich observations. *Advances in neural information processing systems*, 31, 2018.

Hal Daumé, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine learning*, 75(3): 297–325, 2009.

Thomas J DiCiccio and Bradley Efron. Bootstrap confidence intervals. *Statistical science*, 11(3):189–228, 1996.

Miroslav Dudík, Nika Haghtalab, Haipeng Luo, Robert E Schapire, Vasilis Syrgkanis, and Jennifer Wortman Vaughan. Oracle-efficient online learning and auction design. *Journal of the ACM (JACM)*, 67(5):1–57, 2020.

Nika Haghtalab, Yanjun Han, Abhishek Shetty, and Kunhe Yang. Oracle-efficient online learning for beyond worst-case adversaries. *arXiv preprint arXiv:2202.08549*, 2022a.

Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis with adaptive adversaries. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 942–953. IEEE, 2022b.

Elad Hazan and Tomer Koren. The computational power of optimization in online learning. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 128–141, 2016.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021.

Zeyu Jia, Gene Li, Alexander Rakhlin, Ayush Sekhari, and Nati Srebro. When is agnostic reinforcement learning statistically tractable? *Advances in Neural Information Processing Systems*, 36, 2024.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 313–329. Springer, 2020.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pages 143–156. PMLR, 2017.

Yichen Li and Chicheng Zhang. On efficient online imitation learning via classification. *Advances in Neural Information Processing Systems*, 35:32383–32397, 2022.

Kunal Menda, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Dropoutdagger: A bayesian approach to safe imitation learning. *arXiv preprint arXiv:1709.06166*, 2017.

Kunal Menda, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Ensembledagger: A bayesian approach to safe imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5041–5048. IEEE, 2019.

Eliakim H Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the american mathematical society*, 26:294–295, 1920.

Rémi Munos. Error bounds for approximate policy iteration. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pages 560–567, 2003.

Rémi Munos. Performance bounds in l_p-norm for approximate value iteration. *SIAM journal on control and optimization*, 46(2):541–561, 2007.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos A Theodorou, and Byron Boots. Imitation learning for agile autonomous driving. *The International Journal of Robotics Research*, 39(2-3):286–302, 2020.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Jian Qian, Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. Concentration inequalities for multinoulli random variables. *arXiv preprint arXiv:2001.11595*, 2020.

Nived Rajaraman, Yanjun Han, Lin Yang, Jingbo Liu, Jiantao Jiao, and Kannan Ramchandran. On the value of interaction and function approximation in imitation learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Alexander Rakhlin and Karthik Sridharan. Bistro: An efficient relaxation-based method for contextual bandits. In *International Conference on Machine Learning*, pages 1977–1985. PMLR, 2016.

Sasha Rakhlin, Ohad Shamir, and Karthik Sridharan. Relax and randomize: From value to algorithms. *Advances in Neural Information Processing Systems*, 25, 2012.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

Ayush Sekhari, Karthik Sridharan, Wen Sun, and Runzhe Wu. Selective sampling and imitation learning via online regression, 2023.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

David Simchi-Levi and Yunzong Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*, 47(3):1904–1931, 2022.

Jonathan Spencer, Sanjiban Choudhury, Arun Venkatraman, Brian Ziebart, and J Andrew Bagnell. Feedback in imitation learning: The three regimes of covariate shift. *arXiv preprint arXiv:2102.02872*, 2021.

Jonathan Spencer, Sanjiban Choudhury, Matthew Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Sidd Srinivasa. Expert intervention learning: An online framework for robot learning from explicit and implicit human feedback. *Autonomous Robots*, pages 1–15, 2022.

Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *International Conference on Machine Learning*, pages 3309–3318. PMLR, 2017.

Wen Sun, Anirudh Vemula, Byron Boots, and Drew Bagnell. Provably efficient imitation learning from observation alone. In *International conference on machine learning*, pages 6036–6045. PMLR, 2019.

Gokul Swamy, Sanjiban Choudhury, J Andrew Bagnell, and Steven Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pages 10022–10032. PMLR, 2021.

Gokul Swamy, Nived Rajaraman, Matt Peng, Sanjiban Choudhury, J Bagnell, Steven Z Wu, Jiantao Jiao, and Kannan Ramchandran. Minimax optimal online imitation learning via replay estimation. *Advances in Neural Information Processing Systems*, 35:7077–7088, 2022.

Umar Syed and Robert E Schapire. A reduction from apprenticeship learning to classification. *Advances in neural information processing systems*, 23, 2010.

Vasilis Syrgkanis, Akshay Krishnamurthy, and Robert Schapire. Efficient algorithms for adversarial contextual learning. In *International Conference on Machine Learning*, pages 2159–2168. PMLR, 2016.

Csaba Szepesvári and Rémi Munos. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, pages 880–887, 2005.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

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

Tengyang Xie and Nan Jiang. Q* approximation schemes for batch reinforcement learning: A theoretical comparison. In *Conference on Uncertainty in Artificial Intelligence*, pages 550–559. PMLR, 2020.

Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. In *International Conference on Machine Learning*, pages 11404–11413. PMLR, 2021.

Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.

Agnostic Interactive Imitation Learning: New Theory and Practical Algorithms Supplementary Materials

A. The Online Imitation Learning Reduction Framework

Theorem 5 (Restatement of Theorem 2, originally from (Ross et al., 2011), Theorem 3.2)**.** *Suppose $(\mathcal{M}, \pi^{\exp})$ is μ-recoverable with respect to ℓ. In addition, a sequence of policies $\{\pi_n\}_{n=1}^N$ satisfies the following online regret guarantee with respect to base policy class $\mathcal{B}$:*

$$\sum_{n=1}^N F_n(\pi_n) - \min_{\pi \in \mathcal{B}} \sum_{n=1}^N F_n(\pi) \le \mathrm{Reg}(N).$$

Then $\hat{\pi}$, which is by choosing a policy uniformly at random from $\{\pi_n\}_{n=1}^N$ and adhering to it satisfies:

$$J(\hat{\pi}) - J(\pi^{\exp}) \le \mu H \left(\min_{\pi \in \mathcal{B}} \frac{1}{N} \sum_{n=1}^N F_n(\pi) + \frac{\mathrm{Reg}(N)}{N} \right).$$

Proof. Our proof is similar to Proposition 2 of (Li and Zhang, 2022). Since $(\mathcal{M}, \pi^{\exp})$ and ℓ satisfies for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $Q_{\pi^{\exp}}(s,a) - V_{\pi^{\exp}}(s) \le \mu \cdot \ell(a, \pi^{\exp}(s))$, We apply the performance difference lemma (Lemma 6 below) to the sequence of $\{\pi_n\}_{n=1}^N$ and $\pi^{\exp}$, obtaining

$$\begin{aligned}
\frac{1}{N} \sum_{n=1}^N J(\pi_n) - J(\pi^{\exp}) =& \frac{H}{N} \sum_{n=1}^N \mathbb{E}_{s \sim d_{\pi_n}} \mathbb{E}_{a \sim \pi_n(\cdot|s)} \left[Q_{\pi^{\exp}}(s,a) - V_{\pi^{\exp}}(s) \right] \\
\le& \frac{\mu H}{N} \sum_{n=1}^N \mathbb{E}_{s \sim d_{\pi_n}} \mathbb{E}_{a \sim \pi_n(\cdot|s)} \left[\ell(a, \pi^{\exp}(s)) \right] \\
=& \frac{\mu H}{N} \sum_{n=1}^N F_n(\pi_n) \le \mu H \left(\min_{\pi \in \mathcal{B}} \frac{1}{N} \sum_{n=1}^N F_n(\pi) + \frac{\mathrm{Reg}(N)}{N} \right),
\end{aligned}$$

where the last line comes from the definition of $F_n(\pi) := \mathbb{E}_{s \sim d_{\pi_n}, a \sim \pi(\cdot|s)} \ell(a, \pi^{\exp}(s))$ and $\mathrm{Reg}(N)$.

Now, it suffices to show $\frac{1}{N}\sum_{n=1}^N J(\pi_n) = J(\hat{\pi})$. Since $\hat{\pi}$ is executed by choosing a policy uniformly at random from $\{\pi_n\}_{n=1}^N$ and adhering to it, we conclude the proof by

$$J(\hat{\pi}) = \mathbb{E}_{s_1 \sim \rho} \left[V_{\hat{\pi}}(s_1) \right] = \mathbb{E}_{s_1 \sim \rho} \left[\frac{1}{N} \sum_{n=1}^N V_{\pi_n}(s_1) \right] = \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{s_1 \sim \rho} \left[V_{\pi_n}(s_1) \right] = \frac{1}{N} \sum_{n=1}^N J(\pi_n).$$

□

Lemma 6 (Performance Difference Lemma, Lemma 4.3 of (Ross and Bagnell, 2014))**.** *For two stationary policies π and $\pi^{\exp} : \mathcal{S} \to \Delta(\mathcal{A})$, we have*

$$J(\pi) - J(\pi^{\exp}) = H \cdot \mathbb{E}_{s \sim d_\pi} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[Q_{\pi^{\exp}}(s,a) - V_{\pi^{\exp}}(s) \right].$$

Limitations of the reduction-based framework. Our positive result relies on the reduction framework of (Ross et al., 2011), which bounds the learned policy's suboptimality by the sum of estimation gap and policy class bias $\mu H \min_{\pi \in \mathcal{B}} \frac{1}{N} \sum_{n=1}^{N} F_n(\pi)$ (Theorem 2). Importantly, the latter term is runtime dependent and one usually do not have a good control unless additional assumptions are imposed (e.g., there exists some $\pi \in \mathcal{B}$ that disagrees with π^E with low probability under some covering distribution d_0). We believe designing agnostic interactive imitation learning algorithms with runtime-independent guarantees is an important problem.

Moreover, we show in Proposition 7 below that it is possible in the agnostic setting that any no-regret policy sequence $\{\pi_n\}_{n=1}^N$ with respect to the cost-sensitive classification losses $\left\{F_n(\cdot)\right\}_{n=1}^N$ converges to a *globally suboptimal* policy with respect to the ground truth expected reward function $J(\cdot)$. We leave designing agnostic imitation learning algorithms with global optimality guarantees as an important question; without further assumption on the expert policy $\pi^{\exp}$, we believe this problem may be as hard as policy search-based agnostic reinforcement learning (Jia et al., 2024), where only limited positive results are currently known.

In the following, suppose we study the setting when the loss function $\ell(s,a) = A_{\pi^{\exp}}(s,a) := Q_{\pi^{\exp}}(s,a) - V_{\pi^{\exp}}(s)$; this is the setting initially studied by (Ross and Bagnell, 2014). As a result, $F_n(\pi) = \mathbb{E}_{s \sim d_{\pi_n}} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[Q_{\pi^{\exp}}(s,a) - V_{\pi^{\exp}}(s)\right]$, while $F_n(\pi_n) = \frac{1}{H}(J(\pi_n) - J(\pi^{\exp}))$ (by Lemma 6).

Proposition 7. *There exists a policy class $\mathcal{B}$ of size 2, an MDP $\mathcal{M}$, an expert policy π^E, such that any policy sequence $\{\pi_n\}_{n=1}^N \subseteq \mathcal{B}$ guaranteeing a sublinear regret*

$$\sum_{n=1}^{N} F_n(\pi_n) - \min_{\pi \in \mathcal{B}} \sum_{n=1}^{N} F_n(\pi) = o(N)$$

satisfies that

$$\sum_{n=1}^{N} J(\pi_n) - \min_{\pi \in \Pi} J(\pi) = \Omega(N)$$

Remark 8. *The above proposition can be generalized to allow $\{\pi_n\}_{n=1}^N \subseteq \Pi_{\mathcal{B}}$; the proof will carry over except that we argue that at least $1 - o(1)$ of the total weights in π_n's representation will be on h_2.*

Proof. As shown in Figure 4, we define MDP $\mathcal{M}$ with:

- State space $\mathcal{S} = \{S_0, S_1, S_2, S_3, S_4\}$ and action space $\mathcal{A} = \{L, R\}$.
- Initial state distribution $\rho(S_0) = 1$
- Deterministic Transition dynamics: $P_1(S_1|S_0, L) = 1$, $P_1(S_2|S_0, R) = 1, P_2(S_3|S_2, L) = 1, P_2(S_4|S_0, R) = 1$, while $\forall t \in [H]$, $\forall a \in \mathcal{A}$, $P_t(S_1|S_1, a) = P_t(S_3|S_3, a) = P_t(S_4|S_4, a) = 1$, which are self-absorbing before termination.
- Cost function $c(S_0, R) = c(S_2, L) = c(S_3, \cdot) = 0$, $c(S_0, L) = c(S_1, \cdot) = \frac{1}{H}$, $c(S_2, R) = c(S_4, \cdot) = 1$.

Meanwhile, let:

- Base policy class $\mathcal{B} = \{h_1, h_2\}$, where $h_1(S_0) = L$ and $h_2(S_0) = R$, while $h_1(S_2) = h_2(S_2) = R$.
- Deterministic expert $\pi^{\exp}$ such that $\pi^{\exp}(S_0) = R$, $\pi^{\exp}(S_2) = L$.

For this MDP example, it can be seen that $J(\pi^{\exp}) = 0$, $J(h_1) = 1$, $J(h_2) = H - 1$. Also, $V_\pi(S_0) = J(\pi^{\exp}) = 0$, $Q_{\pi^{\exp}}(S_0, L) = 1$, $Q_{\pi^{\exp}}(S_0, R) = 0$, we have $A_{\pi^{\exp}}(S_0, h_1(S_0)) = 1$, $A_{\pi^{\exp}}(S_0, h_2(S_0)) = 0$.

Consider any sequence of policy $\{\pi_n\}_{n=1}^N \subseteq \mathcal{B}$, inducing loss function $\left\{F_n(\pi)\right\}_{n=1}^N$. First, we observe that for every n, $\operatorname{argmin}_{\pi \in \mathcal{B}} F_n(\pi) = h_2$. This is because the only difference between h_1 and h_2 is the action taken at S_0, and so for any π_n, $F_n(h_1) - F_n(h_2) = \frac{1}{H}(A_{\pi^{\exp}}(S_0, h_1(S_0)) - A_{\pi^{\exp}}(S_0, h_2(S_0))) = \frac{1}{H} > 0$.

Therefore we conclude that given any any sequence $\{\pi_n\}_{n=1}^N$ that guarantees a sublinear regret, we have that at least $1-o(1)$ fraction of the π_n's must be h_2.

Then, for large enough N,

$$\sum_{n=1}^{N}\left(J(\pi_n)-\min_{\pi\in\mathcal{B}}J(\pi)\right)\geq\sum_{n=1}^{N}J(h_2)-o(N)-\sum_{n=1}^{N}\min_{\pi\in\mathcal{B}}J(\pi)=N(H-1)-N-o(N)=\Omega(N)$$

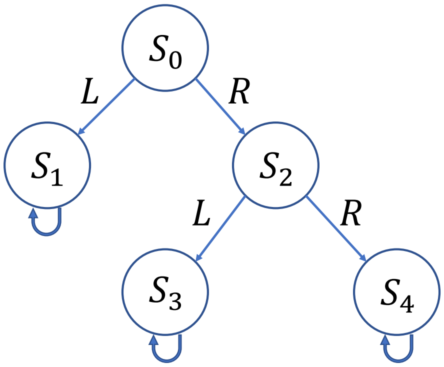

Figure 4: Example MDP to show the limit of reduction-based framework.

□

Remark 9. *We thank an anonymous ICML reviewer who originally provided an example (Choudhury, 2024, Problem (3)) on this issue. To put this in a real-world example, we can view this learning to ski from an expert demonstrator. The expert chooses a fast route $S_0\to S_2\to S_3\to..\to S_3$. Policy h_1 takes an "easy route" that deviates from the expert at step 1, and incurs a small but nonzero cost. Policy h_2 tries to mimic the expert by first choosing to take the fast route; however it fails to mimic the expert from step 2 on and incurs a catastrophically high cost. Although h_2 has a smaller imitation loss than h_1, both policies' inability to keep up with π^E subsequently makes h_1 actually a better choice.*

B. Proofs for Section 4

In the following, we provide detailed proofs for Theorem 3 and Corollary 4 in Section B.2. We first briefly review the interactive imitation learning for discrete action space setting in Section B.1.

B.1. Notations and algorithm

In this section, we first review some basic notations for interactive imitation learning introduced in Sections 3 and 4 and then introduce additional notations for our analysis.

Review of notations. The framework proposed by Ross et al.(Ross et al., 2011) reduces finding a policy $\hat{\pi}$ with a small performance gap compared to the expert policy $J(\hat{\pi})-J(\pi^{\exp})$ into minimization of online regret. As shown in Theorem 2, to find a policy competitive with $\pi^{\exp}$, it suffices to find a sequence of policies $\{\pi_n\}_{n=1}^N$ that optimize the regret defined as $\mathrm{Reg}(N)=\sum_{n=1}^N F_n(\pi_n)-\min_{\pi\in\mathcal{B}}\sum_{n=1}^N F_n(\pi)$, where $F_n(\pi):=\mathbb{E}_{s\sim d_{\pi_n}}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[I(a\neq\pi^{\exp}(s))\right]$.

We propose the MFTPL-P algorithm (Algorithm 1) to achieve sublinear regret, assuming sample access to some covering distribution d_0 (Assumption 2) that satisfies that for any $\pi\in\Pi_{\mathcal{B}}$ and $s\in\mathcal{S}$, $\frac{d_\pi(s)}{d_0(s)}\leq\frac{1}{\sigma}$. Meanwhile, we assume access to an offline classification oracle $\mathcal{O}$ (Assumption 1), which, given a (multi)set of classification examples, returns the policy in the base policy class that has the smallest empirical classification error.

Let $\mathcal{B}$ be the base policy class that contains B deterministic policies. For $u\in\Delta(\mathcal{B})$, define $u[h]$ as the coordinate of u corresponding to the $h\in\mathcal{B}$. Recall the definition of mixed policy class $\Pi_{\mathcal{B}}:=$

$\Big\{\pi_u(a|s) = \sum_{h\in\mathcal{B}} u[h]\cdot I(a=h(s)) : u\in\Delta(\mathcal{B})\Big\}$.

For completeness, we present Algorithm 3, which integrates the two functions in MFTPL-P for a more straightforward representation.

Algorithm 3 MFTPL-P (Mixed Following The Perturbed Leader with Poisson Perturbations)

```
1: Input: MDP $\mathcal{M}$, expert $\pi^{\exp}$, policy class $\mathcal{B}$, offline classification oracle $\mathcal{O}$, covering distribution $d_0$, sample size per iteration $K$, ensemble size $E$, perturbation budget $\lambda$.
2: Initialize $D = \emptyset$.
3: for $n = 1, 2, \ldots, N$ do
4:     for $e = 1, 2, \ldots, E$ do
5:         Sample $X_{n,e} \sim \mathrm{Poi}(\lambda)$.
6:         Sample $Q_{n,e} \leftarrow$ draw i.i.d. perturbation samples $\{(\tilde{s}_{n,e,x}, \tilde{a}_{n,e,x})\}_{x=1}^{X_{n,e}}$ from $\mathcal{D}_0 = d_0 \otimes \mathrm{Unif}(\mathcal{A})$.
7:         Compute $h_{n,e} \leftarrow \mathcal{O}(D \cup Q_{n,e})$.
8:     end for
9:     Set $\pi_n(a \mid s) := \frac{1}{E}\sum_{e=1}^{E} I(h_{n,e}(s) = a)$.
10:    $D_n = \big\{(s_{n,k}, \pi^{\exp}(s_{n,k}))\big\}_{k=1}^{K} \leftarrow$ sample $K$ states i.i.d. from $d_{\pi_n}$ by rolling out $\pi_n$ in $\mathcal{M}$, and query expert $\pi^{\exp}$ on these states.
11:    Aggregate datasets $D \leftarrow D \cup D_n$.
12: end for
13: Return $\pi_{\hat{n}}(a \mid s) := \frac{1}{E}\sum_{e=1}^{E} I(\pi_{\hat{n},e}(s) = a)$, where $\hat{n} \sim \mathrm{Unif}[N]$.
```

Additional notations. (Li and Zhang, 2022) provides a framework for designing and analyzing regret-efficient interactive imitation learning algorithm for discrete action spaces. In a nutshell, the framework views the original classification-based regret minimization problem over $\Pi_{\mathcal{B}}$ as an online linear optimization problem over $\Delta(\mathcal{B})$. Our design and analysis of MFTPL-P also adopt this framework, and thus we introduce the necessary notations in the context of MFTPL-P that facilitate this view.

In the following, we denote $\mathrm{Onehot}(h) \in \Delta(\mathcal{B})$ as the delta mass on a single policy h within the base policy class $\mathcal{B}$. We use $D_{1:n}$ as a shorthand for $\cup_{i=1}^{n} D_i$.

Using the notations π_u and Onehot, in line 7, we can write the policy returned from the oracle in the form of mixed policy, i.e. $h_{n,e} = \pi_{u_{n,e}}$, where $u_{n,e} = \mathrm{Onehot}(\mathcal{O}(D_{1:n-1} \cup Q_{n,e}))$.

We define $\mathcal{D}_{\pi}^{\exp}$ as the distribution of $(s, \pi^{\exp}(s))$, obtained by rolling out π in $\mathcal{M}$ and querying the expert $\pi^{\exp}$. Denote $g_n^* := \Big(\mathbb{E}_{s\sim d_{\pi_n}}\big[I(h(s) \neq \pi^{\exp}(s))\big]\Big)_{h\in\mathcal{B}}$, which is a B dimensional cost vector. We can rewrite $F_n(\pi_u)$ in the form of inner product as:

$$F_n(\pi_u) := \mathbb{E}_{s\sim d_{\pi_n}}\mathbb{E}_{a\sim\pi_u(\cdot|s)}\big[I(a \neq \pi^{\exp}(s))\big] = \mathbb{E}_{s\sim d_{\pi_n}}\sum_{h\in\mathcal{B}} u[h]\big[I(h(s) \neq \pi^{\exp}(s))\big] = \langle g_n^*, u\rangle\,.$$

Thus, the regret can be rewritten in an inner product form:

$$\mathrm{Reg}(N) = \sum_{n=1}^{N} F_n(\pi_n) - \min_{\pi\in\mathcal{B}}\sum_{n=1}^{N} F_n(\pi) = \sum_{n=1}^{N}\langle g_n^*, u_n\rangle - \min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n^*, u\rangle\,, \tag{2}$$

An equivalent representation of π_{n+1} (line 9) in the form of mixed policy is $\pi_{n+1} = \pi_{u_{n+1}}$, where $u_{n+1} = \frac{1}{E}\sum_{e=1}^{E} u_{n+1,e}$. By abusing D_n to denote the uniform distribution over it, we define

$$g_n := \Big(\mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\big[I(h(s) \neq \pi^{\exp}(s))\big]\Big)_{h\in\mathcal{B}}\,,\ \tilde{g}_{n,e} = \left(\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in Q_{n,e}}\left(I(h(\tilde{s}) \neq \tilde{a}) - \frac{A-1}{A}\right)\right)_{h\in\mathcal{B}}\,, \tag{3}$$

which stand for the cost vectors on D_n and $Q_{n,e}$ respectively. With these notations, we can rewrite u_n as a sample-average version of the "Follow-the-Perturbed-Leader" algorithm (Kalai and Vempala, 2005) over E independent trials:

$$u_n = \frac{1}{E}\sum_{e=1}^{E} \operatorname*{argmin}_{u\in\Delta(\mathcal{B})} \left\langle \sum_{i=1}^{n-1} g_i + \tilde{g}_{n,e}, u \right\rangle. \tag{4}$$

We give a formal proof of Eq. (4) in Lemma 10.

We define two $\sigma-$algebras for data and policies accumulated through the learning procedure of MFTPL-P:

$$\mathcal{F}_n := \sigma\left(u_1, D_1, u_2, D_2, \cdots u_n, D_n\right),\ \mathcal{F}_n^+ := \sigma\left(u_1, D_1, u_2, D_2, \cdots, u_n, D_n, u_{n+1}\right), \tag{5}$$

where it can be verified that filtration $(\mathcal{F}_n)_{n=1}^N$ and $(\mathcal{F}_n^+)_{n=1}^N$ satisfies $\mathcal{F}_1 \subset \mathcal{F}_1^+ \subset \mathcal{F}_2 \subset \mathcal{F}_2^+ \subset \cdots$.

Following the definition of perturbation sets $Q_{n,e}$ in Algorithm 3, given $\lambda > 0$, for any $n, n' \in [N]$ and any $e, e' \in [E]$, $Q_{n,e}$ and $Q_{n',e'}$ are equal in distribution. With this observation, we introduce a random variable Q_n that has the same distribution as $Q_{n,e}$ and

$$\tilde{g}_n = \left(\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in Q_n}\left(I(h(\tilde{s}) \neq \tilde{a}) - \frac{A-1}{A}\right)\right)_{h\in\mathcal{B}}$$

which has the same distribution as $\tilde{g}_{n,e}$. Without loss of generality, $\forall n \in [N], e \in [E]$, for any function f of $(Q_{n,e}, D_{1:n-1})$,we abbreviate $\mathbb{E}\left[f(Q_{n,e}, D_{1:n-1})|\mathcal{F}_{n-1}\right]$ as $\mathbb{E}_{Q_n}\left[f(Q_n, D_{1:n-1})\right]$ throughout and define

$$u_n^* := \mathbb{E}\left[u_{n,e}|\mathcal{F}_{n-1}\right] \tag{6}$$

Similar to Eq. (4) for u_n, we rewrite

$$u_n^* = \mathbb{E}_{Q_n}\left[\text{Onehot}(\mathcal{O}(D_{1:n-1}\cup Q_n)\right] = \mathbb{E}_{Q_n}\left[\operatorname*{argmin}_{u\in\Delta(\mathcal{B})}\left\langle \sum_{i=1}^{n-1} g_i + \tilde{g}_n, u\right\rangle\right],$$

Meanwhile, given any function f' of $(D_n, D_{1:n-1})$, we abbreviate $\mathbb{E}\left[f'(D_n, D_{1:n-1})|\mathcal{F}_{n-1}^+\right]$ as $\mathbb{E}_{D_n}\left[f'(D_n, D_{1:n-1})\right]$. We further define

$$u_{n+1}^{**} := \mathbb{E}\left[u_{n+1,e} \mid \mathcal{F}_{n-1}^+\right]. \tag{7}$$

By the law of iterated expectation, this can be also written as

$$u_{n+1}^{**} = \mathbb{E}\left[\mathbb{E}\left[u_{n+1,e}|\mathcal{F}_n\right]|\mathcal{F}_{n-1}^+\right] = \mathbb{E}\left[u_{n+1}^* \mid \mathcal{F}_{n-1}^+\right] = \mathbb{E}_{D_n}\left[u_{n+1}^*\right] = \mathbb{E}_{D_n}\mathbb{E}_{Q_{n+1,e}}\left[u_{n+1,e}\right] \tag{8}$$

where the second equality follows from the definition of u_{n+1}^*, and the third equality uses the observation that u_{n+1}^* is a function of $(D_n, D_{1:n-1})$, and the last equality is from that $u_{n+1}^* = \mathbb{E}_{Q_{n+1,e}}\left[u_{n+1,e}\right]$.

By this observation, u_{n+1}^{**} can be rewritten as

$$u_{n+1}^{**} = \mathbb{E}_{D_n}\mathbb{E}_{Q_{n+1}}\left[\operatorname*{argmin}_{u\in\Delta(\mathcal{B})}\left\langle \sum_{i=1}^{n-1} g_i + g_n + \tilde{g}_{n+1}, u\right\rangle\right].$$

We remark that the notations u_n, u_n^*, u_n^{**}, as well as $g_n, g_n^*, \tilde{g}_n$, are introduced solely for analytical purposes.

As a quick recap, we provide a dependency graph of important variables that appear in the analysis in Figure 5, while summarizing frequently-used notations in Table 2 below.

Table 2: A review of notations in this paper.

Name	**Description**	**Name**	**Description**
$\mathcal{M}$	Markov decision process	$\mathcal{O}$	Classification oracle
H	Episode length	$\Pi_{\mathcal{B}}$	Mixed policy class
t	Time step in $\mathcal{M}$	u	Ensemble policy probability weight
$\mathcal{S}$	State space	π_u	Ensemble policy induced by u
S	State space size	$u[h]$	Ensemble weight for h in u
s	State	u_n	Ensemble policy weight at round n
$\mathcal{A}$	Action space	K	Sample budget per round
A	Action space size	k	Sample iteration index
a	Action	D	Aggregated dataset
ρ	Initial distribution	D_n	Set of Classification examples at round n
P	Transition probability distribution	g_n	Loss vector induced by D_n
C	Cost distribution	$\mathcal{D}_{\pi}^{\exp}$	$(s, \pi^{\exp}(s))$ distribution induced by π, $\mathcal{M}$ and $\pi^{\exp}$
c	Cost	g_n^*	Expected loss vector induced by π_n, $\mathcal{M}$ and $\pi^{\exp}$
π	Stationary policy	d_0	Covering base distribution
$\pi(\cdot\|s)$	Action distribution of π given state s	$\mathcal{D}_0$	(s, a) distribution induced by $d_0 \otimes \mathrm{Unif}(\mathcal{A})$
d_π	State occupancy distribution	σ	Smooth factor
τ	Trajectory	E	Ensemble size
$J(\pi)$	Expected cumulative cost	e	Ensemble index
Q_π	Action value function	$\mathrm{Poi}(\lambda)$	Poisson distribution
V_π	State value function	λ	Perturbation budget
$\pi^{\exp}$	Expert policy	$X_{n,e}$	Perturbation set size
ℓ	Loss function	x	Sample index within a perturbation set
μ	Recoverability of $(\pi^{\exp}, M)$ for ℓ	$Q_{n,e}$	Perturbation set
N	Number of learning rounds	$\tilde{g}_{n,e}$	Perturbation loss vector in $\mathbb{R}^B$ induced by $Q_{n,e}$
n	Learning round index	$\mathcal{F}_n, \mathcal{F}_n^+$	σ-algebras induced by $\{u_i\}_{i=1}^n$ and $\{D_i\}_{i=1}^{n-1}$
$F_n(\pi)$	Online loss function	$\mathbb{E}_{D_n}$	Expectation w.r.t. $D_n \sim (D_{\pi_n}^{\pi^{\exp}})^K$
$\mathcal{B}$	Deterministic base policy class	$\mathbb{E}_{Q_n}$	Expectation w.r.t. $Q_n \sim (\mathcal{D}_0)^X$, where $X \sim \mathrm{Poi}(\lambda)$
B	Base policy class size	u_n^*	Expectation of u_n w.r.t Q_n
h	Deterministic stationary policy in $\mathcal{B}$	u_n^{**}	Expectation of u_n w.r.t Q_n and D_{n-1}
$\mathrm{Reg}(N)$	Online regret	$[N]$	Set $\{1, 2, \cdots, N\}$
$\mathrm{Unif}(\mathcal{E})$	Uniform distribution over $\mathcal{E}$	$\Delta(\mathcal{E})$	All probability distributions over $\mathcal{E}$
$\Pr(U)$	Probability of event U	$\mathrm{Onehot}(\mathcal{B})$	Delta mass (one-hot vector) on $h \in \mathcal{B}$
δ	Failure probability	$I(\cdot)$	Indicator function

B.1.1. Auxiliary Lemmas

Lemma 10. *$\pi_n = \pi_{u_n}$, where*

$$u_n = \frac{1}{E} \sum_{e=1}^{E} \operatorname*{argmin}_{u \in \Delta(\mathcal{B})} \left\langle \sum_{i=1}^{n-1} g_i + \tilde{g}_{n,e}, u \right\rangle.$$

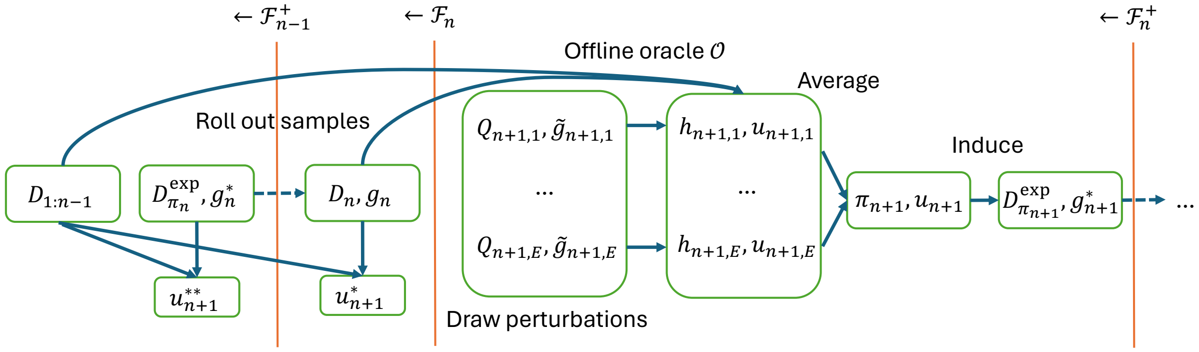

Figure 5: Dependency graph of notations that appear in the analysis. Solid and dashed arrows indicate deterministic and stochastic dependence, respectively. Note that all $(Q_{n+1,e})_{e\in[E]}$'s are drawn independently from fixed sample perturbation distributions and can be treated as fresh iid random examples.

Proof.

$$
\begin{aligned}
u_n =& \frac{1}{E}\sum_{e=1}^{E} u_{n,e} = \frac{1}{E}\sum_{e=1}^{E} \mathrm{Onehot}(\mathcal{O}(D_{1:n-1}\cup Q_{n,e})) \\
=& \frac{1}{E}\sum_{e=1}^{E} \mathrm{Onehot}(\operatorname*{argmin}_{h\in\mathcal{B}} \mathbb{E}_{(s,a)\sim D_{1:n-1}\cup Q_{n,e}}\left[I(h(s)\neq a)\right]) \\
=& \frac{1}{E}\sum_{e=1}^{E} \mathrm{Onehot}\left(\operatorname*{argmin}_{h\in\mathcal{B}}\left(\frac{1}{(n-1)K+X_{n,e}}\left(\sum_{i=1}^{n-1}\sum_{(s,a)\in D_i}(I(h(s)\neq a)) + \sum_{(\tilde{s},\tilde{a})\in Q_{n,e}}(I(h(\tilde{s})\neq\tilde{a}))\right)\right)\right) \\
=& \frac{1}{E}\sum_{e=1}^{E} \mathrm{Onehot}\left(\operatorname*{argmin}_{h\in\mathcal{B}}\left(\sum_{i=1}^{n-1}\mathbb{E}_{(s,a)\sim D_i}\left[I(h(s)\neq a)\right] + \frac{1}{K}\sum_{\sum_{(\tilde{s},\tilde{a})\in Q_{n,e}}}(I(h(\tilde{s})\neq\tilde{a}))\right)\right) \qquad (9) \\
=& \frac{1}{E}\sum_{e=1}^{E}\operatorname*{argmin}_{u\in\Delta(\mathcal{B})}\left\langle\left(\sum_{i=1}^{n-1}\mathbb{E}_{(s,a)\sim D_i}\left[I(h(s)\neq a)\right] + \frac{1}{K}\sum_{\sum_{(\tilde{s},\tilde{a})\in Q_{n,e}}}(I(h(\tilde{s})\neq\tilde{a}))\right)_{h\in\mathcal{B}}, u\right\rangle \\
=& \frac{1}{E}\sum_{e=1}^{E}\operatorname*{argmin}_{u\in\Delta(\mathcal{B})}\left\langle\left(\sum_{i=1}^{n-1}\mathbb{E}_{(s,a)\sim D_i}\left[I(h(s)\neq a)\right] + \frac{1}{K}\sum_{\sum_{(\tilde{s},\tilde{a})\in Q_{n,e}}}\left(I(h(\tilde{s})\neq\tilde{a}) - \frac{A-1}{A}\right)\right)_{h\in\mathcal{B}}, u\right\rangle \\
=& \frac{1}{E}\sum_{e=1}^{E}\operatorname*{argmin}_{u\in\Delta(\mathcal{B})}\left\langle\sum_{i=1}^{n-1} g_i + \tilde{g}_{n,e}, u\right\rangle,
\end{aligned}
$$

where we apply the invariant property of argmax operator on positive scaling and shifting. Note that $Q_{n,e}$ contains $X_{n,e}$ perturbation examples and each D_n contains K examples. □

B.2. Proof of Theorem 3

The proofs in this section follows the flowchart in Figure 6, which is divided to three stages:

- **At stage 1**, we apply the existing results (Li and Zhang, 2022) to reduce bounding the distribution-dependent online regret $\mathrm{Reg}(N) = \sum_{n=1}^{N} \langle g_n^*, u_n \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n^*, u \rangle$ to bounding the data-dependent online regret $\sum_{n=1}^{N} \langle g_n, u_n^* \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle$ using standard martingale concentration inequalities. By the end of stage 1, it remains to bound the regret of the idealized sequence of predictors $\{u_n^*\}_{n=1}^N$ on the observed linear losses $\{g_n\}_{n=1}^N$.
- **At stage 2**, a bound on the "ideal regret" is established by a standard analysis of an in-expectation version of the "Follow the perturbed Leader" algorithm. By Lemma 12 and 14, we prove that

$$\sum_{n=1}^{N} \langle g_n, u_n^* \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle \leq \mathbb{E}_{Q_1} \left[\max_{u \in \Delta(\mathcal{B})} \langle -\tilde{g}_1, u \rangle \right] + \sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^* \right\rangle .$$

  The first term on the right hand side can be straightforwardly bounded by Lemma 15. It remains to bound $\sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^* \right\rangle$.
- **At stage 3**, we aim to control $\sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^* \right\rangle$. Existing smoothed online learning analysis (Haghtalab et al., 2022a) (implicitly) provide bounds on $\mathbb{E}\left[\sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^* \right\rangle\right]$, which is insufficient for our goal of establishing high-probability bounds. Furthermore, Haghtalab et al. (2022a) only considers the online learning setting where one example is given at each round and the action space is binary, which is insufficient for batch mode multiclass online classification setting for our imitation learning application. To bridge the gap between existing techniques and our problem, we further decompose $\sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^* \right\rangle$ to three terms: stability term, generalization error, and an approximation term. Our analysis of stability term and generalization error generalizes the analysis of Haghtalab et al. (2022a) to multiclass batch setting (Lemmas 18 and 20). For the new approximation term, we observe that it has martingale structure and thus concentrates well (see proof of Lemma 17). With these, we have all terms bounded and conclude Theorem 3.

We provide a roadmap of our analysis of the the three stages in Figure 6, highlighting the key quantities and the key lemmas, as well as their relationships.

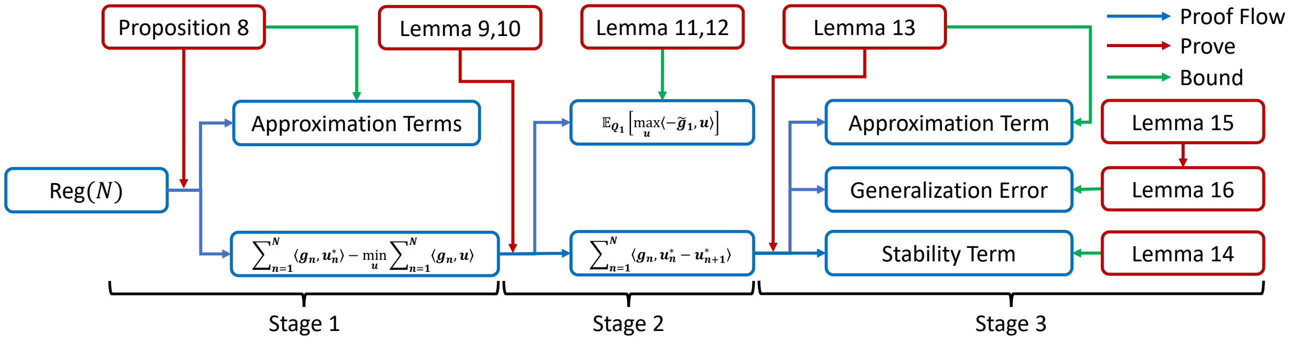
Proposition 8
Lemma 9,10
Lemma 11,12
Lemma 13
Proof Flow
Prove
Bound
Approximation Terms
$\mathbb{E}_{Q_1}\left[\max_u \langle -\tilde{g}_1, u \rangle\right]$
Approximation Term
Lemma 15
Reg(N)
Generalization Error
Lemma 16
$\sum_{n=1}^{N} \langle g_n, u_n^* \rangle - \min_u \sum_{n=1}^{N} \langle g_n, u \rangle$
$\sum_{n=1}^{N} \langle g_n, u_n^* - u_{n+1}^* \rangle$
Stability Term
Lemma 14
Stage 1
Stage 2
Stage 3

Figure 6: Flowchart of the proofs to bound the regret.

B.2.1. Proof for Stage 1

Following the previous results MFTPL, we guarantee that our algorithm MFTPL-P satisfies:

Proposition 11. *For any $\delta \in (0, 1]$,* MFTPL-P *outputs policies $\{\pi_n\}_{n=1}^N$ such that with probability at least $1 - \delta/2$,*

$$\mathrm{Reg}(N) \leq \sum_{n=1}^{N} \langle g_n, u_n^* \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle + O\left(\sqrt{\frac{N \ln(B/\delta)}{K}} \right) + N \sqrt{\frac{2A \left(\ln(NS) + \ln(\frac{12}{\delta}) \right)}{E}} .$$

Proof. By the inner product form of the regret (Eq. (2)), we have the following decomposition

$$
\begin{aligned}
\mathrm{Reg}(N) &= \sum_{n=1}^{N} \langle g_n^*, u_n \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n^*, u \rangle \\
&= \sum_{n=1}^{N} \langle g_n, u_n \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle + \sum_{n=1}^{N} \langle g_n^* - g_n, u_n \rangle + \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n^*, u \rangle \\
&\leq \sum_{n=1}^{N} \langle g_n, u_n \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle + \sqrt{\frac{2N \ln(\frac{12}{\delta})}{K}} + \sqrt{2N \frac{\ln(B) + \ln(\frac{12}{\delta})}{K}} \\
&= \sum_{n=1}^{N} \langle g_n, u_n^* \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle + \sum_{n=1}^{N} \langle g_n, u_n - u_n^* \rangle + O\left(\sqrt{\frac{N \ln(B/\delta)}{K}} \right) \\
&\leq \sum_{n=1}^{N} \langle g_n, u_n^* \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle + O\left(\sqrt{\frac{N \ln(B/\delta)}{K}} \right) + N \sqrt{\frac{2A \left(\ln(NS) + \ln(\frac{12}{\delta}) \right)}{E}},
\end{aligned}
$$

where the first and second inequalities are from propositions 24 and 25 in Appendix B.3 respectively, which in turn are from the proposition 6 and the proof of lemma 8 in (Li and Zhang, 2022). □

B.2.2. Proof for Stage 2

In this section, we prove a bound on the regret of the "idealized policy sequence" $\left\{ \pi_{u_n^*} \right\}_{n=1}^{N}$, i.e., $\sum_{n=1}^{N} \langle g_n, u_n^* \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle$. Such result should be well-known in the context of analysis of the "Follow the Perturbed Leader" algorithm in online linear optimization (Kalai and Vempala, 2005); we provide full details here since we cannot find in the literature this exact lemma statement we need. An in-expectation version of a similar bound has been implicitly shown in Haghtalab et al. (2022a). in the language of admissible relaxations (Rakhlin et al., 2012).

Lemma 12. *For g_n induced by* MFTPL-P*, MDP $\mathcal{M}$ and expert $\pi^{\exp}$, the sequence of u_n^* defined in equation (6) satisfies*

$$
\sum_{n=1}^{N} \langle g_n, u_n^* \rangle - \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle \leq \mathbb{E}_{Q_1} \left[\max_{u \in \Delta(\mathcal{B})} \langle -\tilde{g}_1, u \rangle \right] + \sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^* \right\rangle, \tag{10}
$$

where $\tilde{g}_1 := \left(\frac{1}{K} \sum_{(\tilde{s}, \tilde{a}) \in Q_1} \left(I(h(\tilde{s}) \neq \tilde{a}) - \frac{A-1}{A} \right) \right)_{h \in \mathcal{B}}$, $Q_1 := \left\{ \{(\tilde{s}_{1,x}, \tilde{a}_{1,x})\}_{x=1}^{X} : X_1 \sim Poi(\lambda), (\tilde{s}_x, \tilde{a}_x) \sim \mathcal{D}_0 \right\}$.

Remark 13. *Intuitively, the right hand side of Eq. (10) exhibits a bias-variance tradeoff: $\mathbb{E}_{Q_1} \left[\max_{u \in \Delta(\mathcal{B})} \langle -\tilde{g}_1, u \rangle \right]$ and $\sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^* \right\rangle$ are the "bias" and "variance" terms that increases and decrease with respect to the amount of perturbation noise λ, respectively. We bound the first term in Lemma 15 and the second term in Section B.2.3, respectively.*

Proof. Notice that $\sum_{n=1}^{N} \langle g_n, u_n^* \rangle$ appears on both sides of equation (10), by arranging the terms, it suffices to show

$$
\sum_{n=1}^{N} \left\langle g_n, u_{n+1}^* \right\rangle \leq \min_{u \in \Delta(\mathcal{B})} \sum_{n=1}^{N} \langle g_n, u \rangle + \mathbb{E}_{Q_1} \left[\max_{u \in \Delta(\mathcal{B})} \langle -\tilde{g}_1, u \rangle \right]
$$

By Lemma 14, we have $\forall n \in [N]$,

$$
\left\langle g_n, u_{n+1}^* \right\rangle \leq \mathbb{E}_{Q_n} \left[\max_{u \in \Delta(\mathcal{B})} \left\langle -\sum_{i=1}^{n-1} g_i - \tilde{g}_n, u \right\rangle \right] - \mathbb{E}_{Q_{n+1}} \left[\max_{u \in \Delta(\mathcal{B})} \left\langle -\sum_{i=1}^{n} g_i - \tilde{g}_{n+1}, u \right\rangle \right].
$$

Summing the above inequality over $n \in [N]$ (and noting that the right hand side is a telescoping sum) gives that

$$
\sum_{n=1}^{N} \left\langle g_n, u_{n+1}^* \right\rangle \leq \mathbb{E}_{Q_1} \left[\max_{u \in \Delta(\mathcal{B})} \langle -\tilde{g}_1, u \rangle \right] - \mathbb{E}_{Q_{N+1}} \left[\max_{u \in \Delta(\mathcal{B})} \left\langle -\sum_{n=1}^{N} g_n - \tilde{g}_{N+1}, u \right\rangle \right], \tag{11}
$$

Meanwhile, we further notice that

$$\mathbb{E}_{Q_{N+1}}\left[\max_{u\in\Delta(\mathcal{B})}\left\langle -\sum_{n=1}^{N} g_n - \tilde{g}_{N+1}, u\right\rangle\right] \geq \max_{u\in\Delta(\mathcal{B})}\mathbb{E}_{Q_{N+1}}\left[\left\langle -\sum_{n=1}^{N} g_n - \tilde{g}_{N+1}, u\right\rangle\right] = \max_{u\in\Delta(\mathcal{B})}\left\langle -\sum_{n=1}^{N} g_n, u\right\rangle, \tag{12}$$

where we apply Jensen's inequality and observe that $\forall s \in \mathcal{S}, h \in \mathcal{B}$, $\mathbb{E}_{y\sim\mathrm{Unif}(\mathcal{A})}\left[I(h(s) \neq y)\right] = (A-1)/A$, meaning that $\forall u \in \Delta(\mathcal{B})$,

$$\mathbb{E}_{Q_{N+1}}\left[\langle -\tilde{g}_{N+1}, u\rangle\right] = \mathbb{E}_{Q_{N+1}}\left[\left\langle -\left(\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in Q_{N+1}}\Big(I(h(\tilde{s}) \neq \tilde{a}) - (A-1)/A\Big)\right)_{h\in\mathcal{B}}, u\right\rangle\right] = 0.$$

Therefore, we conclude the proof by plugging equation (12) in (11):

$$\sum_{n=1}^{N}\left\langle g_n, u^*_{n+1}\right\rangle \leq \mathbb{E}_{Q_1}\left[\max_{u\in\Delta(\mathcal{B})}\langle -\tilde{g}_1, u\rangle\right] - \max_{u\in\Delta(\mathcal{B})}\left\langle -\sum_{n=1}^{N} g_n, u\right\rangle = \mathbb{E}_{Q_1}\left[\max_{u\in\Delta(\mathcal{B})}\langle -\tilde{g}_1, u\rangle\right] + \min_{u\in\Delta(\mathcal{B})}\left\langle \sum_{n=1}^{N} g_n, u\right\rangle.$$

□

Lemma 14. *For $\{g_n\}_{n=1}^N$ induced by* MFTPL-P *and $\{u^*_n\}_{n=1}^N$ defined in Eq. (6),*

$$\left\langle g_n, u^*_{n+1}\right\rangle \leq \mathbb{E}_{Q_n}\left[\max_{u\in\Delta(\mathcal{B})}\left\langle -\sum_{i=1}^{n-1} g_i - \tilde{g}_n, u\right\rangle\right] - \mathbb{E}_{Q_{n+1}}\left[\max_{u\in\Delta(\mathcal{B})}\left\langle -\sum_{i=1}^{n} g_i - \tilde{g}_{n+1}, u\right\rangle\right].$$

Proof. Note that Q_n and Q_{n+1} have identical probability distributions. Therefore, the lemma statement is equivalent to:

$$\left\langle g_n, u^*_{n+1}\right\rangle \leq \mathbb{E}_{Q}\left[\max_{u\in\Delta(\mathcal{B})}\left\langle -\sum_{i=1}^{n-1} g_i - \tilde{g}, u\right\rangle\right] - \mathbb{E}_{Q}\left[\max_{u\in\Delta(\mathcal{B})}\left\langle -\sum_{i=1}^{n} g_i - \tilde{g}, u\right\rangle\right],$$

where $\tilde{g} := \left(\frac{1}{K}\sum_{(\tilde{s},\tilde{a})\in Q}\left(I(h(\tilde{s}) \neq \tilde{a}) - \frac{A-1}{A}\right)\right)_{h\in\mathcal{B}}$.

By the definition of u^*_n in equation (6), we have:

$$u^*_n = \mathbb{E}_{Q}\left[\operatorname*{argmin}_{u\in\Delta(\mathcal{B})}\left\langle \sum_{i=1}^{n-1} g_i + \tilde{g}, u\right\rangle\right] = \mathbb{E}_{Q}\left[\operatorname*{argmax}_{u\in\Delta(\mathcal{B})}\left\langle \sum_{i=1}^{n-1} -g_i - \tilde{g}, u\right\rangle\right].$$

By denoting $u_{n,Q} := \mathrm{argmax}_{u \in \Delta(\mathcal{B})} \left\langle -\sum_{i=1}^{n-1} g_i - \tilde{g}, u \right\rangle$, we notice that $\mathbb{E}_Q \left[u_{n,Q} \right] = u_n^*$and write:

$$\begin{aligned}
\mathbb{E}_Q \left[\max_{u \in \Delta(\mathcal{B})} \left\langle -\sum_{i=1}^{n-1} g_i - \tilde{g}, u \right\rangle \right] + \left\langle -g_n, u_{n+1}^* \right\rangle =& \mathbb{E}_Q \left[\left\langle -\sum_{i=1}^{n-1} g_i - \tilde{g}, u_{n,Q} \right\rangle \right] + \left\langle -g_n, u_{n+1}^* \right\rangle \\
\geq& \mathbb{E}_Q \left[\left\langle -\sum_{i=1}^{n-1} g_i - \tilde{g}, u_{n+1,Q} \right\rangle \right] + \left\langle -g_n, u_{n+1}^* \right\rangle \\
=& \mathbb{E}_Q \left[\left\langle -\sum_{i=1}^{n-1} g_i - \tilde{g}, u_{n+1,Q} \right\rangle \right] + \mathbb{E}_Q \left[\left\langle -g_n, u_{n+1,Q} \right\rangle \right] \\
=& \mathbb{E}_Q \left[\left\langle -\sum_{i=1}^{n} g_i - \tilde{g}, u_{n+1,Q} \right\rangle \right] \\
=& \mathbb{E}_Q \left[\max_{u \in \Delta(\mathcal{B})} \left\langle -\sum_{i=1}^{n} g_i - \tilde{g}, u \right\rangle \right]
\end{aligned}$$

where the inequality is by the optimality of $u_{n,Q}$. We conclude our proof by rearranging the terms. □

Lemma 15.

$$\mathbb{E}_{Q_1} \left[\max_{u \in \Delta(\mathcal{B})} \langle -\tilde{g}_1, u \rangle \right] \leq \sqrt{\frac{\lambda \ln(B)}{2K^2}}$$

Proof. We first recall the definition of $\tilde{g}_1 = \left(\frac{1}{K} \sum_{(\tilde{s}, \tilde{a}) \in Q_1} \left(I(h(\tilde{s}) \neq \tilde{a}) - \frac{A-1}{A} \right) \right)_{h \in \mathcal{B}}$ in equation (3) and rewrite

$$\begin{aligned}
\mathbb{E}_{Q_1} \left[\max_{u \in \Delta(\mathcal{B})} \langle -\tilde{g}_1, u \rangle \right] =& \mathbb{E}_{Q_1} \left[\max_{h \in \mathcal{B}} \frac{1}{K} \sum_{(\tilde{s}, \tilde{a}) \in Q_1} \left(\frac{A-1}{A} - I(h(\tilde{s}) \neq \tilde{a}) \right) \right] \\
=& \frac{1}{K} \mathbb{E}_{Q_1} \left[\max_{h \in \mathcal{B}} \left(X_1 \frac{A-1}{A} - \sum_{(\tilde{s}, \tilde{a}) \in Q_1} I(h(\tilde{s}) \neq \tilde{a}) \right) \right] \qquad (13)
\end{aligned}$$

where the size of Q_1 is denoted by X_1. When $(\tilde{s}, \tilde{a}) \sim \mathcal{D}_0$, $\tilde{a} \sim \mathrm{Unif}(\mathcal{A})$, therefore, it is not hard to see $\forall h \in \mathcal{B}$,

$$\mathbb{E}_{(\tilde{s}, \tilde{a}) \sim \mathcal{D}_0} \left[I(h(\tilde{s}') \neq \tilde{a}') \right] = \frac{A-1}{A}.$$

Thus, conditioned on X_1, for every $h \in \mathcal{B}$, $X_1 \frac{A-1}{A} - \sum_{(\tilde{s}, \tilde{a}) \in Q_1} I(h(\tilde{s}) \neq \tilde{a})$ is a zero-mean, $\frac{X_1}{4}$-subgaussian random variable. Therefore, Massart's Lemma (see Lemma 16 below) implies that

$$\mathbb{E}_{Q_1} \left[\max_{h \in \mathcal{B}} \left(X_1 \frac{A-1}{A} - \sum_{(\tilde{s}, \tilde{a}) \in Q_1} I(h(\tilde{s}) \neq \tilde{a}) \right) \mid X_1 \right] \leq \sqrt{\frac{X_1 \ln B}{2}}.$$

By the law of iterated expectation and $\mathbb{E}\left[X_1 \right] = \lambda$, we have that Eq. (13) can be bounded by

$$\frac{1}{K} \mathbb{E} \left[\sqrt{\frac{X_1 \ln B}{2}} \right] \leq \frac{1}{K} \left[\sqrt{\frac{\mathbb{E}\left[X_1 \right] \ln B}{2}} \right] = \sqrt{\frac{\lambda \ln B}{2K^2}}.$$ □

Lemma 16 (Massart's Lemma (Lemma 26.8 of Shalev-Shwartz and Ben-David (2014)))**.** *Suppose $X_1, \ldots, X_B$ is collection of zero-mean, σ^2-subgaussian random variables. Then,*

$$\mathbb{E} \left[\max_{i=1}^{B} X_i \right] \leq \sigma \sqrt{2 \ln(B)}.$$

B.2.3. Proof for Stage 3

Lemma 17. *For any $\delta \in [0,1]$, $\lambda \geq \max\left\{\frac{2AK^2}{\sigma}, \frac{8AK\ln(KN)}{\sigma}\right\}$, the sequence of $\{g_n\}$ defined in equation (3) and $\{u_n^*\}$ defined in equation (6) satisfies that with probability at least $1-\delta/2$,*

$$\sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^* \right\rangle \leq N\sqrt{\frac{AK^2}{\lambda\sigma}} + N\sqrt{\frac{2A\ln(N)\ln(BN^2)}{\lambda\sigma}} + \frac{\lambda\sigma}{4AN\ln(N)} + 4\sqrt{2N\ln(4/\delta)}. \tag{14}$$

Proof. To begin with, we recall that $u_{n+1}^{**} = \mathbb{E}_{D_n}\left[u_{n+1}^*\right]$ as shown in Eq. (8) and decompose $\sum_{n=1}^{N}\left\langle g_n, u_n^* - u_{n+1}^*\right\rangle$ into three parts as follows:

$$\begin{aligned}
\sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^* \right\rangle &= \sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^{**} \right\rangle + \sum_{n=1}^{N} \left\langle g_n, u_{n+1}^{**} - u_{n+1}^* \right\rangle \\
&= \underbrace{\sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^{**} \right\rangle}_{\text{Stability Term}} + \underbrace{\sum_{n=1}^{N} \mathbb{E}_{D_n}\left[\left\langle g_n^* - g_n, u_{n+1}^* \right\rangle\right]}_{\text{Generalization Error}} \\
&\quad + \underbrace{\sum_{n=1}^{N} \left\langle g_n, u_{n+1}^{**} - u_{n+1}^* \right\rangle - \sum_{n=1}^{N} \mathbb{E}_{D_n}\left[\left\langle g_n^* - g_n, u_{n+1}^* \right\rangle\right]}_{\text{Approximation Term}}.
\end{aligned} \tag{15}$$

As shown in equation (15), we apply a decomposition similar to Lemma 4.4 of Haghtalab et al. (2022a), which also involves a stability term and a generalization error term. Our decomposition uniquely introduces a new approximation term due to the need in establishing high probability regret bounds. We generalize Haghtalab et al. (2022a) to multi-class classification. By Lemma 18 (deferred after this proof), the stability term satisfies

$$\sum_{n=1}^{N} \left\langle g_n, u_n^* - u_{n+1}^{**} \right\rangle \leq N\sqrt{\frac{AK^2}{\lambda\sigma}}.$$

Similarly, we follow the proof idea of Lemma 4.6 of Haghtalab et al. (2022a) and bound the generalization error by Lemma 20 (deferred after this proof):

$$\sum_{n=1}^{N} \mathbb{E}_{D_n}\left[\left\langle g_n^* - g_n, u_{n+1}^* \right\rangle\right] \leq N\sqrt{\frac{2A\ln(N)\ln(BN^2)}{\lambda\sigma}} + \frac{\lambda\sigma}{4AN\ln(N)} + NKe^{-\frac{\lambda}{8K}}$$

by our assumption that $\lambda \geq \frac{8AK\ln(KN)}{\sigma}$, the last term $NKe^{-\frac{\lambda}{8K}} \leq 1$.

In the following, we bound the approximation term. Before going into details, we first show that g_n^*, u_{n+1}^{**} are functions of $(u_n, D_{1:n-1})$ and is thus is $\mathcal{F}_{n-1}^+$-measurable. Indeed,

$$g_n^* = \left(\mathbb{E}_{s\sim d_{\pi_n}}\left[I(h(s) \neq \pi^{\exp}(s))\right]\right)_{h\in\mathcal{B}} = \mathbb{E}_{D_n}\left[\left(\mathbb{E}_{s\sim D_n}\left[I(h(s) \neq \pi^{\exp}(s))\right]\right)_{h\in\mathcal{B}}\right] = \mathbb{E}_{D_n}\left[g_n\right],$$

which is a function of u_n . Similarly,

$$u_{n+1}^{**} = \mathbb{E}_{D_n}\left[u_{n+1}^*\right] = \mathbb{E}_{D_n}\mathbb{E}_{Q_{n+1}}\left[\text{Onehot}(\mathcal{O}(D_{1:n-1} \cup D_n \cup Q_{n+1})\right]$$

which is a function of u_n and $D_{1:n-1}$.

Approximation Term: Define $Y_n := \left\langle g_n, u_{n+1}^{**} - u_{n+1}^* \right\rangle - \mathbb{E}_{D_n}\left[\left\langle g_n^* - g_n, u_{n+1}^* \right\rangle\right]$. In the following, we show that $\{Y_n\}_{n=1}^N$ is a martingle difference sequence with respect to filtration $\left\{\mathcal{F}_{n-1}^+\right\}_{n=1}^N$, i.e., $\forall n \in \mathbb{N}$, $\mathbb{E}\left[Y_n | \mathcal{F}_{n-1}^+\right] = 0$. First,

$$\mathbb{E}\left[\left\langle g_n, u_{n+1}^{**}\right\rangle \mid \mathcal{F}_{n-1}^+\right] = \left\langle \mathbb{E}\left[g_n \mid \mathcal{F}_{n-1}^+\right], u_{n+1}^{**}\right\rangle = \left\langle g_n^*, u_{n+1}^{**}\right\rangle = \left\langle g_n^*, \mathbb{E}_{D_n}[u_{n+1}^*]\right\rangle = \mathbb{E}_{D_n}\left[\left\langle g_n^*, u_{n+1}^*\right\rangle\right]$$

where the first equality is from that u^{**}_{n+1} is $\mathcal{F}^+_{n-1}$-measurable and linearity of expectation. Second,

$$\mathbb{E}\left[\left\langle g_n, u^*_{n+1}\right\rangle \mid \mathcal{F}^+_{n-1}\right] = \mathbb{E}_{D_n}\left[\left\langle g_n, u^*_{n+1}\right\rangle\right],$$

since conditioned on $\mathcal{F}^+_{n-1} = \sigma(u_1, D_1, \ldots, u_{n-1}, D_{n-1}, u_n)$, the only randomness in the expression $\left\langle g_n, u^*_{n+1}\right\rangle$ comes from their dependence on D_n.

Together we have,

$$\begin{aligned}
\mathbb{E}\left[Y_n|\mathcal{F}^+_{n-1}\right] =& \mathbb{E}_{D_n}\left[\left\langle g^*_n, u^*_{n+1}\right\rangle\right] - \mathbb{E}_{D_n}\left[\left\langle g_n, u^*_{n+1}\right\rangle\right] - \mathbb{E}\left[\mathbb{E}_{D_n}\left[\left\langle g^*_n - g_n, u^*_{n+1}\right\rangle\right] |\mathcal{F}^+_{n-1}\right] \\
=& \mathbb{E}_{D_n}\left[\left\langle g_n, u^{**}_{n+1}\right\rangle\right] - \mathbb{E}_{D_n}\left[\left\langle g_n, u^*_{n+1}\right\rangle\right] - \mathbb{E}_{D_n}\left[\left\langle g^*_n, u^*_{n+1}\right\rangle\right] + \mathbb{E}_{D_n}\left[\left\langle g_n, u^*_{n+1}\right\rangle\right] \\
=& 0.
\end{aligned}$$

Meanwhile, since each entry of g_n and g^*_n are upper-bounded by 1 and lower-bounded by 0, we have

$$|Y_n| \leq \|g_n\|_\infty \cdot \|u^{**}_{n+1} - u^*_{n+1}\|_1 + \mathbb{E}_{D_n}\left[\|g_n - g^*_n\|_\infty \cdot \|u^*_{n+1}\|_1\right] \leq 2 + 1 = 3.$$

With the martingale difference sequence conditions satisfied, by Azuma-Hoeffding's inequality, for any $\delta \in (0, 1]$, with probability $1 - \delta/2$,

$$\left|\sum_{n=1}^{N}\left(\left\langle g_n, u^{**}_{n+1} - u^*_{n+1}\right\rangle - \mathbb{E}_{D_n}\left[\left\langle g^*_n - g_n, u^*_{n+1}\right\rangle\right]\right)\right| = \left|\sum_{n=1}^{N} Y_n\right| \leq 3\sqrt{2N\ln(4/\delta)}.$$

Combining bounds for three terms in equation (15), we conclude the proof. □

Lemma 18. *Under the notation of* MFTPL-P, *when* $\lambda \geq \frac{2AK^2}{\sigma}$, $\forall n \in \mathbb{N}$, g_n,u^*_n,u^{**}_{n+1} *defined in equation* (3), (6), *and* (7) *satisfies*

$$\left\langle g_n, u^*_n - u^{**}_{n+1}\right\rangle \leq 2\sqrt{\frac{AK^2}{\lambda\sigma}}.$$

Proof. Since $\|g_n\|_\infty \leq 1$, it is straightforward to see that

$$\left\langle g_n, u^*_n - u^{**}_{n+1}\right\rangle \leq \|g_n\|_\infty \cdot \|u^*_n - u^{**}_{n+1}\|_1 \leq \|u^*_n - u^{**}_{n+1}\|_1.$$

Our proof structure is similar to Lemma 4.4 and Lemma 4.5 of Haghtalab et al. (2022a), where we bound $\|u^*_n - u^{**}_{n+1}\|_1$ by the discrepancy of distributions of two datasets. We generalize the results of Haghtalab et al. (2022a) to multiclass classification and online learning with batches of K samples at each round to keep track of the number of copies of each $(s, a) \in \mathcal{S} \times \mathcal{A}$.

The main technical challenge here lies in using batches of examples. While in the batch size 1 case, Haghtalab et al. (2022a) reduced bounding $\|u^*_n - u^{**}_{n+1}\|$ to bounding the discrepancy between an SA-dimensional product Poisson distribution and its one-sample shifted version, the same approach becomes difficult to compute when dealing with batches of more than one examples. Specifically, a straightforward calculation leads to the total variation (TV) distance between an SA-dimensional product Poisson distribution and a mixture of product shifted Poisson distributions, where the shifts are drawn from a multinomial distribution. This mixture significantly complicates the computation, making it a much harder to solve and present. [‡]

[‡] Concretely, we can define $p_{n,e}$ to be a SA dimensional random variable that represents the "histogram" of all examples in $\cup_{i=1}^{n-1} D_i \cup Q_{n,e}$; specifically, $p_{n,e}(s, a) = \sum_{i=1}^{n-1}\sum_{(s',a')\in D_i} I(s = s', a = a') + \sum_{(s',a')\in Q_{n,e}} I(s = s', a = a')$. By the data-processing inequality, $\|u^*_n - u^{**}_{n+1}\|_1$ is upper bounded by the TV distance between $(p_{n,e}(s, a))_{s\in\mathcal{S},a\in\mathcal{A}} \mid \mathcal{F}_{n-1}$ and $(p_{n+1,e}(s, a))_{s\in\mathcal{S},a\in\mathcal{A}} \mid \mathcal{F}^+_{n-1}$, which is equal to the TV distance between a product Poisson and a mixture of product shifted Poisson distribution.

We work around this challenge by further dividing dataset D into K groups by the arrival index $k \in [K]$ within batch. We denote the (singleton) dataset that contains the k^{th} draw in D_n by $D_{n,k}$ and the union of $D_{i,k}$ for $i = \{1, 2, \cdots, n\}$ as $\cup_{i=1}^{n} D_{i,k}$. The perturbation samples are treated similarly: we partition $Q_{n,e}$ to K groups, associating a group index (an auxiliary random variable) $\tilde{k}_{n,e,x} \sim \text{Unif}([K])$ to each perturbation example $(\tilde{s}_{n,e,x}, \tilde{a}_{n,e,x})$.

Specifically, for $n \in [N]$, we define a $S \cdot A \cdot K$-dimensional random variable $p_{n,e}$. The role of $p_{n,e}(s, a, k)$ is to count the occurrences (s, a) within $\cup_{i=1}^{n} D_{i,k}$ as well as the k-th subgroup of $Q_{n,e}$. Its formal definition is as follows:

$$p_{n,e}(s, a, k) := \sum_{i=1}^{n-1} I(s = s_{i,k}, a = \pi^{\exp}(s_{i,k})) + \sum_{x=1}^{X_{n,e}} I(s = \tilde{s}_{n,e,x}, a = \tilde{a}_{n,e,x}, k = \tilde{k}_{n,e,x}). \tag{16}$$

By recalling the definition $u_{n,e}$ in MFTPL-P, we rewrite $u_{n,e}$ as a function of $p_{n,e}$:

$$\begin{aligned} u_{n,e} =& \operatorname{Onehot}(\mathcal{O}(D_{1:n-1} \cup Q_{n,e})) \\ =& \operatorname{Onehot}(\operatorname*{argmin}_{h \in \mathcal{B}} \mathbb{E}_{(s,a) \sim D_{1:n-1} \cup Q_{n,e}} \left[I(h(s) \neq a) \right]) \\ =& \operatorname{Onehot}(\operatorname*{argmin}_{h \in \mathcal{B}} \sum_{(s,a) \in D_{1:n-1} \cup Q_{n,e}} \left[I(h(s) \neq a) \right]) \\ =& \operatorname{Onehot}(\operatorname*{argmin}_{h \in \mathcal{B}} \sum_{(s,a,k) \in \mathcal{S} \times \mathcal{A} \times [K]} p_{n,e}(s, a, k) \left[I(h(s) \neq a) \right]). \end{aligned}$$

Observe that $u_n^* = \mathbb{E}_{Q_{n,e}} \left[u_{n,e} \right]$ and $u_{n+1}^{**} = \mathbb{E}_{D_n} \mathbb{E}_{Q_{n+1,e}} \left[u_{n+1,e} \right]$ can also be viewed as the conditional distributions of $h_{n,e} \mid \mathcal{F}_{n-1}$ and $h_{n+1,e} \mid \mathcal{F}_{n-1}^{+}$, respectively. We define $\mathcal{P}_n(\cdot | \mathcal{F}_{n-1})$ as the conditional distribution of $p_{n,e} \mid \mathcal{F}_{n-1}$ and define $\mathcal{P}_{n+1}(\cdot | \mathcal{F}_{n-1}^{+})$ represent the conditional distribution of $p_{n+1,e} \mid \mathcal{F}_{n-1}^{+}$. By applying data-processing inequality (Beaudry and Renner, 2011), we obtain

$$\| u_n^* - u_{n+1}^{**} \|_1 = 2\text{TV}(u_n^*, u_{n+1}^{**}) \leq 2\text{TV}(\mathcal{P}_n(\cdot | \mathcal{F}_{n-1}), \mathcal{P}_{n+1}(\cdot | \mathcal{F}_{n-1}^{+})).$$

Note that $\mathcal{P}_n(\cdot | \mathcal{F}_{n-1})$ and $\mathcal{P}_{n+1}(\cdot | \mathcal{F}_{n-1}^{+})$ depend on the same historical dataset $D_{1:n-1}$, we further define

$$q_{n,e}(s, a, k) := \sum_{x=1}^{X} I(s = \tilde{s}_{n,e,x}, a = \tilde{a}_{n,e,x}, k = \tilde{k}_{n,e,x}), \tag{17}$$

$$r_{n,e}(s, a, k) := I(s = s_{n-1,k}, a = \pi^{\exp}(s_{n-1,k})) + \sum_{x=1}^{X} I(s = \tilde{s}_{n,e,x}, a = \tilde{a}_{n,e,x}, k = \tilde{k}_{n,e,x}), \tag{18}$$

It is not hard to see that following the definition in equation (16),

$$\begin{aligned} p_{n,e} =& q_{n,e} + \left(\sum_{i=1}^{n-1} I\left(s = s_{i,k}, a = \pi^{\exp}(s_{i,k}) \right) \right)_{(s,a,k) \in \mathcal{S} \times \mathcal{A} \times [K]}, \\ p_{n+1,e} =& r_{n+1,e} + \left(\sum_{i=1}^{n-1} I\left(s = s_{i,k}, a = \pi^{\exp}(s_{i,k}) \right) \right)_{(s,a,k) \in \mathcal{S} \times \mathcal{A} \times [K]} \end{aligned}$$

Notice that the distribution of $q_{n,e} \mid \mathcal{F}_{n-1}$ is independent of both n and e. Indeed, $q_{n,e}$ is a function of $Q_{n,e}$ and random variables $(\tilde{k}_{n,e,x})_{x=1}^{X_{n,e}}$. By the subsampling property of Poisson distribution, we can view each entry of $q_{n,e}$ as a independent Poisson random variable, following $q_{n,e}(s, a, k) \sim \text{Poi}(\tilde{\lambda}(s))$, where $\tilde{\lambda}(s) := \frac{\lambda d_0(s)}{AK}$. Therefore, $q_{n,e} \mid \mathcal{F}_{n-1}$ is drawn from a product of Poisson distributions:

$$\prod_{s \in \mathcal{S}} \prod_{a \in \mathcal{A}} \prod_{k \in [K]} \text{Poi}(q(s, a, k); \tilde{\lambda}(s)) =: \mathcal{Q}(q),$$

where $\text{Poi}(q, \lambda) = e^{-\lambda} \frac{\lambda^q}{q!} I(q \in \mathbb{N})$ denotes the probability mass function for Poisson distribution.

Therefore, in subsequent proofs, we denote the distribution of $q_{n,e} \mid \mathcal{F}_{n-1}$ by $\mathcal{Q}$ for simplicity. Meanwhile, since the conditional distribution of $r_{n+1,e} \mid \mathcal{F}_{n-1}^+$ is constant over all e , we denote the conditional distribution of $r_{n+1,e} \mid \mathcal{F}_{n-1}^+$ by $\mathcal{R}_{n+1,e}(\cdot|\mathcal{F}_{n-1}^+)$, and use the notation $\mathcal{R}_{n+1}$ for simplicity when it is clear from the context. Here, we apply the translation invariance property of TV distance and obtain

$$\mathrm{TV}(\mathcal{P}_n(\cdot|\mathcal{F}_{n-1}), \mathcal{P}_{n+1}(\cdot|\mathcal{F}_{n-1}^+)) = \mathrm{TV}(\mathcal{Q}(\cdot), \mathcal{R}_{n+1,e}(\cdot|\mathcal{F}_{n-1}^+)) = \mathrm{TV}(\mathcal{Q}, \mathcal{R}_{n+1}).$$

Next, we rewrite $\mathcal{R}_{n+1}$ by the tower property:

$$\begin{aligned}
\mathcal{R}_{n+1,e}(r|\mathcal{F}_{n-1}^+) =& \mathbb{P}(r_{n+1,e} = r \mid \mathcal{F}_{n-1}^+) \\
=& \mathbb{E}\left[\mathbb{P}(r_{n+1,e} = r \mid D_n, \mathcal{F}_{n-1}^+) \mid \mathcal{F}_{n-1}^+\right] \\
=& \mathbb{E}\left[\mathcal{R}_{n+1}(r|D_n, \mathcal{F}_{n-1}^+) \mid \mathcal{F}_{n-1}^+\right] = \mathbb{E}_{D_n}\left[\mathcal{R}_{n+1}(r \mid D_n, \mathcal{F}_{n-1}^+)\right].
\end{aligned}$$

Now, it suffices to bound $\mathrm{TV}(\mathcal{Q}, \mathbb{E}_{D_n}\left[\mathcal{R}_{n+1}(\cdot|D_n, \mathcal{F}_{n-1}^+)\right])$, which we bound in a way similar to bounding $\mathrm{TV}(P, Q)$ in Section 4.2.1 of (Haghtalab et al., 2022a). By this observation, we have

$$\begin{aligned}
\left\langle g_n, u_n^* - u_{n+1}^{**}\right\rangle \leq& 2\mathrm{TV}(\mathcal{Q}, \mathbb{E}_{D_n}\left[\mathcal{R}_{n+1}(\cdot|D_n, \mathcal{F}_{n-1}^+)\right]) \\
\leq& \sqrt{2\chi^2\left(\mathbb{E}_{D_n}\left[\mathcal{R}_{n+1}(\cdot|D_n, \mathcal{F}_{n-1}^+)\right], \mathcal{Q}\right)} \\
=& \sqrt{2\left(\mathbb{E}_{D_n, D_n'}\left[\mathbb{E}_{q\sim\mathcal{Q}}\left[\frac{\mathcal{R}_{n+1}(q|D_n, \mathcal{F}_{n-1}^+) \cdot \mathcal{R}_{n+1}(q|D_n', \mathcal{F}_{n-1}^+)}{\mathcal{Q}(q)^2}\right]\right] - 1\right)},
\end{aligned} \tag{19}$$

where we apply similar technique in Section 4.2.1 of Haghtalab et al. (2022a) by using χ^2 distance (Lemma E.1 of Haghtalab et al. (2022a)) and Ingster's method (Lemma E.2 of Haghtalab et al. (2022a)). Note that all examples in $D_n = \left\{(s_{n,k}, \pi^{\exp}(s_{n,k}))\right\}_{k=1}^K$, $D_n' = \left\{(s_{n,k}', \pi^{\exp}(s_{n,k}'))\right\}_{k=1}^K$ are i.i.d. drawn from $\mathcal{D}_{\pi_n}^{\exp}$.

Recall that $Q_{n,e} \overset{d}{=} Q_{n+1,e}$, meaning the difference between the distributions of $q_{n,e}$ and $r_{n+1,e}$ is induced by the K examples from D_n. Conditioned on $D_n = \left\{(s_{n,k}, \pi^{\exp}(s_{n,k}))\right\}_{k=1}^K$, we have

$$q_{n,e} \overset{d}{=} r_{n+1,e} - \left(I(s = s_{n,k}, a = \pi^{\exp}(s_{n,k}))\right)_{(s,a,k)\in\mathcal{S}\times\mathcal{A}\times[K]}.$$

This allows us to write the probability mass function of $\mathcal{R}_{n+1}(\cdot|D_n, \mathcal{F}_{n-1}^+)$ as:

$$\mathcal{R}_{n+1}(r|D_n, \mathcal{F}_{n-1}^+) = \prod_{s\in\mathcal{S}}\prod_{a\in\mathcal{A}}\prod_{k\in[K]} \mathrm{Poi}\left(r(s,a,k) - I\left(s = s_{n,k}, a = \pi^{\exp}(s_{n,k})\right); \tilde{\lambda}(s)\right).$$

Let D_n, D_n' be the pair of datasets of size K in equation (19), some algebraic calculations yield that

$$\frac{\mathcal{R}_{n+1}(q|D_n, \mathcal{F}_{n-1}^+) \cdot \mathcal{R}_{n+1}(q|D_n', \mathcal{F}_{n-1}^+)}{\mathcal{Q}(q)^2} = \prod_{k\in[K]} \frac{q\left(s_{n,k}, \pi^{\exp}(s_{n,k}), k\right)}{\tilde{\lambda}(s_{n,k})} \cdot \frac{q\left(s_{n,k}', \pi^{\exp}(s_{n,k}'), k'\right)}{\tilde{\lambda}(s_{n,k}')}.$$

By taking expectation over $q \sim \mathcal{Q}$ we have

$$\mathbb{E}_{q\sim\mathcal{Q}}\left[\frac{\mathcal{R}_{n+1}(q|D_n, \mathcal{F}_{n-1}^+) \cdot \mathcal{R}_{n+1}(q|D_n', \mathcal{F}_{n-1}^+)}{\mathcal{Q}(q)^2}\right] = \prod_{k\in[K]}\left(1 + \frac{I(s_{n,k} = s_{n,k}')}{\tilde{\lambda}(s_{n,k})}\right).$$

Furthermore, we take conditional expectation with respect to D_n, D'_n and obtain

$$\begin{aligned}\mathbb{E}_{D_n,D'_n}\left[\mathbb{E}_{q\sim\mathcal{Q}}\left[\frac{\mathcal{R}_{n+1}(q|D_n,\mathcal{F}^+_{n-1})\cdot\mathcal{R}_{n+1}(q|D'_n,\mathcal{F}^+_{n-1})}{\mathcal{Q}(q)^2}\right]\right] &= \prod_{k\in[K]}\left(1+\frac{AK}{\lambda}\sum_{s\in\mathcal{S}}\frac{d_{\pi_n}(s)^2}{d_0(s)}\right)\\ &\le \left(1+\frac{AK}{\lambda}\sum_{s\in\mathcal{S}}\frac{d_{\pi_n}(s)}{\sigma}\right)^K = \left(1+\frac{AK}{\lambda\sigma}\right)^K,\end{aligned} \tag{20}$$

where in the above derivation, we recall that $\tilde{\lambda}(s) := \frac{\lambda d_0(s)}{AK}$, $\mathbb{E}_{D_n,D'_n}\left[\frac{I(s_{n,k}=s'_{n,k})}{d_0(s_{n,k})}\right] = \sum_{s\in\mathcal{S}}\frac{d_{\pi_n}(s)^2}{d_0(s)}$, and $\frac{d_\pi(s)}{d_0(s)} \le \frac{1}{\sigma}$.

Finally, we conclude the proof by plugging equation (20) into equation (19) and setting $\lambda \ge \frac{2AK^2}{\sigma}$:

$$\langle g_n, u^*_n - u^{**}_{n+1}\rangle \le \sqrt{2\left(\left(1+\frac{AK}{\lambda\sigma}\right)^K - 1\right)} \le 2K\sqrt{\frac{A}{\lambda\sigma}},$$

which is due to $\forall x \in [0,\frac{1}{2}]$, $1+x \le e^x \le 1+2x$, meaning when $\frac{AK}{\lambda\sigma} \le \frac{AK^2}{\lambda\sigma} \le \frac{1}{2}$,

$$\left(1+\frac{AK}{\lambda\sigma}\right)^K \le \left(\exp\left(\frac{AK}{\lambda\sigma}\right)\right)^K = \exp\left(\frac{AK^2}{\lambda\sigma}\right) \le 1+\frac{2AK^2}{\lambda\sigma}.$$

□

To simplify the expression in the following proofs, we introduce shorthand $z_{n,k} := (s_{n,k}, \pi^{\exp}(s_{n,k}))$ and $\tilde{z}_{n,e,x} := (\tilde{s}_{n,e,x}, \tilde{a}_{n,e,x}))$. By definitions in MFTPL-P, $z_{n,k} \sim \mathcal{D}^{\exp}_{\pi_n}$, $\tilde{z}_{n,e,x} \sim \mathcal{D}_0$. We provide a generalized coupling lemma similar to (Haghtalab et al., 2022a;b), showing that multiple draws from the covering distribution can be seen as containing a batch of examples from a covering distribution with high probability.

Lemma 19 (Generalized coupling). *Let $G \in \mathbb{N}$ and $\tilde{z}_1, \cdots, \tilde{z}_G \sim \mathcal{D}_0$. For all π that satisfies $\forall s \in \mathcal{S}$, $\frac{d_\pi(s)}{d_0(s)} \le \frac{1}{\sigma}$. By some external randomness R, there exists an index $\mathcal{I} = \mathcal{I}(\tilde{z}_1, \cdots, \tilde{z}_G, R) \in [G]$ and a success event $U = U(\tilde{z}_1, \cdots, \tilde{z}_G, R)$ such that $\Pr[U^c] \le (1-\sigma/A)^G$, and*

$$\left(\tilde{z}_{\mathcal{I}} \mid U, \tilde{z}_{\setminus\mathcal{I}}\right) \sim \mathcal{D}^{\exp}_\pi,$$

where $\tilde{z}_{\setminus\mathcal{I}}$ denotes $\{\tilde{z}_1, \cdots, \tilde{z}_G\} \setminus \{\tilde{z}_{\mathcal{I}}\}$.

Proof. For all π that satisfies $\forall s \in \mathcal{S}$, $\frac{d_\pi(s)}{d_0(s)} \le \frac{1}{\sigma}$ and its corresponding $\mathcal{D}^{\exp}_\pi$, we have $z = \left(s, \pi^{\exp}(s)\right) \sim \mathcal{D}^{\exp}_\pi$, following $s \sim d_\pi$, $a = \pi^{\exp}(s)$. Since $\tilde{z} = (\tilde{s}, \tilde{a}) \sim \mathcal{D}_0$ follows $\tilde{s} \sim d_0$, $\tilde{a} \sim \text{Unif}(\mathcal{A})$. By their definition It is straight forward to see

$$\frac{\mathcal{D}^{\exp}_\pi(z)}{\mathcal{D}_0(z)} = \frac{d_\pi(s)}{d_0(s)} \cdot \frac{1}{\text{Unif}(a;\mathcal{A})} \le \frac{A}{\sigma}.$$

We conclude the proof by letting $X = \mathcal{D}^{\exp}_\pi$ and $Y = \mathcal{D}_0$ in Lemma 4.7 of Haghtalab et al. (2022a). □

Lemma 20. $\forall n \in \{1, \cdots, N\}$, MFTPL-P *with* $\lambda \ge \frac{4AK\ln(N)}{\sigma}$ *achieves*

$$\mathbb{E}_{D_n}\left[\langle g^*_n - g_n, u^*_{n+1}\rangle\right] \le \sqrt{\frac{2A\ln(N)\ln(BN^2)}{\lambda\sigma}} + \frac{\lambda\sigma}{4AN^2\ln(N)} + Ke^{-\frac{\lambda}{8K}}.$$

Remark 21. *When applying this lemma downstream, we will treat the last two terms as lower order terms: First, our final setting of λ will be such that $\lambda = O(N)$, in which case $\frac{\lambda\sigma}{4AN^2\ln(N)} \le O(\frac{1}{N})$; Second, we will focus on the regime that $\lambda \ge \frac{8AK\ln(N)}{\sigma} \ge 8K\ln N$, in which case $Ke^{-\frac{\lambda}{8K}} = O(\frac{K}{N})$.*

Proof. To begin with, we first rewrite

$$\mathbb{E}_{D_n}\left[\left\langle g_n^* - g_n, u_{n+1}^*\right\rangle\right] = \mathbb{E}_{D_n}\mathbb{E}_{Q_{n+1,e}}\left[\left\langle g_n^* - g_n, u_{n+1,e}\right\rangle\right].$$

Following the same method in Lemma 18, for the e^{th} perturbation set $Q_{n+1,e}$ at round n, we divide it to K subsets $(Q_{n+1,e,k})_{k=1}^K$ by randomly assigning $\tilde{k}_{n+1,e,x} \sim \text{Unif}([K])$ to each example $(\tilde{s}_{n+1,e,x}, \tilde{a}_{n+1,e,x})$ in $Q_{n+1,e}$ for $x \in X_{n+1,e}$. Note that the divisions in this proof is only for analytical use.

By the subsampling property of Poisson distribution, we have the size of each $Q_{n+1,e,k}$ denoted by $X_{n+1,e,k}$ follows $\text{Poi}(\lambda/K)$. For notational simplicity, we use X_k to denote $X_{n+1,e,k}$, $Q_k = (\tilde{s}_x, \tilde{a}_x)_{x=1}^{X_k} = (\tilde{z}_x)_{x=1}^{X_k}$ to denote $Q_{n+1,e,k}$, and $z_k = (s_k, \pi^{\exp}(s_k))$ to denote $z_{n+1,k}$ in the following proof.

By our assumption that $\lambda \geq \frac{4AK\ln(N)}{\sigma}$, without loss of generality, for now we assume $\frac{\lambda}{2K}$ to be integral multiple of $G := \lceil \frac{2A\ln(N)}{\sigma} \rceil$, meaning $\frac{\lambda}{2K} = MG$ for some $M \in \mathbb{N}$. By defining event $\tilde{U}_k := \left\{X_k \geq \frac{\lambda}{2K}\right\}$ and $\tilde{U} := \cap_{k\in[K]}\tilde{U}_k$, we have

$$\Pr(\tilde{U_k^c}) = \Pr(X_k < \frac{\lambda}{2K}) \leq \exp(-\frac{\lambda}{8K}),\ \Pr(\tilde{U^c}) \leq \sum_{k=1}^K \Pr(\tilde{U_k^c}) \leq Ke^{-\frac{\lambda}{8K}},$$

where we use the fact that for $X \sim \text{Poi}(\lambda' = \lambda/K)$, $\Pr(X < \lambda'/2) \leq \exp(-\lambda'/8)$, and apply union bound for $\tilde{U}^c$.

At round n, conditioned on the favorable event $\tilde{U}$ happening, we further divide Q_k into M groups denoted by $Q_{k,m}$ for $m \in [M]$, where each group has size greater or equal to G.

Conditioned on $\tilde{U}$, we apply Lemma 19 to each $Q_{k,m}$ with distribution $\mathcal{D}_{\pi_n}^{\exp}$ (induced by π_n), obtaining M independent events $U_{k,m}$ for $m \in [M]$, where

$$\Pr(U_{k,m}^c|\tilde{U}) \leq (1-\sigma/A)^G = (1-\frac{\sigma}{A})^{\frac{A}{\sigma}\cdot 2\ln(N)} \leq e^{-2\ln(N)} = N^{-2}.$$

Conditioned on $U_{k,m}$, there exist an element $\zeta_{k,m} \in Q_{k,m}$ such that

$$(\zeta_{k,m}|U_{k,m}, Q_{k,m}\backslash\left\{\zeta_{k,m}\right\}) \sim \mathcal{D}_{\pi_n}^{\exp}.$$

Define event $U := \cap_{k\in[K],m\in[M]}U_{k,m}$ to be the intersection of those independent events (at round n), where by union bound and the definition of M we have

$$\Pr(U^c|\tilde{U}) \leq \frac{KM}{N^2} \leq \frac{K}{N^2}\cdot\frac{\lambda}{2K\lceil\frac{2A\ln(N)}{\sigma}\rceil} \leq \frac{\lambda\sigma}{4AN^2\ln(N)}.$$

Now we introduce shorthand $\xi_{k,m} := Q_{k,m}\backslash\left\{\zeta_{k,m}\right\}$, $\xi := \cup_{k\in[K],m\in[M]}\xi_{k,m}$ and write

$$(z_{n,1},\cdots,z_{n,K},\zeta_{1,1},\cdots,\zeta_{1,M},\zeta_{2,1},\cdots,\zeta_{K,M}|\xi,U,\tilde{U},\mathcal{F}_{n-1}^+) \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\pi_n}^{\exp},$$

which is by the independence between each group as well as the samples from $\mathcal{D}_{\pi_n}^{\exp}$.

Now, we can split the generalization error into three terms by the law of total expectation:

$$\begin{aligned}\mathbb{E}_{D_n,Q_{n+1,e}}\left[\left\langle g_n^* - g_n, u_{n+1,e}\right\rangle\right] =& \Pr(U\cap\tilde{U})\cdot\mathbb{E}_{D_n,Q_{n+1,e}}\left[\left\langle g_n^* - g_n, u_{n+1,e}\right\rangle|U,\tilde{U}\right] + \Pr(U^c\cup\tilde{U}^c)\\ \leq& \mathbb{E}_{D_n,Q_{n+1,e}}\left[\left\langle g_n^* - g_n, u_{n+1,e}\right\rangle|U,\tilde{U}\right] + \Pr(U^c\cap\tilde{U}) + \Pr(\tilde{U}^c)\\ \leq& \mathbb{E}_{D_n,Q_{n+1,e}}\left[\left\langle g_n^* - g_n, u_{n+1,e}\right\rangle|U,\tilde{U}\right] + \frac{\lambda\sigma}{4AN^2\ln(N)} + Ke^{-\frac{\lambda}{8K}},\end{aligned} \tag{21}$$

where we apply the fact that $\left\langle g_n^* - g_n, u_{n+1}^*\right\rangle \leq \|g_n^* - g_n\|_\infty \cdot \|u_{n+1}^*\|_1 \leq 1$, and bring in the bounds for $\Pr(\tilde{U}^c)$ and $\Pr(U^c|\tilde{U})$ shown above.

For the remaining term $\mathbb{E}_{D_n,Q_{n+1,e}}\left[\left\langle g_n^* - g_n, u_{n+1,e}\right\rangle|U,\tilde{U}\right]$, we abbreviate it as $\mathbb{E}_{D_n,Q_{n+1,e}|U,\tilde{U}}\left[\left\langle g_n^* - g_n, u_{n+1,e}\right\rangle\right]$ and split it by the linearity of expectation

$$\mathbb{E}_{D_n,Q_{n+1,e}|U,\tilde{U}}\left[\left\langle g_n^* - g_n, u_{n+1,e}\right\rangle\right] = \underbrace{\mathbb{E}_{D_n,Q_{n+1,e}|U,\tilde{U}}\left[\left\langle g_n^*, u_{n+1,e}\right\rangle\right]}_{\mathbf{I}} - \underbrace{\mathbb{E}_{D_n,Q_{n+1,e}|U,\tilde{U}}\left[\left\langle g_n, u_{n+1,e}\right\rangle\right]}_{\mathbf{II}}.$$

We first focus on term II. For now, we abbreviate $Q_{n+1,e}$ as Q_{n+1} when it is clear from the context. By introducing shorthand of $h_{n+1} = \mathcal{O}(\cup_{i=1}^n D_i \cup Q_{n+1})$ corresponding to the only policy in the support of $u_{n+1,e}$, and denote $\ell(h_{n+1},(s,a)) := I(h_{n+1}(s) \neq a)$, we rewrite **II** as

$$\begin{aligned}\mathbb{E}_{D_n,Q_{n+1}|U,\tilde{U}}\left[\left\langle g_n, u_{n+1,e}\right\rangle\right] =& \mathbb{E}_{D_n,Q_{n+1}|U,\tilde{U}}\left[\frac{1}{K}\sum_{k=1}^K I(h_{n+1}(z_{n,k}) \neq \pi^{\exp}(z_{n,k}))\right] \\ =& \frac{1}{K}\sum_{k=1}^K \mathbb{E}_{D_n,Q_{n+1}|U,\tilde{U}}\left[\ell(h_{n+1},z_{n,k})\right].\end{aligned}$$

Here we further denote $\zeta = \{\zeta_{k,m}\}_{k\in[K],m\in[M]}$. With this notation, $Q_{n+1} = \zeta \cup \xi$. The following holds for all $m \in [M]$:

$$\begin{aligned}\mathbf{II} =& \frac{1}{K}\sum_{k=1}^K \mathbb{E}_{\xi|U,\tilde{U},\mathcal{F}_{n-1}^+}\mathbb{E}\left[\ell(h_{n+1},z_{n,k})|\xi,U,\tilde{U},\mathcal{F}_{n-1}^+\right] \\ =& \frac{1}{K}\sum_{k=1}^K \mathbb{E}_{\xi|U,\tilde{U},\mathcal{F}_{n-1}^+}\mathbb{E}\left[\ell(\mathcal{O}(\cup_{i=1}^{n-1}D_i \cup D_n \cup Q_{n+1})),z_{n,k})|\xi,U,\tilde{U},\mathcal{F}_{n-1}^+\right] \\ =& \frac{1}{K}\sum_{k=1}^K \mathbb{E}_{\xi|U,\tilde{U},\mathcal{F}_{n-1}^+}\mathbb{E}\left[\ell(\mathcal{O}(\cup_{i=1}^{n-1}D_i \cup (D_n\backslash\{z_{n,k}\}) \cup (Q_{n+1}\backslash\{\zeta_{k,m}\}) \cup \{z_{n,k}\} \cup \{\zeta_{k,m}\})),z_{n,k})|\xi,U,\tilde{U},\mathcal{F}_{n-1}^+\right] \\ =& \frac{1}{K}\sum_{k=1}^K \mathbb{E}_{\xi|U,\tilde{U},\mathcal{F}_{n-1}^+}\mathbb{E}\left[\ell(\mathcal{O}(\cup_{i=1}^{n-1}D_i \cup (D_n\backslash\{z_{n,k}\}) \cup (Q_{n+1} \setminus \{\zeta_{k,m}\}) \cup \{z_{n,k}\} \cup \{\zeta_{k,m}\})),\zeta_{k,m})|\xi,U,\tilde{U},\mathcal{F}_{n-1}^+\right] \\ =& \frac{1}{K}\sum_{k=1}^K \mathbb{E}_{D_n,Q_{n+1}|U,\tilde{U}}\left[\ell(h_{n+1},\zeta_{k,m})\right],\end{aligned} \tag{22}$$

where in the fourth equality we apply the independence between samples and exchangeability between $z_{n,k}$ and $\zeta_{k,m}$ conditioned on U, $\tilde{U}$, $\mathcal{F}_{n-1}^+$ and ξ. By slightly abusing notations and denoting $z_{n,k}$ as $\zeta_{k,0}$, this implies:

$$\mathbf{II} = \frac{1}{M+1}\sum_{m=0}^M\left(\frac{1}{K}\sum_{k=1}^K \mathbb{E}_{D_n,Q_{n+1}|U,\tilde{U}}\left[\ell(h_{n+1},\zeta_{k,m})\right]\right) = \mathbb{E}_{D_n,Q_{n+1}|U,\tilde{U}}\left[\frac{1}{K(M+1)}\sum_{k=1}^K\sum_{m=0}^M \ell(h_{n+1},\zeta_{k,m})\right].$$

Meanwhile, we denote $z \sim \mathcal{D}_{\pi_n}^{\exp}$ and rewrite **I** with the same h_{n+1} notation:

$$\mathbf{I} = \mathbb{E}_{D_n,Q_{n+1}|U,\tilde{U}}\left[\mathbb{E}_{z\sim\mathcal{D}_{\pi_n}^{\exp}}\left[\ell(h_{n+1},z)\right]\right].$$

Combining what we have, we finally conclude that $\mathbf{I}-\mathbf{II}$ is bounded by

$$\begin{aligned}\mathbf{I}-\mathbf{II} =&\mathbb{E}_{D_n,Q_{n+1}|U,\tilde{U}}\left[\mathbb{E}_{z\sim\mathcal{D}^{\exp}_{\pi_n}}\left[\ell(h_{n+1},z)\right]-\frac{1}{K(M+1)}\sum_{k=1}^{K}\sum_{m=0}^{M}\ell(h_{n+1},\zeta_{k,m})\right]\\ =&\mathbb{E}_{\xi|U,\tilde{U},\mathcal{F}^{+}_{n-1}}\mathbb{E}\left[\mathbb{E}_{z\sim\mathcal{D}^{\exp}_{\pi_n}}\left[\ell(h_{n+1},z)\right]-\frac{1}{K(M+1)}\sum_{k=1}^{K}\sum_{m=0}^{M}\ell(h_{n+1},\zeta_{k,m})\mid\xi,U,\tilde{U},\mathcal{F}^{+}_{n-1}\right]\\ \leq&\mathbb{E}_{\xi|U,\tilde{U},\mathcal{F}^{+}_{n-1}}\mathbb{E}\left[\sup_{h\in\mathcal{B}}\left(\mathbb{E}_{z\sim\mathcal{D}^{\exp}_{\pi_n}}\left[\ell(h,z)\right]-\frac{1}{K(M+1)}\sum_{k=1}^{K}\sum_{m=0}^{M}\ell(h,\zeta_{k,m})\right)\mid\xi,U,\tilde{U},\mathcal{F}^{+}_{n-1}\right]\\ \leq&\sqrt{\frac{\ln(B)}{2K(M+1)}}\leq\sqrt{\frac{2A\ln(N)\ln(B)}{\lambda\sigma}},\end{aligned}\tag{23}$$

where in the second equality, we use the law of iterated expectations; in the first inequality, we upper bound the random variable of interest by the supremum over the policy class, since $h_{n+1}\in\mathcal{B}$. The second inequality is from Massart's Lemma (Lemma 16).

We finish the proof by bringing equation (23) into equation (21). □

Theorem 22. *For any $\delta\in(0,1]$,* MFTPL-P *with any $\lambda\geq\max\left\{\frac{2AK^2}{\sigma},\frac{8AK\ln(KN)}{\sigma}\right\}$ and $E=NA\ln(NS)$ outputs $\{\pi_n\}_{n=1}^{N}$ that satisfies that with probability at least $1-\delta$,*

$$\mathrm{Reg}(N)\leq\tilde{O}\left(\sqrt{\frac{\lambda\ln B}{K^2}}+N\sqrt{\frac{AK^2}{\lambda\sigma}}+N\sqrt{\frac{A\ln B}{\lambda\sigma}}+\frac{\lambda\sigma}{AN}+\sqrt{N\ln\frac{1}{\delta}}\right).\tag{24}$$

Specifically, if $N\geq\tilde{O}\left(\sqrt{\frac{A}{\sigma}}\sqrt{\min(\ln B,K^2)}\vee\frac{K^2}{A}\vee\frac{K^4}{A\ln B}\right)$, setting $\lambda=\Theta\left(NK\sqrt{\frac{A}{\sigma}}+NK^2\sqrt{\frac{A}{\sigma\ln B}}\right)$ gives

$$\mathrm{Reg}(N)\leq\tilde{O}\left(\sqrt{N}\left(\frac{A(\ln B)^2}{\sigma K^2}\right)^{\frac{1}{4}}+\sqrt{N}\left(\frac{A\ln B}{\sigma}\right)^{\frac{1}{4}}+\sqrt{N\ln(1/\delta)}\right).\tag{25}$$

Proof. Fix $\delta\in(0,1)$. By combining Lemmas 12, 15, and 17, when $\lambda\geq\max\left\{\frac{2AK^2}{\sigma},\frac{8AK\ln(KN)}{\sigma}\right\}$, we have that , with probability at least $1-\delta/2$,

$$\begin{aligned}&\sum_{n=1}^{N}\langle g_n,u_n^*\rangle-\min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n,u\rangle\\ \leq&\mathbb{E}_{Q_1}\left[\max_{u\in\Delta(\mathcal{B})}\langle-\tilde{g}_1,u\rangle\right]+\sum_{n=1}^{N}\langle g_n,u_n^*-u_{n+1}^*\rangle\\ \leq&\sqrt{\frac{\lambda\ln(B)}{2K^2}}+N\sqrt{\frac{AK^2}{\lambda\sigma}}+N\sqrt{\frac{2A\ln(N)\ln(BN^2)}{\lambda\sigma}}+\frac{\lambda\sigma}{4AN\ln(N)}+4\sqrt{2N\ln(4/\delta)}.\end{aligned}\tag{26}$$

We now apply Proposition 11, which, by the choice of E, gives that with probability $1-\delta/2$,

$$\sum_{n=1}^{N}\langle g_n,u_n-u_n^*\rangle\leq N\sqrt{\frac{2A\left(\ln(NS)+\ln(\frac{12}{\delta})\right)}{E}}\leq O\left(\sqrt{N\ln\frac{1}{\delta}}\right).$$

Therefore, combining the two inequalities and using the union bound, with probability at least $1-\delta$, Eq. (24) holds.

Now, for proving Eq. (25), we first note that by the assumption that $N \geq \tilde{O}\left(\sqrt{\frac{A}{\sigma}}\sqrt{\min(\ln B, K^2)}\right)$, the choice of $\lambda = \Theta\left(NK\sqrt{\frac{A}{\sigma}} + NK^2\sqrt{\frac{A}{\sigma \ln B}}\right)$ satisfies that $\lambda \geq \max\left\{\frac{2AK^2}{\sigma}, \frac{8AK\ln(KN)}{\sigma}\right\}$; so Eq. (24) applies. It can be checked by algebra that the first three terms in Eq. (24) evaluates to $\tilde{O}\left(\sqrt{N}\left(\frac{A(\ln B)^2}{\sigma K^2}\right)^{\frac{1}{4}} + \sqrt{N}\left(\frac{A\ln B}{\sigma}\right)^{\frac{1}{4}}\right)$.

Furthermore, our choice of λ and the assumption that $N \geq \tilde{O}\left(\frac{K^2}{A} \vee \frac{K^4}{A\ln B}\right)$ implies that

$$\frac{\lambda\sigma}{AN} \leq O\left(\frac{K}{\sqrt{A}} + \frac{K^2}{\sqrt{A\ln B}}\right) \leq O\left(\sqrt{N}\right);$$

therefore, the last two terms in Eq. (24) is at most $O(\sqrt{N\ln\frac{1}{\delta}})$. This concludes the proof of Eq. (25).

□

Theorem 22 immediately implies the following corollary:

Corollary 23. *For $\alpha > 0$ small enough,* MFTPL-P *with*

$$N \geq \tilde{O}\left(\frac{1}{\alpha^2}\sqrt{\frac{A\ln B}{\sigma}}\right), \text{ and } NK \geq \tilde{O}\left(\frac{1}{\alpha^2}\sqrt{\frac{A(\ln B)^2}{\sigma}}\right), \tag{27}$$

is such that, with the choices of parameters $\lambda = \Theta\left(NK\sqrt{\frac{A}{\sigma}} + NK^2\sqrt{\frac{A}{\sigma\ln B}}\right)$ and $E = NA\ln(NS)$, achieves $\frac{\mathrm{Reg}(N)}{N} \leq \alpha$ with probability $1-\delta$; its number of calls to the offline oracle is $NE = \tilde{O}(N^2 A)$.

Corollary 23 implies Corollary 4 in the main text, as we show below:

Proof of Corollary 4. By the choices of N and K, Eq. (27) is satisfied with $\alpha = \frac{\epsilon}{\mu H}$. Therefore, with the choices of parameters given by Corollary 23, $\frac{\mathrm{Reg}(N)}{N} \leq \alpha \implies \frac{\mu H \mathrm{Reg}(N)}{N} \leq \epsilon$.

The total number of demonstrations requested is $NK = \frac{\mu^2 H^2}{\epsilon^2}\sqrt{\frac{A(\ln B)^2}{\sigma}}$, and the total number of calls to the offline oracle is $O(N^2 A) = \tilde{O}\left(\frac{\mu^4 H^4 A^2 (\ln B)^2}{\epsilon^2 \sigma}\right)$. □

B.3. Deferred proofs from Section B.2

The proposition below is used in the analysis of stage 1; its proof is straightforward and largely follows the proof of Proposition 6 in (Li and Zhang, 2022).

Proposition 24. *For any $\delta \in (0,1]$, the sequence of $\{u_n\}$, $\{g_n\}$, $\{g_n^*\}$ induced by* MFTPL-P*, MDP $\mathcal{M}$ and expert $\pi^{\exp}$, satisfies that with probability at least $1-\delta/3$, it holds simultaneously that:*

$$\sum_{n=1}^{N}\langle g_n^* - g_n, u_n\rangle \leq \sqrt{\frac{2N\ln(\frac{12}{\delta})}{K}},$$

$$\min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n, u\rangle - \min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n^*, u\rangle \leq \sqrt{2N\frac{\ln(B)+\ln(\frac{12}{\delta})}{K}}.$$

Proof. It suffices to show for any $\delta \in (0,1]$, (1). $\sum_{n=1}^{N}\langle g_n^* - g_n, u_n\rangle \leq \sqrt{\frac{2N\ln(\frac{12}{\delta})}{K}}$ with probability at least $1-\delta/6$, (2). $\min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n, u\rangle - \min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n^*, u\rangle \leq \sqrt{2N\frac{\ln(B)+\ln(\frac{12}{\delta})}{K}}$ with probability at least $1-\delta/6$.

For (1), we define $Y_{n,k} = F_n(\pi_n) - \mathbb{E}_{a\sim\pi_n(\cdot|s_{n,k})}\left[I(a \neq \pi^{\exp}(s_{n,k}))\right]$, which satisfies

$$\begin{aligned}\langle g_n^* - g_n, u_n\rangle =& F_n(\pi_n) - \mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\mathbb{E}_{a\sim\pi_n(\cdot|s)}\left[I(a \neq \pi^{\exp}(s))\right] \\ =&\frac{1}{K}\sum_{k=1}^{K}\left(F_n(\pi_n) - \mathbb{E}_{a\sim\pi_n(\cdot|s_{n,k})}\left[I(a\neq\pi^{\exp}(s_{n,k}))\right]\right) = \frac{1}{K}\sum_{k=1}^{K}Y_{n,k},\end{aligned}$$

where we apply $\langle g_n^*, u_n\rangle = F_n(\pi_n)$ and $D_n = (s_{n,k}, \pi^{\exp}(s_{n,k}))_{k=1}^{K}$.

Now, it suffices to bound $\sum_{n=1}^{N}\sum_{k=1}^{K} Y_{n,k}$, which can be verified to be a martingale difference sequence. By the definition of $F_n(\pi) := \mathbb{E}_{s\sim d_{\pi_n}}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[I(a\neq\pi^{\exp}(s))\right]$, it can be shown that $\mathbb{E}\left[Y_{n,k}|Y_{1,1}, Y_{1,2},\cdots, Y_{1,K}, Y_{2,1},\cdots,Y_{n,k-1}\right] = 0$, while $|Y_{n,k}| \leq 1$. By applying Azuma-Hoeffding's inequality, with probability at least $1-\delta/6$,

$$\left|\sum_{n=1}^{N}\langle g_n^* - g_n, u_n\rangle\right| = \frac{1}{K}\left|\sum_{n=1}^{N}\sum_{k=1}^{K}Y_{n,k}\right| \leq \sqrt{\frac{2N\ln(\frac{12}{\delta})}{K}}.$$

For (2), we define $\hat{Y}_{n,k}(h) = F_n(h) - I\left(h(s_{n,k}) \neq \pi^{\exp}(s_{n,k})\right)$, where $h \in \mathcal{B}$, $s_{n,k} \sim d_{\pi_n}$. Following similar proof as (1), it can be verified that $\mathbb{E}\left[\hat{Y}_{n,k}(h)|\hat{Y}_{1,1}(h),\cdots,\hat{Y}_{n,k-1}(h)\right] = 0$ and $|\hat{Y}_{n,k}(h)| \leq 1$. Again we apply Azuma-Hoeffding's inequality and show that given any $\delta \in (0,1]$ and $h\in\mathcal{B}$, with probability at least $1-\frac{\delta}{6B}$,

$$\left|\sum_{n=1}^{N}\left(F_n(h) - \frac{1}{K}\sum_{k=1}^{K}I\left(h(s_{n,k})\neq\pi^{\exp}(s_{n,k})\right)\right)\right| = \left|\frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Y}_{n,k}(h)\right| \leq \sqrt{2N\frac{\ln(B)+\ln(\frac{12}{\delta})}{K}}.$$

By applying union bound over all policies in $\mathcal{B}$, we have for all $h\in\mathcal{B}$, given any $\delta\in(0,1]$, with probability at least $1-\frac{\delta}{6}$, it satisfies that

$$\frac{1}{K}\sum_{k=1}^{K}I\left(h(s_{n,k})\neq\pi^{\exp}(s_{n,k})\right) - \sum_{n=1}^{N}F_n(h) \leq \sqrt{2N\frac{\ln(B)+\ln(\frac{12}{\delta})}{K}}.$$

Since $\min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n^*, u\rangle = \min_{h\in\mathcal{B}}\sum_{n=1}^{N}F_n(h)$, while $\min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n, u\rangle = \min_{h\in\mathcal{B}}\frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{K}I\left(h(s_{n,k})\neq\pi^{\exp}(s_{n,k})\right)$, we denote $h^* = \operatorname{argmin}_{h\in\mathcal{B}}\sum_{n=1}^{N}F_n(h)$ and conclude the proof for (2) by

$$\begin{aligned}\min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n, u\rangle - \min_{u\in\Delta(\mathcal{B})}\sum_{n=1}^{N}\langle g_n^*, u\rangle =& \min_{h\in\mathcal{B}}\frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{K}I\left(h(s_{n,k})\neq\pi^{\exp}(s_{n,k})\right) - \sum_{n=1}^{N}F_n(h^*) \\ =& \min_{h\in\mathcal{B}}\frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{K}I\left(h(s_{n,k})\neq\pi^{\exp}(s_{n,k})\right) \\ &-\frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{K}I\left(h^*(s_{n,k})\neq\pi^{\exp}(s_{n,k})\right) \\ &+\frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{K}I\left(h^*(s_{n,k})\neq\pi^{\exp}(s_{n,k})\right) - \sum_{n=1}^{N}F_n(h^*) \\ \leq&\sqrt{2N\frac{\ln(B)+\ln(\frac{12}{\delta})}{K}}.\end{aligned}$$

Finally, by applying union bound on (1) and (2) we conclude the proof. □

The proposition below is used in the analysis of stage 1; its proof is straightforward and largely follows from Lemma 7 and 8 of (Li and Zhang, 2022).

Proposition 25. *For any* $\delta \in (0,1]$*, the sequence of* $\{u_n\}$*,* $\{g_n\}$*,* $\{u_n^*\}$ *induced by* MFTPL-P*, MDP* $\mathcal{M}$ *and expert* $\pi^{\exp}$*, satisfies that with probability at least* $1-\delta/6$*,*

$$\sum_{n=1}^{N} \langle g_n, u_n - u_n^* \rangle \le N\sqrt{\frac{2A\left(\ln(NS)+\ln(\frac{12}{\delta})\right)}{E}}.$$

Proof. To begin with, we first denote $\pi_n^* := \pi_{u_n^*}$ and $\pi_n^*(\cdot|s)$ the action distribution of π_n^* on state s. Given the expert annotation $\pi^{\exp}(s)$ on state s, we denote the A dimensional classification loss vector by $\vec{c}(\pi^{\exp}(s))$, whose entries are all 1 except that it takes 0 in the $\pi^{\exp}(s)$-th coordinate. With the newly introduced notions, we rewrite and bound $\langle g_n, u_n - u_n^* \rangle$ as follows:

$$\begin{aligned}
\langle g_n, u_n - u_n^* \rangle =& \sum_{h\in\mathcal{B}} u_n[h] \left(\mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\left[I(h(s) \neq \pi^{\exp}(s))\right]\right)_{h\in\mathcal{B}} \\
&- \sum_{h\in\mathcal{B}} u_n^*[h] \left(\mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\left[I(h(s) \neq \pi^{\exp}(s))\right]\right)_{h\in\mathcal{B}} \\
=&\mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\mathbb{E}_{a\sim\pi_n(\cdot|s)}\left[I(a\neq\pi^{\exp}(s))\right] - \mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\mathbb{E}_{a\sim\pi_n^*(\cdot|s)}\left[I(a\neq\pi^{\exp}(s))\right] \\
=&\mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\left[\langle \pi_n(\cdot|s), \vec{c}(\pi^{\exp}(s))\rangle\right] - \mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\left[\langle \pi_n^*(\cdot|s), \vec{c}(\pi^{\exp}(s))\rangle\right] \\
=&\mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\left[\langle \pi_n(\cdot|s) - \pi_n^*(\cdot|s), \vec{c}(\pi^{\exp}(s))\rangle\right] \\
\le&\mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\left[\|\pi_n(\cdot|s) - \pi_n^*(\cdot|s)\|_1 \|\vec{c}(\pi^{\exp}(s))\|_\infty\right] \\
=&\mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\left[\|\pi_n(\cdot|s) - \pi_n^*(\cdot|s)\|_1\right].
\end{aligned} \tag{28}$$

Now, it suffices to bound $\|\pi_n(\cdot|s) - \pi_n^*(\cdot|s)\|_1$, which follows Lemma 7 of (Li and Zhang, 2022). First notice that $\mathbb{E}_{Q_n}\left[u_{n,e}\right] = u_n^*$, which is by the definition of u_n^* in equation (6). Since $h_{n,e}$ corresponds to $u_{n,e}$ in MFTPL-P, this implies $\mathbb{E}_{Q_n}\left[h_{n,e}(\cdot|s)\right] = \pi_n^*(\cdot|s)$. Now that each $h_{n,e}(\cdot|s)$ can be viewed as Multinoulli random variable on $\Delta(\mathcal{A})$ with expectation $\pi_n^*(\cdot|s)$, while $\pi_n(\cdot \mid s) := \frac{1}{E}\sum_{e=1}^{E} h_{n,e}(\cdot|s)$, we apply the concentration inequality for Multinoulli random variables (Qian et al., 2020; Weissman et al., 2003) and conclude given $n \in [N]$ and $s \in \mathcal{S}$, for any $\delta_0 \in (0,1]$, u_n, u_n^*, g_n, satisfies that with probability at least $1 - \frac{\delta_0}{NS}$,

$$\|\pi_n(\cdot|s) - \pi_n^*(\cdot|s)\|_1 \le \sqrt{\frac{2A\left(\ln(NS)+\ln(\frac{2}{\delta_0})\right)}{E}}.$$

By applying union bound over all $n \in [N]$ and all $s \in \mathcal{S}$, we conclude that for any $\delta \in (0,1]$, the sequence of $\{u_n\}$, $\{u_n^*\}$, $\{g_n\}$, satisfies that with probability at least $1-\delta/6$,

$$\begin{aligned}
\sum_{n=1}^{N} \langle g_n, u_n - u_n^* \rangle \le& \sum_{n=1}^{N} \mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\left[\|\pi_n(\cdot|s) - \pi_n^*(\cdot|s)\|_1\right] \\
\le& \sum_{n=1}^{N} \mathbb{E}_{(s,\pi^{\exp}(s))\sim D_n}\sqrt{\frac{2A\left(\ln(NS)+\ln(\frac{12}{\delta})\right)}{E}} \\
=& N\sqrt{\frac{2A\left(\ln(NS)+\ln(\frac{12}{\delta})\right)}{E}}.
\end{aligned} \tag{29}$$

□

Algorithm 4 BOOTSTRAP-DAGGER

```
Input: MDP $\mathcal{M}$, expert $\pi^{\exp}$, policy class $\mathcal{B}$, oracle $\mathcal{O}$, sample size per round $K$, ensemble size $E$.
Initialize $D = \emptyset$.
for $n = 1, 2, \ldots, N$ do
  for $e = 1, 2, \ldots, E$ do
    $\pi_{n,e} \leftarrow$ TRAIN-BASE$(D)$
  end for
  Set $\pi_n(a \mid s) := \frac{1}{E} \sum_{e=1}^{E} I(\pi_{n,e}(s) = a)$.
  $D_n = \left\{(s_{n,k}, \pi^{\exp}(s_{n,k}))\right\}_{k=1}^{K} \leftarrow$ sample $K$ states i.i.d. from $d_{\pi_n}$ by rolling out $\pi_n$ in $\mathcal{M}$, and query expert $\pi^{\exp}$ on these states.
  Aggregate datasets $D \leftarrow D \cup D_n$.
end for
Return $\hat{\pi} \leftarrow$ AGGREGATE-POLICIES$(\left\{\pi_{n,e}\right\}_{n=1,e=1}^{N+1,E})$
function TRAIN-BASE $(D)$:
   $\tilde{D} \leftarrow$ Sample $|D|$ i.i.d. samples $\sim \mathrm{Unif}(D)$ with replacement.
   Return $h \leftarrow \mathcal{O}(\tilde{D})$.
Return $h \leftarrow \mathcal{O}(D \cup Q)$.
function AGGREGATE-POLICIES $(\left\{\pi_{n,e}\right\}_{n=1,e=1}^{N+1,E})$:
   Sample $\hat{n} \sim \mathrm{Unif}([N])$
   Return $\pi_{\hat{n}}(a \mid s) := \frac{1}{E} \sum_{e=1}^{E} I(\pi_{\hat{n},e}(s) = a)$.
```

C. Experimental details

C.1. Full version of BOOTSTRAP-DAGGER

We present the full version of BOOTSTRAP-DAGGER in Algorithm 4.

C.2. Additional Implementation Details

All experiments were conducted on an Ubuntu machine equipped with a 3.3 GHz Intel Core i9 CPU and 4 NVIDIA GeForce RTX 2080 Ti GPUs. Our project is built upon the source code of Disagreement-Regularized Imitation Learning (https://github.com/xkianteb/dril) and shares the same environment dependencies. We have inherited some basic functions and implemented a new online learning pipeline that supports parallelized ensemble policies, in which we instantiate DAGGER, MFTPL-P, and BOOTSTRAP-DAGGER. For each algorithm and experimental setting, we executed ten runs using random seeds ranging from 1 to 10. The detailed control task names are "HalfCheetahBulletEnv-v0", "AntBulletEnv-v0", "Walker2DBulletEnv-v0", and "HopperBulletEnv-v0". For code and more information see https://github.com/liyichen1998/BootstrapDagger-MFTPLP

Table 3: Hyperparameters for Continuous Control Experiment

Hyperparameter	**Values Considered**	**Choosen Value**
Ensemble Size	[1,5,25]	[1,5,25]
Perturbation Size X for MFTPL-P	[7,15,31,62,125,250,500]	[0,7,15,31,62,125]
Hidden Layer Size (non-realizable)	[2,4,8,12,16,24,32,64]	4 (Ant), 8 (Hopper), 12 (Half-Chettah), 24 (Walker)
Learning Rounds (realizable)	[10,20,50,100]	20 (Ant & Hopper), 50 (Half-Cheetah & Walker)
Learning Rounds (non-realizable)	[10,20,40,50,80,100,200]	40 (Ant & Hopper), 50 (Half-Cheetah & Walker)
Data Per Round	[10,20,50,100,200,1000]	50
Learning Rate	2.5×10^{-4}	2.5×10^{-4}
Batch Size	[50,100,200,500,1000]	200
Train Epoch	[500,1000,2000,10000]	2000
Parallel Environments	[5,16,25]	25

As shown in Table 3, we present the considered hyperparameters along with their chosen values. Overall, hyperparameters related to environment interactions, like learning rounds and data per round, are selected to generate a 'favorable' learning curve for DAGGER, enabling it to learn relatively fast but not converge within just a few rounds. The perturbation sizes for MFTPL-P are chosen by the ratio of the perturbation dataset size to 1000 (the maximum size of the cumulative dataset for realizable Ant, Hopper), following the sequence of $\left\{\lfloor 1000/2^i \rfloor | 1 \leq i \leq 7\right\}$. For hyperparameters related to neural network training, such as batch size, training epoch, etc., those are selected to ensure a faithful implementation of a offline learning oracle without imposing heavy computational overhead.

For the justification of using 2000 SGD iterations without validation set, we provide the following reasoning:

1. On the performance of the oracle, 2000 SGD iterations suffices to support a comprehensive comparison over different algorithms, as shown in Figure 7.

2. On the faithfulness of implementing a offline learning oracle, the returned policy should be the best policy on the input data, where the generalization error is not considered. In this case, splitting out a portion of input data for validation may deviate from the definition of the computational oracle.

3. On the reproducibility of the experiment, the implementation of validation set may vary and provide additional noises, i.e. whether the validation set is resampled for each input dataset or gathered incrementally through N rounds and kept unseen from the learner. Though it would be interesting to compare the difference between these and our oracle.

Running Time and Memory Comparison.

We present the running time and memory comparisons in the realizable setting from Section 5.3. As shown in Figure 8, BC, DAGGER, and BD-1 have similar running times across different tasks. Benefiting from the parallel implementation of ensemble models, BD-5 only takes twice as long as the baselines, while those with an ensemble size of 25 require approximately 5 times longer. Considering overall performance and running time, BD-5 is more favorable for practical applications.

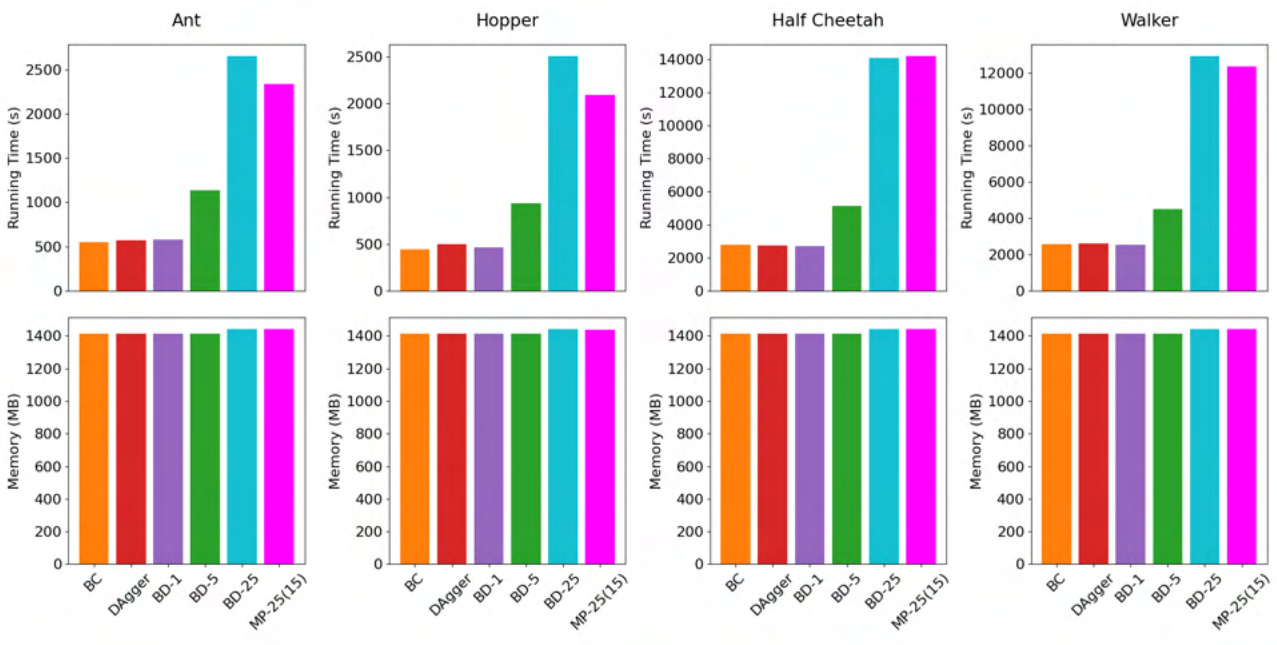

Figure 8: Comparison of running time and memory for different algorithms under the realizable setting.

Comparison of Alternative AGGREGATE-POLICIES Functions for BOOTSTRAP-DAGGER

Additionally, we compare the performance of our BOOTSTRAP-DAGGER algorithm with the AGGREGATE-POLICIES part changed to randomization over the ensemble trained. Our results are shown in Figure 9, where BD-5 and BD-25 represent our BOOTSTRAP-DAGGER algorithm with ensemble sizes 5 and 25 respectively, and "BD-5 random" and "BD-25 random" are modifications of BOOTSTRAP-DAGGER with the AGGREGATE-POLICIES part changed to returning $\bar{\pi}_{N+1}(a \mid s) = \frac{1}{E}\sum_{e=1}^{E} \pi_{N+1,e}(a \mid s)$.

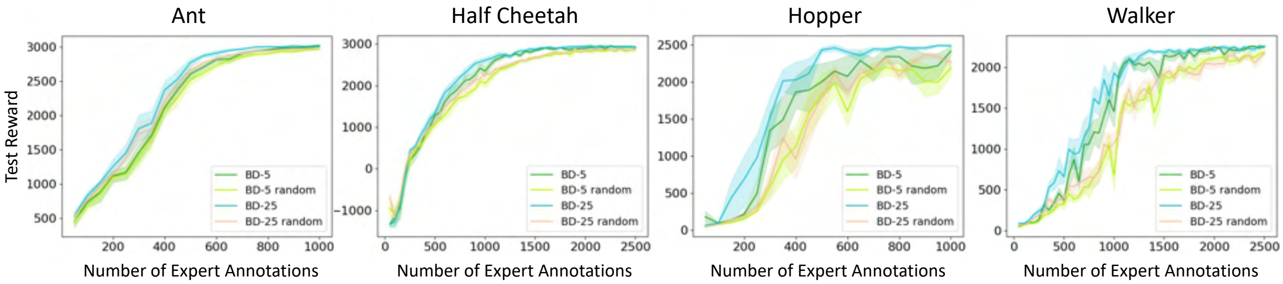

Figure 9: Comparison of the original BOOTSTRAP-DAGGER algorithm with its variant with AGGREGATE-POLICIES changed to returning a policy that is a uniform randomization over the ensemble.

Overall, using bootstrap mean (Bagging) proves marginally better than using bootstrap with randomization for final policy evaluation. This justifies the choice of using the ensemble mean instead of the original AGGREGATE-POLICIES for continuous control.

C.3. Full Results from Section 5.2

In this section, we present all result plots from Section 5.2, including those omitted due to space constraints. As shown in Figure 10, we include the performance of BOOTSTRAP-DAGGER and BC for the linear nonrealizable experiment in Ant and Hopper. Evidently, the performance of BD improves with increasing ensemble size, with BD-25 achieving comparable performance to MFTPL-P. Notice that the performance of BC varies significantly across tasks in the linear model setting, which we leave for further investigation.

C.4. Full results from Section 5.3

In this section, we present all result plots from Section 5.3, including those omitted due to space constraints. Additionally, since imitation learning agents do not usually have access to the ground truth reward, we also evaluate π using a more "objective" performance measure, i.e, its average imitation loss:

$$\text{Imitation Loss}(\pi) = \frac{1}{T}\sum_{i=1}^{T} \frac{1}{|\tau_i^\pi|} \sum_{s \in \tau_i^\pi} \tilde{\ell}(\pi(s), \bar{\pi}^{\exp}(s)),$$

this measures the policy's deviation from the mean action of the expert policy $\bar{\pi}^{\exp}(s)$. Its expectation, $L(\pi) := \mathbb{E}_{s\sim d_\pi, a\sim\pi(\cdot|s)}\ell(a, \pi^{\exp}(s))$, has been a central optimization objective in imitation learning works such as DAGGER (Ross et al., 2011, Eq. (1)) and subsequent works (Ross and Bagnell, 2014; Sun et al., 2017; Cheng et al., 2019a) including ours.

In the following, Figures 11 and 12 present the results of experiments with realizable and non-realizable experts using MLP as the base class. These include the performance of MP-25(X) with varying perturbation sizes and the imitation loss of different algorithms as a function of expert annotation size. Although the advantages of sample-based perturbation are less apparent in the MLP-based experiments, MP-25(15) still noticeably outperforms MP-25(0) in realizable Ant.

By examining the correlations between test rewards and imitation losses, a discrepancy in imitation loss usually correlates with a gap in test reward, which shows the practical relevance of minimizing a policy's imitation loss – it is a reasonable proxy of policy's expected performance.

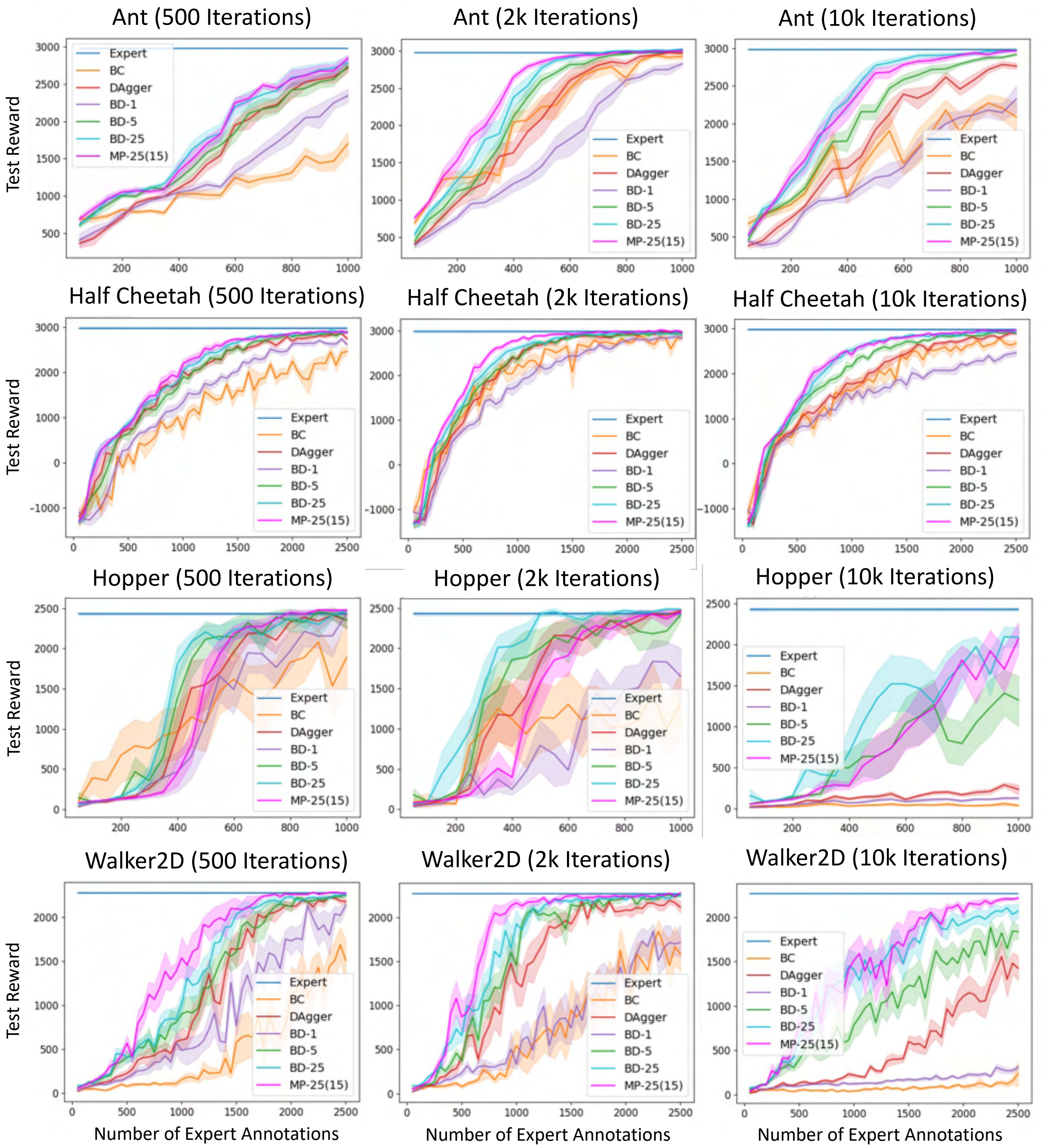

Figure 7: Comparison of 500, 2000, and 10000 iteration steps on continuous control tasks with realizable expert.

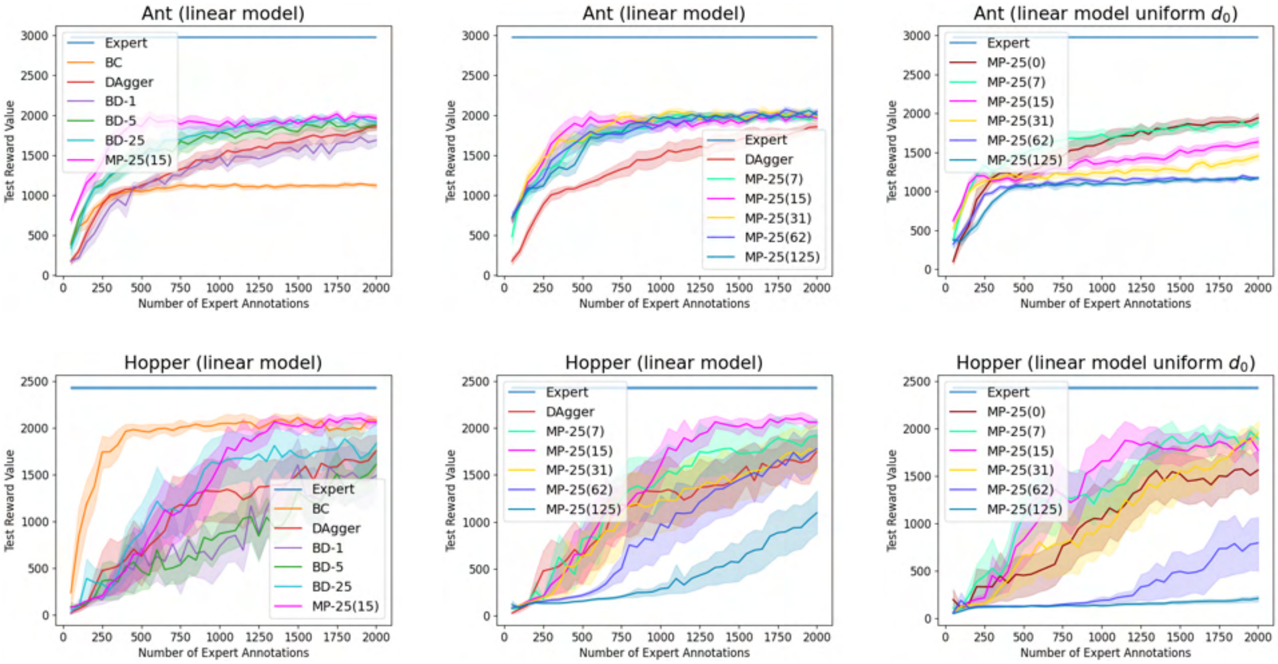

Figure 10: Continuous control experiments with linear model and nonrealizable noisy expert.

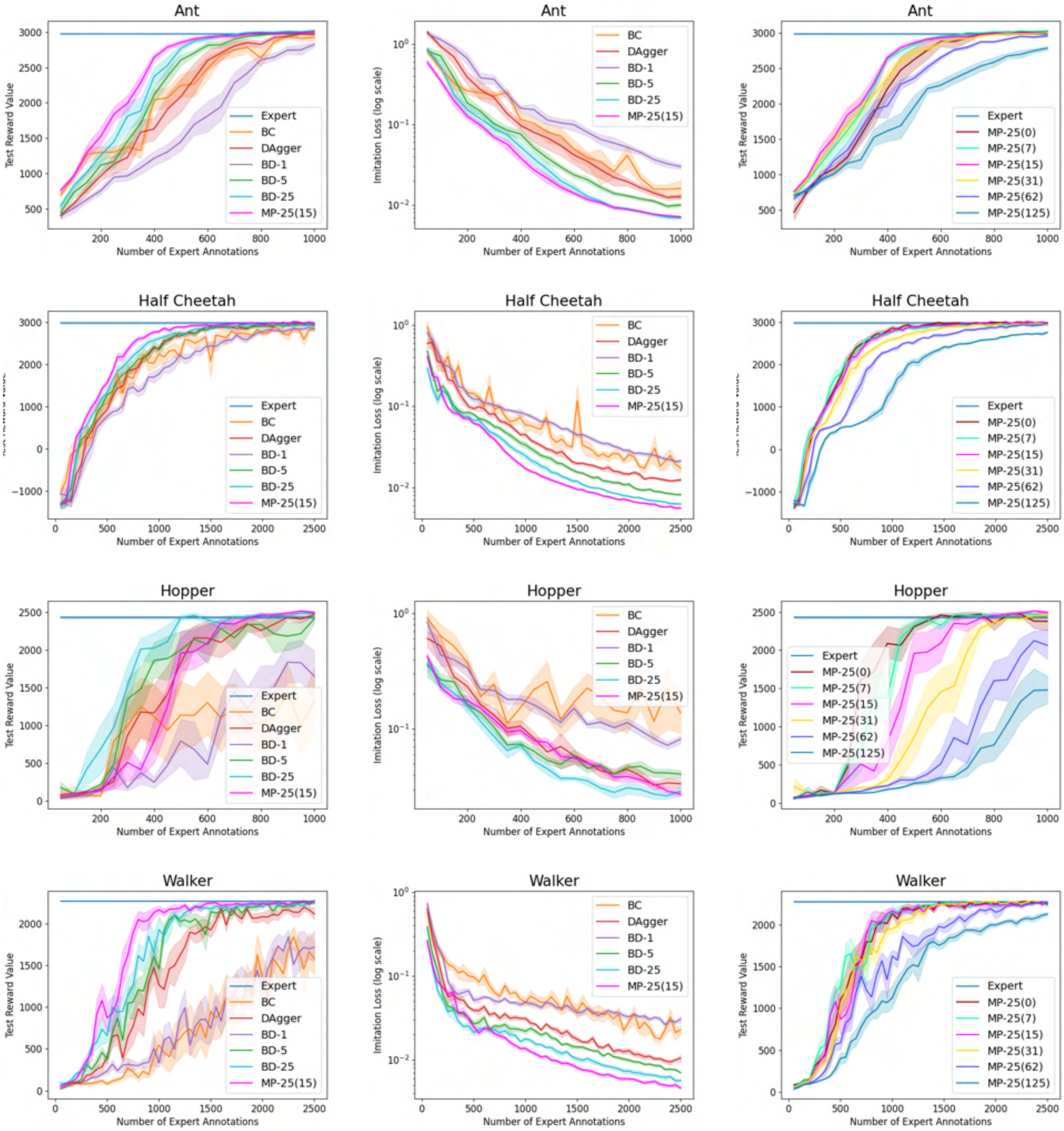

Figure 11: Continuous control experiments with realizable noisy expert.

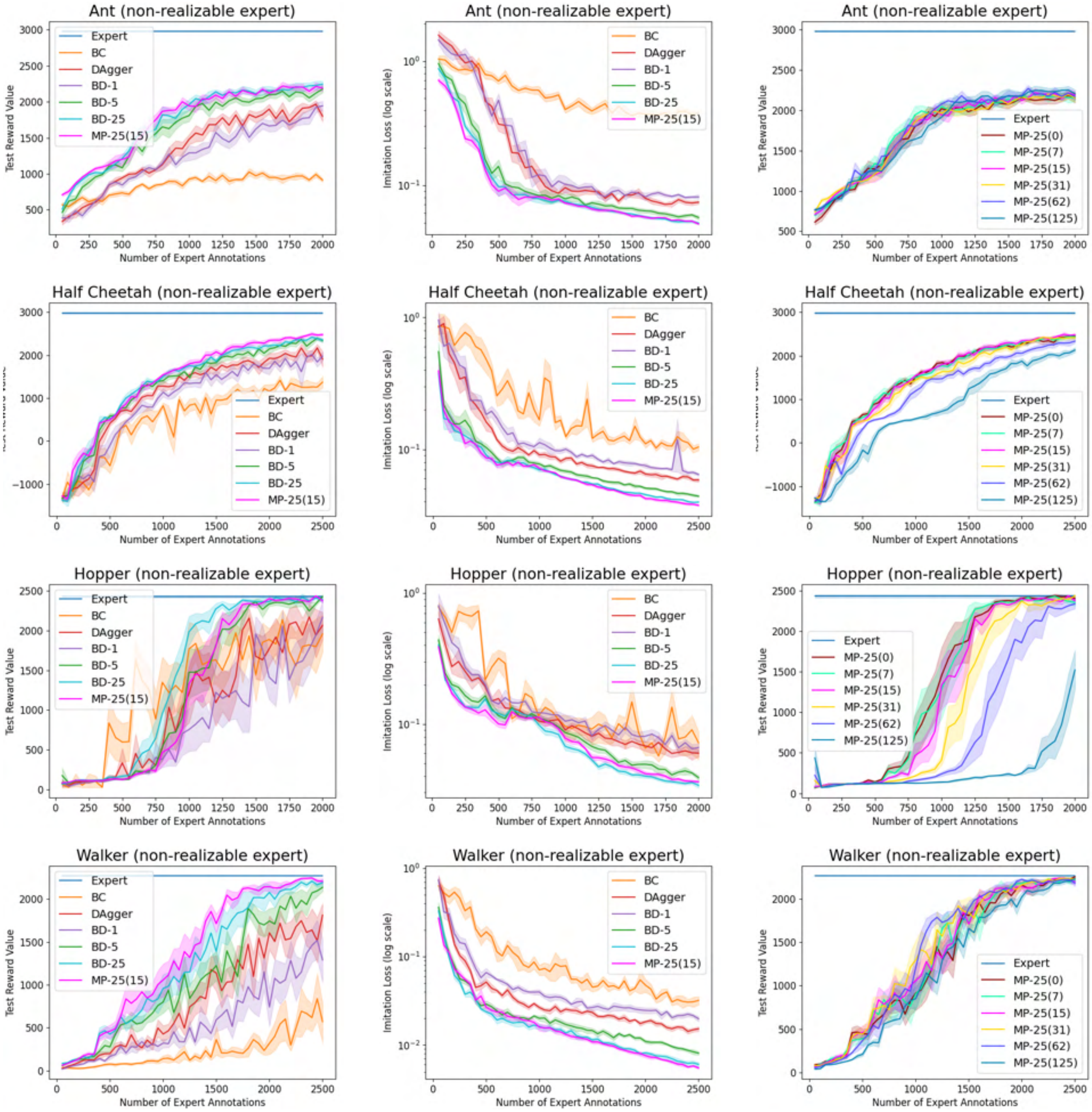

Figure 12: Comparison between algorithms with non-realizable noisy expert.

C.5. Full results from Section 5.4 and Data Visualization via t-SNE (Van der Maaten and Hinton, 2008).

In this section, we present all result plots from Section 5.4, including those omitted due to space constraints, as shown in Figure 13.

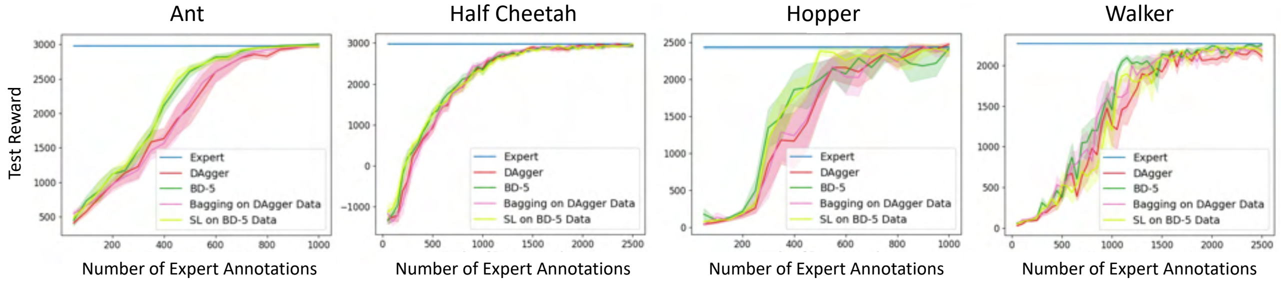

Figure 13: Full results on comparing BD-5 and DAGGER, along with the two additional approaches in Section 5.4.

To further understand how the data quality collected by BD-5 improves over DAGGER, we visualize the states collected and queried by different algorithms in Section 5.3 via t-SNE. As motivation, Figure 14 shows a comparison between states of offline expert demonstration and states queried by DAGGER in Hopper. It can be seen that with the same expert annotation budget, DAGGER collects a dataset that encompasses a broader support compared to the expert, while the policy trained over it achieves a higher average reward.

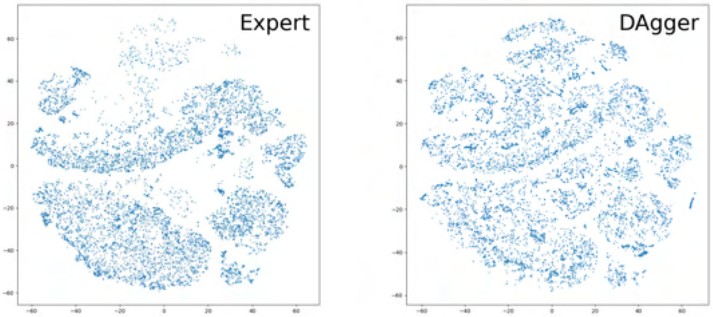

Figure 14: Two-dimensional t-SNE visualization of states collected by expert and DAGGER in continuous control task Hopper, using the same mapping. It can be observed that the support of state distribution by DAGGER contains regions (top and middle) that are not supported by the expert's state distribution. Over 10 repeated trials, supervised learning over the datasets collected by expert and DAGGER have average reward of 1320 and 2470, respectively.

Figure 15,16,17,18 showcase the t-SNE visualization of states obtained by different algorithms across four environments under the realizable expert setting, using the same mapping for the same environment. State points are color-mapped from blue to red based on their arrival rounds. As presented in these figures, the observations reaffirm the findings of Section 5.4. For example, in the state visualization of Ant (Figure 15), we notice similar state coverage among DAGGER-style algorithms, which is distinct from the expert's distribution. This suggests that BD-5 may not collect annotations over different state distributions than DAGGER. Meanwhile, the color of points within the zoomed-in area for BD-5, BD-25, and MP-25(15) appears bluer than DAGGER, indicating a more efficient exploration by ensembles in regions beyond the support of the expert's state distribution. From these results, we can see that BD actively explores the state space, swiftly adapting to and rectifying its errors, ensuring a more rapid and efficient learning process compared to DAGGER.

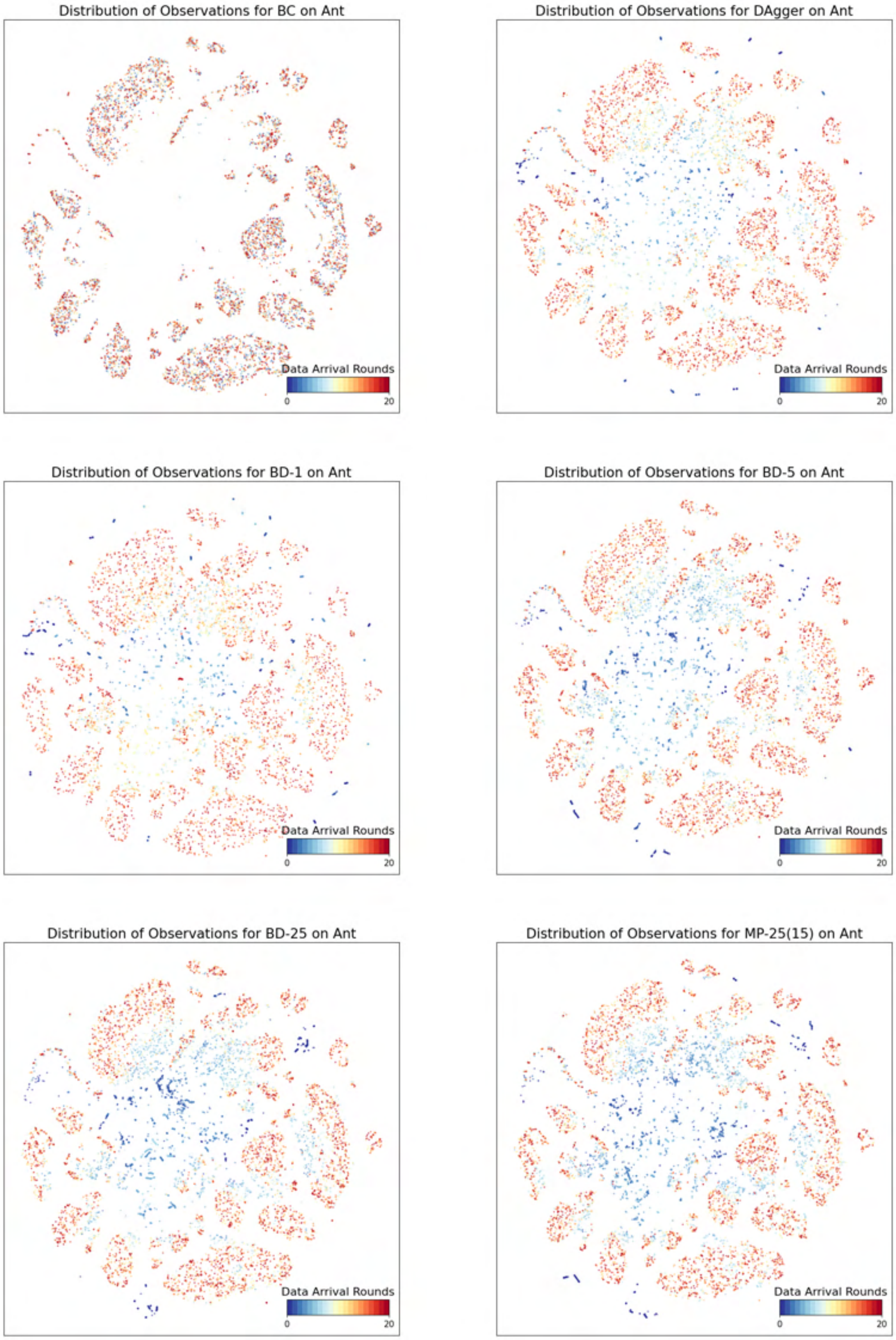

Figure 15: Two-dimensional t-SNE visualizations of Ant environment ss collected by different algorithms.

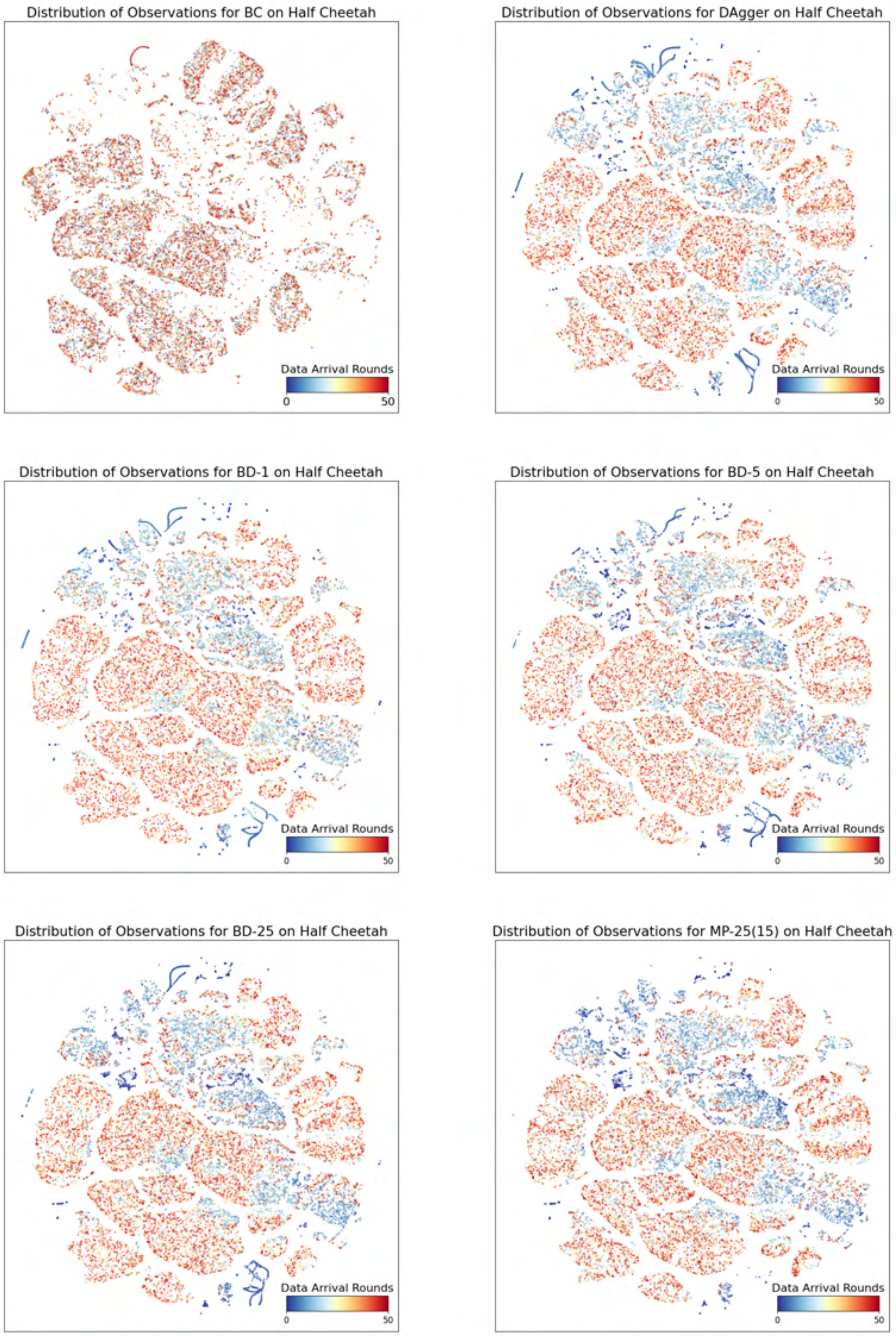

Figure 16: Two-dimensional t-SNE visualizations of Half-Cheetah environment states collected by different algorithms.

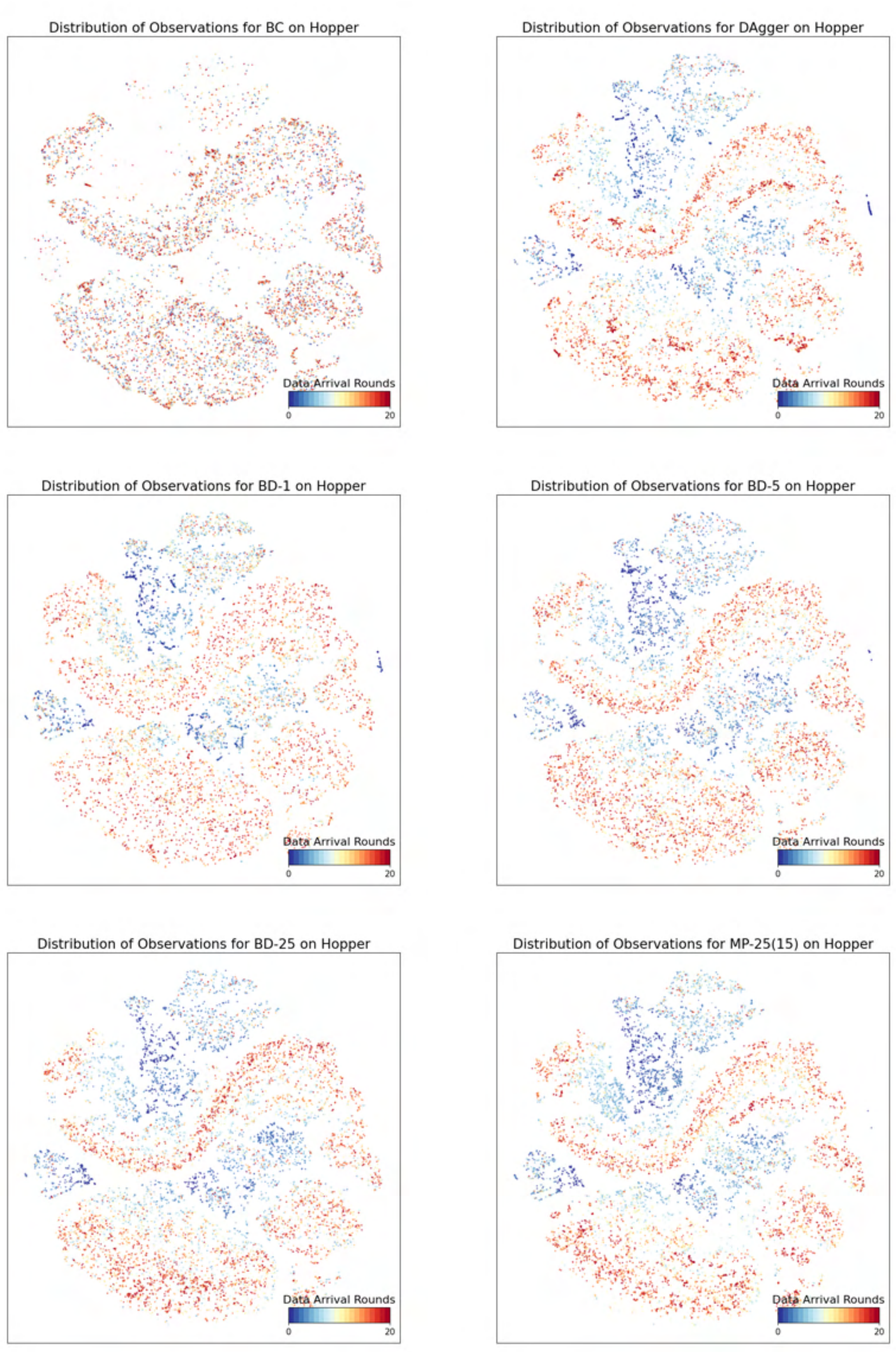

Figure 17: Two-dimensional t-SNE visualizations of Hopper environment states collected by different algorithms.

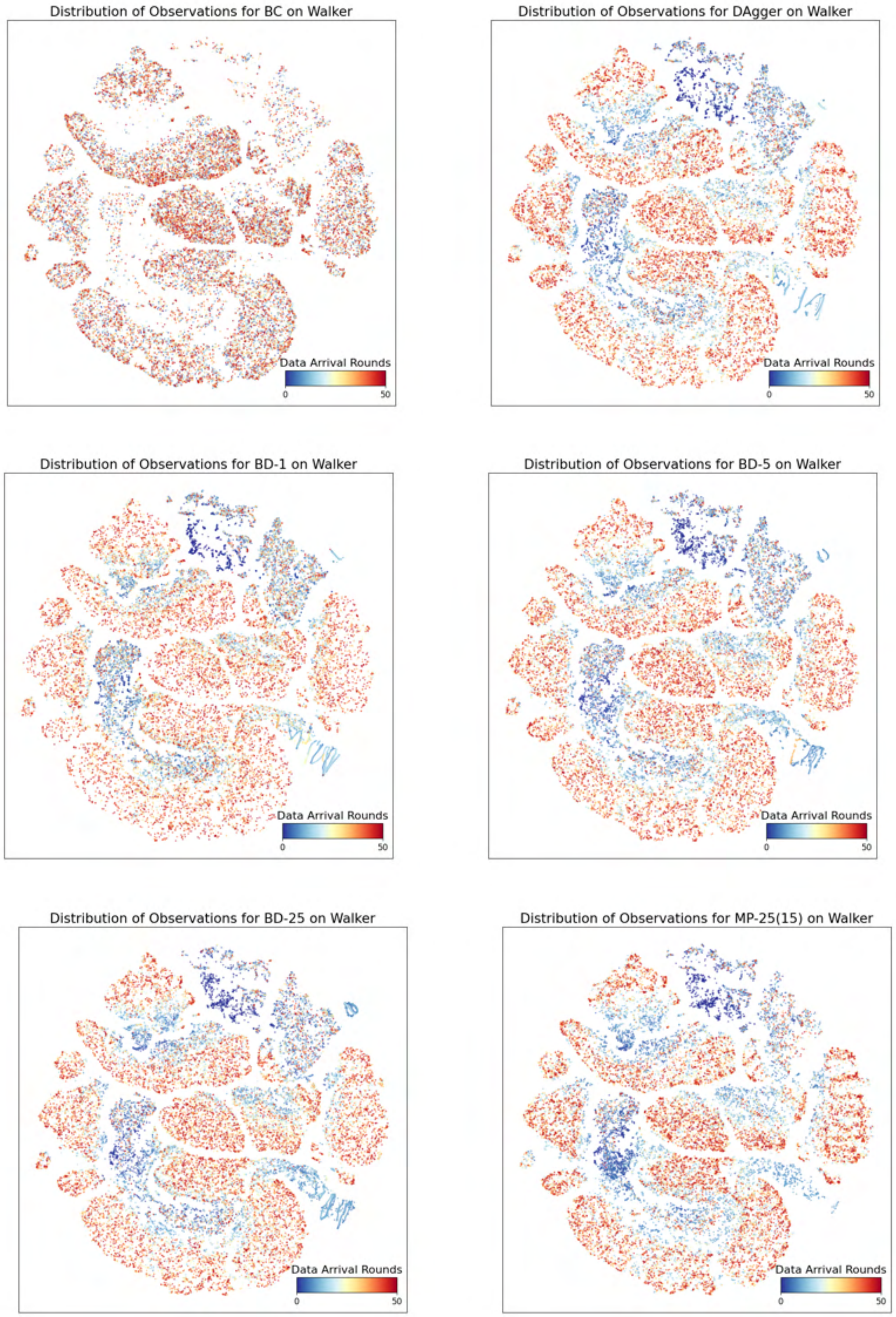

Figure 18: Two-dimensional t-SNE visualizations of Walker environment states collected by different algorithms.