Interactive and Hybrid Imitation Learning: Provably Beating Behavior Cloning

Yichen Li
University of Arizona
yichenl@arizona.edu

Chicheng Zhang
University of Arizona
chichengz@cs.arizona.edu

Abstract

Imitation learning (IL) is a paradigm for learning sequential decision-making policies from experts, leveraging offline demonstrations, interactive annotations, or both. Recent advances show that when annotation cost is tallied per trajectory, Behavior Cloning (BC)—which relies solely on offline demonstrations—cannot be improved in general, leaving limited conditions for interactive methods such as DAgger to help. We revisit this conclusion and prove that when the annotation cost is measured per state, algorithms using interactive annotations can provably outperform BC. Specifically: (1) we show that STAGGER, a one-sample-per-round variant of DAgger, provably beats BC under low-recovery-cost settings; (2) we initiate the study of hybrid IL where the agent learns from offline demonstrations and interactive annotations. We propose WARM-STAGGER whose learning guarantee is not much worse than using either data source alone. Furthermore, motivated by compounding error and cold-start problem in imitation learning practice, we give an MDP example in which WARM-STAGGER has significant better annotation cost; (3) experiments on MuJoCo continuous-control tasks confirm that, with modest cost ratio between interactive and offline annotations, interactive and hybrid approaches consistently outperform BC. To the best of our knowledge, our work is the first to highlight the benefit of state-wise interactive annotation and hybrid feedback in imitation learning.

1 Introduction

Imitation learning, also known as learning from demonstrations, is a widely-used approach for learning policies to make sequential decisions [45, 6, 5]. In many applications, it offers a preferable alternative to reinforcement learning, as it bypasses the need for carefully designed reward functions and avoids costly exploration [43, 67].

Two prominent data collection regimes exist in imitation learning: offline and interactive. In offline imitation learning, expert demonstration data in the format of trajectories is collected ahead of time, which is a non-adaptive process that is easy to maintain. In contrast, in interactive imitation learning, the learner is allowed to query the expert for annotations in an adaptive manner [51, 50, 67]. The most basic and well-known approach for offline imitation learning is Behavior Cloning [49, 17], which casts the policy learning problem as a supervised learning problem that learns to predict expert actions from states. Although simple and easy to implement, offline imitation learning has the drawback that the quality of the data can be limited [45]. As a result, the trained model can well suffer from compounding error, where imperfect imitation leads the learned policy to enter unseen states, resulting in a compounding sequence of mistakes. In contrast, in interactive imitation learning, the learner maintains a learned policy over time, with the demonstrating experts providing corrective feedback *on-policy*, which enables targeted collection of demonstrations and improves sample efficiency.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Recent work [17], via a sharp analysis of Behavior Cloning, shows that the sample efficiency of Behavior Cloning cannot be improved in general when measuring using the number of trajectories annotated. Interactive methods like DAgger [49] can enjoy sample complexity benefits, but so far the benefits are only exhibited in limited examples, with the most general ones in the tabular setting [46]. This leaves open the question:

> *Can interaction provide sample efficiency benefit for imitation learning under diverse settings, notably in the presence of function approximation?*

In this paper, we make progress towards this question, with a focus on the *deterministically realizable* setting (i.e. the expert policy π^E is deterministic and is in the learner's policy class $\mathcal{B}$). Specifically, we make the following contributions:

1. Motivated by the costly nature of interactive labeling on entire trajectories [29, 37], we propose to measure the cost of annotation using the number of states annotated by the demonstrating expert. We propose a general state-wise interactive imitation learning algorithm, STAGGER, and show that as long as the expert can recover from mistakes at low cost in the environment [51], it significantly improves over Behavior Cloning in terms of its number of state-wise demonstrations required.
2. Motivated by practical imitation learning applications where sets of offline demonstration data are readily available, we study *hybrid imitation learning*, where the learning agent can additionally query the demonstration expert interactively to improve its performance. We design a hybrid imitation learning algorithm, WARM-STAGGER, and prove that its policy optimality guarantee is not much worse than using either of the data sources alone.
3. Inspired by compounding error [45] and cold start problem [35, 42], we provide an MDP example, for which we show hybrid imitation learning can achieve strict sample complexity savings over using either source alone, and provide simulations that verify this claim.
4. We conduct experiments in MuJoCo continuous control tasks and show that if the cost of state-wise interactive demonstration is not much higher than its offline counterpart, interactive algorithms can enjoy a better cost efficiency than Behavior Cloning. Under some cost regimes and some environments, hybrid imitation learning can outperform approaches that use either source alone.

2 Preliminaries

Basic notation. Define $[n] := \{1, \dots, n\}$. Denote by $\Delta(\mathcal{X})$ the set of probability distributions over a set $\mathcal{X}$. For $u \in \Delta(\mathcal{X})$ and $x \in \mathcal{X}$, we denote by $u(x)$ the x-th coordinate of u and e_x the delta mass on x. We use the shorthand $x_{1:n}$ to represent the sequence $(x_i)_{i=1}^n$; we will also apply this shorthand to tuples, e.g. using $(x, y)_{1:n}$ to denote $(x_i, y_i)_{i=1}^n$. We will frequently use the Hellinger distance to measure the difference between two distributions: $D_H^2(\mathbb{P}, \mathbb{Q}) = \int (\sqrt{\frac{d\mathbb{P}}{d\omega}} - \sqrt{\frac{d\mathbb{Q}}{d\omega}})^2 d\omega$, where $\mathbb{P}$ and $\mathbb{Q}$ share a dominating measure ω.

Episodic Markov decision process and agent-environment interaction. A fixed-horizon episodic MDP $\mathcal{M}$ is defined as a tuple $(\mathcal{S}, \mathcal{A}, P, \rho, \mathcal{R}, H)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the transition dynamics, $\rho \in \Delta(\mathcal{S})$ is the initial state distribution, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \Delta([0, 1])$ denotes the reward distribution, and H denotes episode length. [1] Given a stationary policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, we use $\pi(\cdot|s)$ to denote the action distribution of π on state s. Rolling out policy π in MDP $\mathcal{M}$ gives a distribution over trajectories $\tau = (s_h, a_h, r_h)_{h=1}^H$ by first drawing the initial state $s_1 \sim \rho$, and then iteratively taking actions $a_h \sim \pi(\cdot \mid s_h)$, receiving rewards $r_h \sim \mathcal{R}(s_h, a_h)$, and transitioning to the next state $s_{h+1} \sim P(s_h, a_h)$ (except at step H). Let $\mathbb{E}^\pi$ and $\mathbb{P}^\pi$ denote expectation and probability law for $(s_h, a_h, r_h)_{h=1}^H$ induced by π and $\mathcal{M}$. Given π, denote by $d^\pi(s) := \frac{1}{H} \sum_{h=1}^H \mathbb{P}^\pi(s_h = s)$ its state visitation distribution. The expected return of policy π is defined as $J(\pi) := \mathbb{E}^\pi \left[\sum_{h=1}^H r_h \right]$, and the value functions of π are given by $V_h^\pi(s) := \mathbb{E}^\pi \left[\sum_{h'=h}^H r_{h'} \mid s_h = s \right]$, and $Q_h^\pi(s, a) := \mathbb{E}^\pi \left[\sum_{h'=h}^H r_{h'} \mid s_h = s, a_h = a \right]$. If for

[1] Here we assume that the transition dynamics and reward functions are stationary, i.e., it does not depend on time step in the episode. To translate our results to the nonstationary transition dynamics and reward setting (to have a fair comparison with [e.g., 17, 47]), we can augment the state with a step index, i.e. define $\tilde{s} = (s, h)$.

policy π, step h, and state s, $\pi(\cdot \mid s)$ is the delta-mass on an action, we also sometimes slightly abuse the notation and let $\pi(s)$ denote that action.

Additional policy-related notations. Throughout, we assume the access to a class $\mathcal{B}$ of stationary policies of finite size B. A (MDP, Expert) pair $(\mathcal{M}, \pi^{\mathrm{E}})$ is said to be μ-recoverable if for all $h \in [H]$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $Q_h^{\pi^{\mathrm{E}}}(s,a) - V_h^{\pi^{\mathrm{E}}}(s) \leq \mu$. Additionally, we assume normalized return [17], where there exists some $R > 0$ such that for any trajectory $(s_h, a_h, r_h)_{h=1}^H$ rolled out in $\mathcal{M}$, $\sum_{h=1}^H r_h \in [0, R]$. It always holds that $\mu \leq R$, while in many applications we expect μ to be much smaller [51, Section 2.2]. Throughout this paper, we make the assumption that our imitation learning problem is *deterministically realizable*:

Assumption 1 (Deterministic Realizability)**.** *The expert policy π^{E} is deterministic and is contained in the learner's policy class $\mathcal{B}$.*

In our algorithm and analysis, we frequently use the following "convexification" of policy class $\mathcal{B}$:

Definition 1 (Each-step mixing and each-step mixing policy class)**.** *For u in $\Delta(\mathcal{B})$, define its each-step mixing policy $\bar{\pi}_u$ as a stationary policy from $\mathcal{S}$ to $\Delta(\mathcal{A})$:*

$$\bar{\pi}_u(a \mid s) := \sum_{\pi \in \mathcal{B}} u(\pi)\pi(a \mid s).$$

Define $\bar{\Pi}_{\mathcal{B}} := \{\bar{\pi}_u : u \in \Delta(\mathcal{B})\}$ as $\mathcal{B}$'s each-step mixing policy class.

An each-step mixing policy $\bar{\pi}_u \in \bar{\Pi}_{\mathcal{B}}$ can be executed by drawing $\pi \sim u$ freshly-at-random at each step $h \in [H]$ and taking action $a_h \sim \pi(\cdot|s_h)$ (e.g. [33, 34]).

Offline imitation learning and Behavior Cloning. In offline imitation learning, the agent is given a collection of expert trajectories $\mathcal{D} = \{\tau_1, \ldots, \tau_{N_{\mathrm{off}}}\}$, where $\tau_i = (s_{i,h}, a_{i,h})_{h=1}^H$ is the i-th (reward-free) trajectory, all of which are drawn iid from the trajectory distribution of expert policy π^{E}. Behavior Cloning finds policy $\pi \in \mathcal{B}$ that minimizes its log loss on expert's actions on the seen states:

$$\hat{\pi} = \operatorname*{argmin}_{\pi \in \mathcal{B}} \sum_{i=1}^{N_{\mathrm{off}}} \sum_{h=1}^{H} \log \frac{1}{\pi(a_{i,h} \mid s_{i,h})}.$$

Recent work of [17] establishes a sharp horizon-independent sample complexity bound of Behavior Cloning, which we recall here:

Theorem 2 ([17])**.** *Suppose Assumption 1 holds, then with probability $1-\delta$, the policy returned by BC $\hat{\pi}$ satisfies:*

$$J(\pi^{\mathrm{E}}) - J(\hat{\pi}) \leq \tilde{O}\left(\frac{R \log B}{N_{\mathrm{off}}}\right).$$

Interactive imitation learning protocol. In interactive IL, the learner has the ability to query the demonstration expert interactively. A first way to model interaction with expert is through a *trajectory-wise demonstration oracle* $\mathcal{O}^{\mathrm{Traj}}$ [51, 17]: given a state sequence $(s_h)_{h=1}^H$, return $(a_h)_{h=1}^H$ such that $a_h = \pi^{\mathrm{E}}(s_h)$ for all h. Subsequent works have considered modeling the interaction with expert as interacting with a *state-wise demonstration oracle* [22, 7, 40, 55] $\mathcal{O}^{\mathrm{State}}$: given a state s_h and step h, return $a_h = \pi^{\mathrm{E}}(s_h)$. We consider the learner interacting with the environment and demonstration oracles using the following protocol:

For $i = 1, 2, \ldots$

- Select policy π^i and roll it out in $\mathcal{M}$, observing a reward-free trajectory $(s_1, a_1, \ldots, s_H, a_H)$.
- Query the available oracle(s) to obtain expert annotations.

Goal: Return policy $\hat{\pi}$ such that $J(\pi^{\mathrm{E}}) - J(\hat{\pi})$ is small, with a few number of queries to $\mathcal{O}^{\mathrm{Traj}}$ or $\mathcal{O}^{\mathrm{State}}$.

In practice, we expect the cost of querying $\mathcal{O}^{\mathrm{Traj}}$ to be higher than that of collecting a single offline expert trajectory [29]. Since H queries to $\mathcal{O}^{\mathrm{State}}$ can simulate one query to $\mathcal{O}^{\mathrm{Traj}}$, the cost of a single $\mathcal{O}^{\mathrm{State}}$ query should be at least $\frac{1}{H}$ the cost of $\mathcal{O}^{\mathrm{Traj}}$. Consequently, we also expect one $\mathcal{O}^{\mathrm{State}}$ query

Algorithm 1 STAGGER: DAgger with State-wise annotation oracle

Input: MDP $\mathcal{M}$, state-wise expert annotation oracle $\mathcal{O}^{\mathrm{State}}$ with query budget N_{int}, stationary policy class $\mathcal{B}$, online learning oracle $\mathbb{A}$.
for $n = 1, \ldots, N_{\mathrm{int}}$ **do**
  Query $\mathbb{A}$ and receive π^n.
  Execute π^n and sample $s^n \sim d^{\pi^n}$. Query $\mathcal{O}^{\mathrm{State}}$ for $a^{*,n} = \pi^{\mathrm{E}}(s^n)$.
  Update $\mathbb{A}$ with loss function

$$\ell^n(\pi) := \log\left(\frac{1}{\pi(a^{*,n}|s^n)}\right). \tag{1}$$

end for
Output $\hat{\pi}$, a first-step uniform mixture of $\{\pi^n\}_{n=1}^{N_{\mathrm{int}}}$.

to be more expensive than obtaining an additional offline (state, expert action) pair. We denote the ratio between these two costs as C, where $C \geq 1$ is an application-dependent constant. [2]

3 State-wise Annotation in Interactive Imitation Learning

Recent work [17] on refined analysis of Behavior Cloning (BC) casts doubt on the utility of interaction in imitation learning: when measuring sample complexity in the number of trajectories annotated, BC is minimax optimal even among interactive algorithms [17, Corollary 2.1 and Theorem 2.2]. Although benefits of interactive approaches have been shown in specific examples, progresses so far have been sparse [17, 46], with the most general results in the less-practical tabular setting [46]. In this section, we reexamine this conclusion and show that interaction benefits imitation learning in a fairly general sense in the general function approximation setting: when measuring sample complexity using the number of state-wise annotations, we design an interactive algorithm with sample complexity better than BC, as long as the expert policy has a low recovering cost μ in the environment.

3.1 Interactive IL Enables Improved Sample Complexity with State-wise Annotations

Our algorithm STAGGER (short for State-wise DAgger), namely Algorithm 1, interacts with the demonstration expert using a state-wise annotation oracle $\mathcal{O}^{\mathrm{State}}$. Similar to the original DAgger [51], it requires base policy class $\mathcal{B}$ and reduces interactive imitation learning to no-regret online learning. At iteration n, it rolls out the current policy π^n obtained from an online learning oracle $\mathbb{A}$ and samples state s^n from d^{π^n}. A classical example of $\mathbb{A}$ is the exponential weight algorithm that chooses policies from $\bar{\Pi}_{\mathcal{B}}$ ([9]; see also Proposition 41 in Appendix F). It then queries $\mathcal{O}^{\mathrm{State}}$ to get expert action $a^{*,n}$ and updates $\mathbb{A}$ with loss function $\ell^n(\pi)$ induced by this new example (Eq. (1)). The final policy $\hat{\pi}$ is returned as a uniform first-step mixture of the historical policies $\{\pi^n\}_{n=1}^{N_{\mathrm{int}}}$, i.e., sample one π^n uniformly at random and execute it for the episode. In contrast to the DAgger variants analyzed in [17, 46] , which trains a distinct policy at each step—yielding H policies in total—and employs trajectory-level annotations, our algorithm utilizes parameter sharing of the policy representation across all steps and uses state-wise annotations.[3]

We show the following performance guarantee of Algorithm 1 with $\mathbb{A}$ instantiated as the exponential weight algorithm:

Theorem 3. *Suppose* STAGGER *is run with a state-wise expert annotation oracle $\mathcal{O}^{\mathrm{State}}$, an MDP $\mathcal{M}$ where $(\mathcal{M}, \pi^{\mathrm{E}})$ is μ-recoverable, a policy class $\mathcal{B}$ such that deterministic realizability (Assumption 1) holds, and the online learning oracle $\mathbb{A}$ set as the exponential weight algorithm with decision space*

[2] For practical settings such as human-in-the-loop learning with expert interventions [37, 63], obtaining a short segment of corrective demonstrations may be cheaper than querying $\mathcal{O}^{\mathrm{State}}$ for each state therein. Thus, our cost model may need refinements to match practical applications, which we leave as interesting future work.

[3] A one-sample-per-iteration version of AggreVate [50] has been analyzed in [2, Section 15.5]. Our analysis follows a similar structure and further takes direct advantage of the expert action feedback in DAgger and the deterministic realizability assumption to get refined sample complexity.

$\bar{\Pi}_{\mathcal{B}}$. *Then it returns* $\hat{\pi}$ *such that, with probability at least* $1-\delta$,

$$J(\pi^{\mathrm{E}}) - J(\hat{\pi}) \leq \mu H \cdot \frac{\log(B) + 2\log(1/\delta)}{N_{\mathrm{int}}}.$$

Theorem 3 shows that STAGGER returns a policy of suboptimality $O(\frac{\mu H \log B}{N_{\mathrm{int}}})$ using N_{int} interactive state-wise annotations from the expert. In comparison, with the cost of N_{int} state-wise annotations, one can obtain $\frac{CN_{\mathrm{int}}}{H}$ trajectory-wise annotations; [17]'s analysis shows that Behavior Cloning with this number of trajectories from π^{E} returns a policy of suboptimality $O(\frac{RH \log B}{CN_{\mathrm{int}}})$ (recall Theorem 2). Thus, if $C \ll \frac{R}{\mu}$, Algorithm 1 has a better cost-efficiency guarantee than Behavior Cloning.

We now sketch the proof of Theorem 3. In line with [17], we define the online, on-policy state-wise estimation error as

$$\mathrm{OnEst}_N^{\mathrm{State}} := \sum_{n=1}^{N} \mathbb{E}_{s\sim d^{\pi^n}} \left[D_{\mathrm{H}}^2(\pi^n(\cdot \mid s), \pi^{\mathrm{E}}(\cdot \mid s)) \right].$$

The proof proceeds by bounding this error and translating it to the performance difference between $\hat{\pi}$ and π^{E}. While our definition of estimation error is similar to [17, Appendix C.2], their definition requires all H states per trajectory, while ours depends on the distribution over a state sampled uniformly from the rollout of policy π^n. This enables each labeled state to serve as immediate online feedback, fully utilizing the adaptivity of online learning.

3.2 Experimental Comparison

We conduct a simple simulation study comparing the sample efficiency of log-loss Behavior Cloning [17] and STAGGER in four MuJoCo [72, 8] continuous control tasks with $H = 1000$ and pretrained deterministic MLP experts [52, 53]. Considering MuJoCo's low sensitivity to horizon length [17], we reveal expert states one by one along consecutive trajectories for BC to allow fine-grained state-wise sample complexity comparison, while STAGGER queries exactly one state per iteration by sampling from the latest policy's rollout and updating immediately with the expert's annotation. In STAGGER, we implement the online learning oracle $\mathbb{A}$ so that it outputs a policy that approximately minimizes the log loss. In addition to log loss, we also include results with online learning oracle minimizing historical examples' total square loss in Appendix G.2. We defer other implementation details to Appendix G.

Figure 1 shows the performance of the learned policy as a function of the number of state-wise annotations. When each interactive state-wise annotation has the same cost as an offline (state, expert action) pair ($C = 1$), STAGGER has superior and more stable performance than Behavior Cloning. For a given target performance (e.g., near expert-level), STAGGER often requires significantly fewer state-wise annotations than BC—especially on harder tasks—though the gains are less pronounced on easier ones like Ant and Hopper. To highlight sample efficiency, we plot STAGGER using only half the annotation budget of BC; despite this, it still matches or surpasses BC on several tasks, suggesting meaningful benefits from interaction when C is small (e.g., $C = 3$ for Walker).

4 Hybrid Imitation Learning: Combining Offline Trajectory-wise and Interactive State-wise Annotations

Practical deployments of imitation learning systems often learn simultaneously from offline and interactive feedback modalities [27, 20]: for example, in autonomous driving [78, 4, 79], the learner has access to some offline expert demonstrations to start with, and also receives interactive expert demonstration feedback in trajectory segments for subsequent finetuning. Motivated by this practice, we formulate the following problem:

Hybrid Imitation Learning (HyIL): Problem Setup. The learner has access to two complementary sources of expert supervision:

- N_{off} offline expert trajectories $D_{\mathrm{off}} = \left\{(s_{i,h}, a_{i,h})_{h=1}^{H}, i \in [N_{\mathrm{off}}]\right\}$, sampled i.i.d. from rolling out π^E in $\mathcal{M}$;
- A state-wise annotation oracle $\mathcal{O}^{\mathrm{State}}$ that can be queried interactively up to N_{int} times.

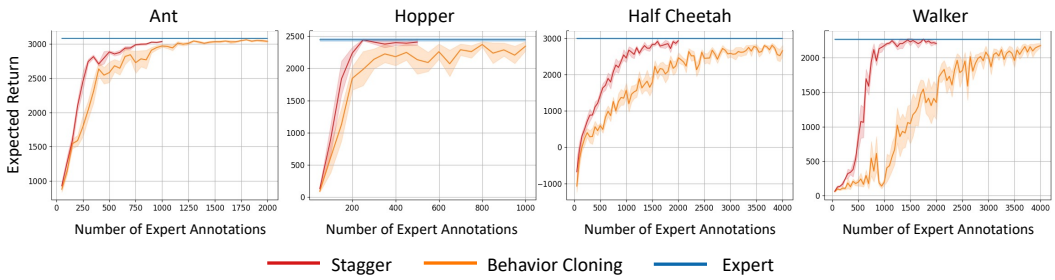

Figure 1: State-wise sample complexity comparison between Behavior Cloning and STAGGER. Shaded areas show the 10th–90th percentile bootstrap confidence intervals [15] over 10 runs. STAGGER matches or exceeds BC with 50% fewer annotations, achieving better state-wise annotation efficiency.

Algorithm 2 WARM-STAGGER: Warm-start STAGGER with offline demonstrations

1: **Input:** MDP $\mathcal{M}$, state-wise expert annotation oracle $\mathcal{O}^{\mathrm{State}}$, stationary policy class $\mathcal{B}$, online learning oracle $\mathbb{A}$, offline expert dataset D_{off} of size N_{off}, online budget N_{int}
2: Initialize $\mathbb{A}$ with policy class $\mathcal{B}_{\mathrm{bc}} := \{\pi \in \mathcal{B} : \pi(s_h) = a_h,\ \forall h \in [H], \forall (s,a)_{1:H} \in D_{\mathrm{off}}\}$.
3: **for** $n = 1, \ldots, N_{\mathrm{int}}$ **do**
4: Query $\mathbb{A}$ and receive π^n.
5: Execute π^n and sample $s^n \sim d^{\pi^n}$. Query $\mathcal{O}^{\mathrm{State}}$ for $a^{*,n} = \pi^{\mathrm{E}}(s^n)$.
6: Update $\mathbb{A}$ with loss function:

$$\ell^n(\pi) := \log\left(\frac{1}{\pi(a^{*,n} \mid s^n)}\right). \tag{2}$$

7: **end for**
8: **Output:** $\hat{\pi}$, a first-step uniform mixture of $\{\pi^1, \ldots, \pi^N\}$.

Each offline (state, action) pair takes a unit cost, and the cost of an interactive query is $C \geq 1$. The total cost budget is therefore $H \cdot N_{\mathrm{off}} + C \cdot N_{\mathrm{int}}$. The goal is to return a policy $\hat{\pi}$ that minimizes its suboptimality relative to the expert policy $J(\pi^E) - J(\hat{\pi})$.

We ask: can we design a HyIL algorithm with provable sample efficiency guarantee? Furthermore, can its performance surpass pure BC and pure interactive IL with the same cost budget?

4.1 WARM-STAGGER: Algorithm and Analysis

We answer the above questions by proposing the WARM-STAGGER algorithm, namely Algorithm 2. It extends STAGGER to incorporate offline expert demonstrations, in that it constructs $\mathcal{B}_{\mathrm{bc}}$, a restricted policy class that contains all policies in $\mathcal{B}$ consistent with all offline expert demonstrations (line 2). It subsequently performs online log-loss optimization on $\mathcal{B}_{\mathrm{bc}}$ over state-action pairs collected online, where the state s^n is obtained by rolling out π^n in the MDP $\mathcal{M}$, and the action $a^{*,n}$ is annotated by the state-wise expert annotation oracle $\mathcal{O}^{\mathrm{State}}$. For analysis, we introduce the following definitions.

Definition 4 (Non-stationary Markovian policies)**.** *A non-stationary Markovian policy $\nu = (\nu_1, \ldots, \nu_H)$ is a collection of H mappings, with each ν_h in $\Delta(\mathcal{A})^{\mathcal{S}}$, where upon rolling out ν, at every step $h \in [H]$, the agent takes action a_h by sampling from $\nu_h(\cdot \mid s_h)$.*

Definition 5 (Step-wise completion of stationary policy class)**.** *For a stationary policy class $\mathcal{B} \subseteq \Delta(\mathcal{A})^{\mathcal{S}}$, define its step-wise completion to be a class of nonstationary Markovian policies:*

$$\tilde{\mathcal{B}} = \{\nu = (\nu_1, \ldots, \nu_H) : \nu_h \in \mathcal{B}, \textit{ for all } h \in [H]\}$$

In words, each $\pi \in \tilde{\mathcal{B}}$ uses a possibly distinct policy π_h from $\mathcal{B}$ to take action at step h. By definition, $\tilde{B} := |\tilde{\mathcal{B}}|$ is at most B^H. An interesting special case is the *non-parameter sharing setting* [49, 47, 46, 17], where the set of possible states visited at different steps are disjoint (see also footnote 1 for examples), and $\mathcal{B}$ are *factorized*, in the sense that parameters associated with the policies to use at different steps are separate. In this case, we have that $\tilde{\mathcal{B}} = \mathcal{B}$ and thus $\tilde{B} = B$.

Theorem 6. *If* WARM-STAGGER *is run with a state-wise expert annotation oracle $\mathcal{O}^{\mathrm{State}}$, an MDP $\mathcal{M}$ where $(\mathcal{M}, \pi^{\mathrm{E}})$ is μ-recoverable, a policy class $\mathcal{B}$ such that deterministic realizability (Assumption 1) holds, and the online learning oracle $\mathbb{A}$ set as the exponential weight algorithm with each-step mixing policies, then it returns $\hat{\pi}$ such that, with probability at least $1-\delta$,*

$$J(\pi^{\mathrm{E}}) - J(\hat{\pi}) \leq O\left(\min\left(\frac{R\log(\tilde{B}/\delta)}{N_{\mathrm{off}}}, \frac{\mu H \log(B_{bc}/\delta)}{N_{\mathrm{int}}}\right)\right), \tag{3}$$

where we recall that $B \leq \tilde{B} \leq B^H$, and $B_{bc} := |\mathcal{B}_{bc}| \leq B$.

Theorem 6 shows that WARM-STAGGER finds a policy with suboptimality guarantee not signficantly worse than BC or STAGGER: first, Behavior Cloning using the offline data has a suboptimality guarantee of $O\left(\frac{R\log(B/\delta)}{N_{\mathrm{off}}}\right)$ (cf. Theorem 2), and WARM-STAGGER's guarantee is worse by at most a factor of H; second, STAGGER without using offline data has a suboptimality of $O\left(\frac{\mu H \log(B/\delta)}{N_{\mathrm{int}}}\right)$ (cf. Theorem 3), which is on par with the second term of Eq. (8). We conjecture that the $\log \tilde{B}$ dependence may be sharpened to $\log B$, and leave this as an interesting open question.

Remark 7. *One may consider another baseline that naively switches between BC and* STAGGER *based on a comparison between their bounds; however, such a baseline needs to know R and μ ahead of time. In practice, we expect* WARM-STAGGER *to perform much better than this baseline, since it seamlessly incorporates both sources of data, and its design does not rely on theoretical bounds that may well be pessimistic.*

4.2 On the Benefit of Hybrid Imitation Learning

Theorem 6 is perhaps best viewed as a fall-back guarantee for WARM-STAGGER: its performance is not much worse than using either of the feedback source alone. In this section, we demonstrate that the benefit of hybrid feedback modalities can go beyond this: we construct an MDP, in which hybrid imitation learning has a significantly better sample efficiency than both offline BC and interactive STAGGER. Specifically, we prove the following theorem:

Theorem 8. *For large enough S, H, A, there exists an episodic MDP $\mathcal{M}$ with S states, A actions, and horizon H, and expert policy π^{E} such that:*

- *With $\Omega(S)$ offline expert trajectories for BC, the learned policy is $\Omega(H)$-suboptimal;*
- *With $\Omega(HS)$ interactive expert annotations for* STAGGER*, the learned policy is $\Omega(H)$-suboptimal;*
- *With $\tilde{O}(S/H)$ offline trajectories and $O(1)$ expert interactions,* WARM-STAGGER *learns a policy $\hat{\pi}$ such that $J(\hat{\pi}) = J(\pi^E)$.*

Theorem 8 suggests that when $HS \gg \max(1, C)$ and $C \gg \frac{1}{H}$, WARM-STAGGER achieves expert-level performance with significantly lower cost than two baselines. To see this, observe that WARM-STAGGER has a total cost of $O(S + C)$, which is much smaller than $\Omega(HS)$ by BC, and $\Omega(HSC)$ by STAGGER.

The MDP construction and simulation results. We now sketch our construction of MDP $\mathcal{M}$. $\mathcal{M}$ has an episode length $H = \Omega(\log(S))$ and action space of size greater than $10H$. For each state, one of the actions is taken by the expert; the rest are "wrong" actions. We illustrate $\mathcal{M}$'s state space on the left of Figure 2; specifically, it consists of the following subsets:

- Unrecoverable state $\mathbf{B} := \{\mathbf{b}\}$: a special absorbing state that is unrecoverable by any action (dead).
- Expert ideal states $\mathbf{E}$, where $|\mathbf{E}| = N_0$: this can model for example, an autonomous driving agent driving stably on the edge of a cliff [51], where any incorrect action transitions the agent to the unrecoverable state $\mathbf{b}$ (e.g., car falling off the cliff). Taking the expert action keeps the agent in $\mathbf{E}$ with high probability $(1-\beta)$, and with a small probability β, moves the agent to $\mathbf{E}'$ (e.g., a safe slope).
- Expert recoverable states $\mathbf{E}'$: this models the agent driving outside the edge of the cliff in a safe slope. When in $\mathbf{E}'$, taking the expert action allows the agent to return to a uniformly

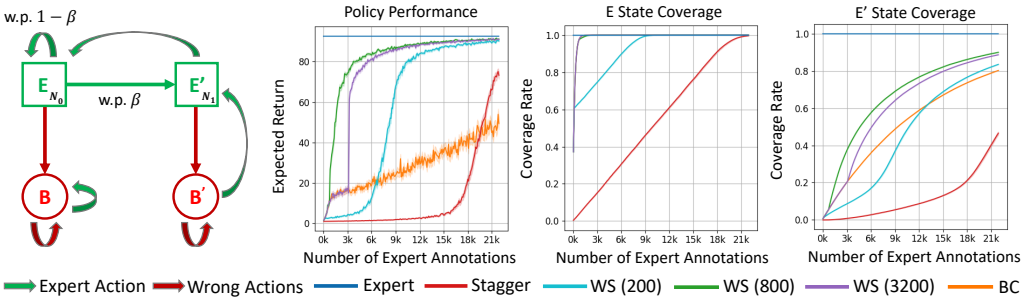

Figure 2: MDP construction and simulation results of algorithms with rewards assigned only in $\mathbf{E}$. We evaluate WARM-STAGGER (WS) with 200, 800, 3200 offline (state, expert action) pairs. All methods are evaluated under equal total annotation cost with $C = 1$. With 800 offline (state, expert action) pairs, WS significantly improves the sample efficiency over the baselines and explores $\mathbf{E}'$ more effectively.

sampled state in $\mathbf{E}$. Taking a wrong action from $\mathbf{E}'$ leads to reaching a recoverable state $\mathbf{b}'$ (e.g., rest area).

- Recoverable state $\mathbf{B}' = \{\mathbf{b}'\}$: Not knowing how to act in $\mathbf{b}'$ will result in the agent getting trapped in $\mathbf{b}'$ for the episode.

We now briefly justify each algorithm's learning performance as stated in Theorem 8. First, BC only observes expert actions in $\mathbf{E}$ and $\mathbf{E}'$, but never in $\mathbf{b}'$. As a result, near-expert performance at test time requires high coverage over $\mathbf{E}'$; otherwise, BC's trained policy will likely incur compounding errors and get trapped in $\mathbf{b}'$. Second, STAGGER suffers from a cold-start problem: early policies fail to explore $\mathbf{E}$ efficiently, and incorrect actions can cause transitions into $\mathbf{b}$. Consequently, coverage over $\mathbf{E}$ grows slowly, and the policy may still fail on unannotated states in $\mathbf{E}$ even with $\Omega(HS)$ queries. Lastly, WARM-STAGGER benefits from offline data that fully covers $\mathbf{E}$, and uses a small number of interactions to visit $\mathbf{b}'$ and query the expert, avoiding costly exploration in $\mathbf{E}'$ while matching expert performance.

We also conduct a simulation of the aforementioned three algorithms in a variant of the above MDP with $N_0 = 200$, $N_1 = 1000$, $H = 100$, and $\beta = 0.08$, using another reward function that assigns a reward of 1 only when the agent visits the states in $\mathbf{E}$. Here, we let the online learning oracle $\mathbb{A}$ optimize 0-1 loss on the data seen so far, which is equivalent to minimizing log loss under a class of deterministic policies and discrete actions. Figure 2 shows return and state coverage as functions of the number of expert annotations, averaged over 200 runs.

We observe that: (1) BC exhibits slow improvement, as $\mathbf{b}'$ remains unseen (and thus unannotated) throughout training, resulting in poor performance even with substantial coverage (e.g., 80%) over $\mathbf{E}'$; (2) STAGGER is sample-inefficient due to slow exploration over states in $\mathbf{E}$, consistent with the cold-start intuition; (3) WARM-STAGGER (WS), when initialized with limited (e.g., 200) offline (state, expert action) pairs, still needs to explore $\mathbf{E}$ first before it can safely reach $\mathbf{E}'$ without failure; and (4) WARM-STAGGER with sufficient offline coverage on $\mathbf{E}$ (e.g., initialized with 3200 offline (state, expert action) pairs) directly benefits from exploring $\mathbf{b}'$ with immediate performance gain, and enables safe and even faster exploration than the expert in $\mathbf{E}'$.

4.3 Hybrid IL on Continuous Control Benchmarks

Following our earlier MuJoCo-based comparison of Behavior Cloning and STAGGER, we now evaluate WARM-STAGGER (WS) on the same benchmarks. This experiment aims to answer: Does WS reduce total annotation cost compared to the baselines?

Based on the observation in Figure 1, we assign 400 total state-wise annotations for Hopper and Ant, and 1200 for HalfCheetah and Walker2D. For WARM-STAGGER, we allocate 1/8, 1/4, or 1/2 of the total annotations to offline data, with the remainder used for interactive queries. For a fair comparison, all methods are evaluated under the same total annotation cost, with $C = 1$ or $C = 2$. This makes the baselines stronger, as they have full cost budget assigned to a single source.

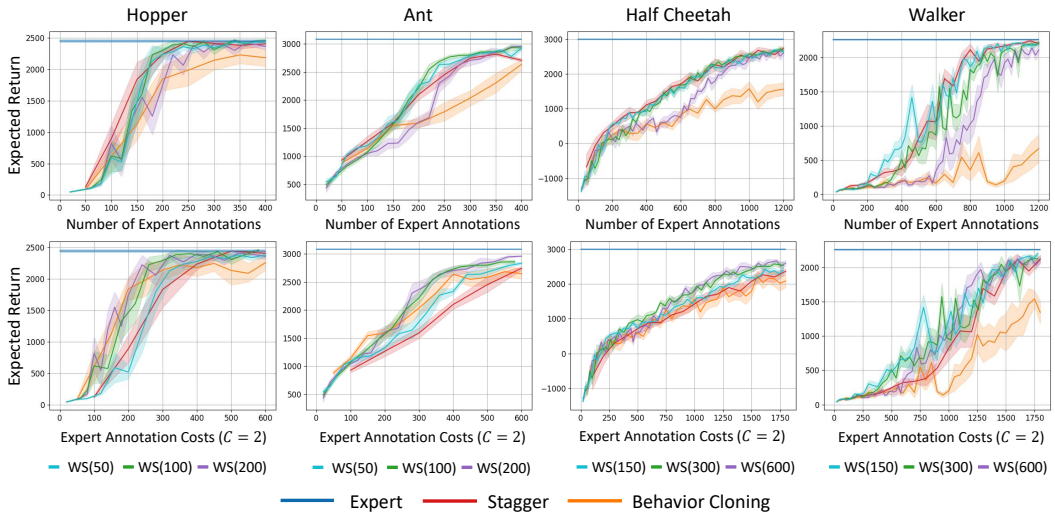

Figure 3: Sample and cost efficiency on MuJoCo tasks. The top row shows expected return vs. number of annotations ($C = 1$); the bottom row shows performance in a cost-aware setting ($C = 2$). WARM-STAGGER (WS) is initialized with 1/8, 1/4, or 1/2 of the total annotation budget as offline demonstrations. Specifically, WS(n) refers to WS with offline expert trajectory demonstrations of total length n. For a good range of n's, WS(n) matches STAGGER in sample efficiency and outperforms the baselines when $C = 2$.

In terms of the number of state-wise annotations ($C = 1$), the results align with our theoretical findings: WS performs not significantly worse than BC or STAGGER, regardless of the offline dataset size. WS still achieves performance competitive with STAGGER, and even outperforms it on Ant when $C = 1$. Furthermore, as shown by the purple curves, WS with appropriate offline sample size has preferable performance over 4 tasks when $C = 2$, highlighting its utility in cost-aware regimes. These results confirm that WARM-STAGGER reduces total annotation cost for moderate C.

5 Related Work

Imitation Learning with offline demonstrations, pioneered in autonomous driving [45], was solved by offline, state-wise supervised learning in early works [49, 70] and named Behavior Cloning (BC). A recent analysis by [17] employs trajectory-wise Hellinger distance to tighten the dependence of BC on the horizon at the trajectory level, although its sample complexity measured per state still grows quadratically with the horizon in the worst case. This shortcoming, often termed covariate shift or compounding error [45], arises when the learned policy's imperfect imitation drives the learner to unseen states, resulting in a cascading sequence of mistakes. From a data collection perspective, this can be mitigated by noise-injection approaches such as [30, 26]. By leveraging additional environment interactions, generative-adversarial IL methods [19, 66, 25, 64] frame learning as a two-player game, and aims to find a policy that matches expert's state-action visitation distributions. This setting is also known as "apprenticeship learning using inverse reinforcement learning" in earlier works [1, 69]. Quantitative comparisons with these methods are beyond our scope, as they rely on extensive interactions with the MDP and access to a class of discriminator functions, while we focus on understanding the utility of state-wise interactive annotations. This line of works also include recent work of [48], who introduce "Hybrid Inverse Reinforcement Learning", which leverages hybrid reinforcement learning [61] to accelerate its inner loop of policy search; different from theirs, our "hybrid" setting focus on utilizing heterogeneous data modalities. Recent offline imitation learning approaches [10, 75] do not require MDP access but still require access to offline datasets possibly collected by non-expert policies, either with broad expert coverage or a large transition buffer. Our work assumes that interacting with the environment does not incur costs; we leave a detailed analysis that takes into account environment interaction cost as future work.

Imitation Learning with interactive demonstrations, first proposed by [49], allows the expert to provide corrective feedback to the learner's action retroactively. Assuming low costs of expert recovery from mistakes (μ-recoverability; recall Section 2), DAgger [51], and following works [28, 50, 67, 11, 12, 46] outperform traditional BC both theoretically and empirically. However, this efficiency demands substantial annotation effort [37]. Although DAgger [51] and some subsequent works [67, 46, 68, 17] popularized the practice of annotating full trajectories, there has also been growing interest in state-wise annotations [40, 33, 54, 34], which appeared as early as [49, 22]. In fact, practical applications of DAgger often adopt partial trajectory annotation in expert-in-the-loop [37, 62, 36] designs, as seen in [77, 27, 20, 73], where issues such as inconsistencies caused by retroactive relabeling [29] can be mitigated. These methods often leverage human- or machine-gated expert interventions to ensure safety during data collection [78, 38], provide more targeted feedback [39, 13], and enable learning on the fly [59]. The use of selective state-wise queries aligns with our goal of promoting interactive imitation learning with efficient supervision and provable sample efficiency. We regard our contribution as providing a starting point for understanding this increasingly popular paradigm of partial trajectory annotations.

Utilizing Offline Data for Interactive Learning. Many practical deployments of interactive learning systems do not start from tabula rasa; instead, prior knowledge of various forms is oftentimes available. For example, combining offline data and interactive feedback has recently gained much popularity in applications such as training large language models to follow human instructions [16, 44], and bandit machine translation [41]. Many recent theoretical works in reinforcement learning try to quantify the computational and statistical benefit of combining offline and online feedback: for example, [32, 71] show provable reduction of sample complexity using hybrid reinforcement learning, using novel notions of partial coverage; [61] shows that under some structural assumptions on the MDP, hybrid RL can bypass computational barriers in online RL [24]. Many works also quantify the benefit of utilizing additional offline data sources in the contextual bandit domain; for example, [42, 58, 76] study warm-starting contextual bandit learning using offline bandit logged data or supervised learning examples. While some variants of DAgger [78, 20] also operate in a hybrid setting, our work focuses on a fundamental formulation that explicitly accounts for the cost asymmetry between offline and interactive annotations [56].

6 Conclusion

We revisit imitation learning from the perspective of state-wise annotations. We show via the STAGGER algorithm that, interaction with the demonstrating expert, with its cost properly measured, can enable provable cost efficiency gains over Behavior Cloning. We also propose WARM-STAGGER that combines the benefits of offline data and interactive feedback. Our theoretical analysis shows that such a hybrid method can strictly outperform both pure offline and pure interactive baselines under realistic cost models. Empirical results on our synthetic MDP support our theoretical findings, while MuJoCo experiments demonstrate the practical viability and competitive performance of our methods on continuous control tasks. In Appendix E, we also show that a trajectory-wise annotation variant of DAgger can match the sample complexity of log-loss Behavior Cloning without recoverability assumptions, with additional experiments (Appendix G.3).

Limitations: Our design of imitation learning algorithm only aims at closing the gap between the performance of the expert and the trained policy; thus, the performance of our learned policy is bottlenecked by the expert's performance. In this respect, designing imitation learners that output policies surpassing expert performance is an important direction.

Our theory provide sample complexity guarantees for the discrete-action setting with deterministic and realizable expert. When such assumptions are relaxed, additional challenges arise [60]. In this respect, there remains a gap between our theoretical analysis and our MuJoCo experiment results. In future work, we are interested in conducting additional experiments on discrete-action control problems (e.g., Atari) as well as language model distillation tasks.

Acknowledgments: We thank the anonymous NeurIPS reviewers for their helpful feedback, which significantly improved the presentation of the paper. We thank Kianté Brantley for helpful discussions on state-wise annotations. We thank National Science Foundation IIS-2440266 (CAREER) for research support.

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[2] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32:96, 2019.

[3] Philip Amortila, Nan Jiang, Dhruv Madeka, and Dean P Foster. A few expert queries suffices for sample-efficient rl with resets and linear value approximation. *Advances in Neural Information Processing Systems*, 35:29637–29648, 2022.

[4] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert systems with applications*, 165:113816, 2021.

[5] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

[6] 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.

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

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

[9] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[10] 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.

[11] 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.

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

[13] 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.

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

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

[16] Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *Transactions on Machine Learning Research*.

[17] Dylan J Foster, Adam Block, and Dipendra Misra. Is behavior cloning all you need? understanding horizon in imitation learning. *arXiv preprint arXiv:2407.15007*, 2024.

[18] Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*.

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

[20] 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.

[21] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International conference on machine learning*, pages 1453–1461. PMLR, 2013.

[22] Kshitij Judah, Alan P Fern, Thomas G Dietterich, and Prasad Tadepalli. Active imitation learning: Formal and practical reductions to iid learning. *Journal of Machine Learning Research*, 15(120):4105–4143, 2014.

[23] Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, 2002.

[24] Daniel Kane, Sihan Liu, Shachar Lovett, and Gaurav Mahajan. Computational-statistical gap in reinforcement learning. In *Conference on Learning Theory*, pages 1282–1302. PMLR, 2022.

[25] 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.

[26] Liyiming Ke, Jingqiang Wang, Tapomayukh Bhattacharjee, Byron Boots, and Siddhartha Srinivasa. Grasping with chopsticks: Combating covariate shift in model-free imitation learning for fine manipulation. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 6185–6191. IEEE, 2021.

[27] 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.

[28] Beomjoon Kim, Amir-massoud Farahmand, Joelle Pineau, and Doina Precup. Learning from limited demonstrations. *Advances in Neural Information Processing Systems*, 26, 2013.

[29] Michael Laskey, Caleb Chuck, Jonathan Lee, Jeffrey Mahler, Sanjay Krishnan, Kevin Jamieson, Anca Dragan, and Ken Goldberg. Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 358–365. IEEE, 2017.

[30] 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.

[31] Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudík, Yisong Yue, and Hal Daumé III. Hierarchical imitation and reinforcement learning. In *International conference on machine learning*, pages 2917–2926. PMLR, 2018.

[32] Gen Li, Wenhao Zhan, Jason D Lee, Yuejie Chi, and Yuxin Chen. Reward-agnostic fine-tuning: Provable statistical benefits of hybrid reinforcement learning. *Advances in Neural Information Processing Systems*, 36:55582–55615, 2023.

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

[34] Yichen Li and Chicheng Zhang. Agnostic interactive imitation learning: New theory and practical algorithms, 2023.

[35] Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. *arXiv preprint arXiv:2009.07253*, 2020.

[36] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *The International Journal of Robotics Research*, page 02783649241273901, 2022.

[37] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020.

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

[39] 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.

[40] Khanh Nguyen and Hal Daumé III. Active imitation learning from multiple non-deterministic teachers: Formulation, challenges, and algorithms. *arXiv preprint arXiv:2006.07777*, 2020.

[41] Khanh Nguyen, Hal Daumé III, and Jordan Boyd-Graber. Reinforcement learning for bandit neural machine translation with simulated human feedback. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1464–1474, 2017.

[42] Bastian Oetomo, R Malinga Perera, Renata Borovica-Gajic, and Benjamin IP Rubinstein. Cutting to the chase with warm-start contextual bandits. *Knowledge and Information Systems*, 65(9):3533–3565, 2023.

[43] 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.

[44] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

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

[46] 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.

[47] Nived Rajaraman, Lin Yang, Jiantao Jiao, and Kannan Ramchandran. Toward the fundamental limits of imitation learning. *Advances in Neural Information Processing Systems*, 33:2914–2924, 2020.

[48] Juntao Ren, Gokul Swamy, Steven Wu, Drew Bagnell, and Sanjiban Choudhury. Hybrid inverse reinforcement learning. In *International Conference on Machine Learning*, pages 42428–42448. PMLR, 2024.

[49] 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.

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

[51] 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.

[52] 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.

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

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

[55] Ayush Sekhari, Karthik Sridharan, Wen Sun, and Runzhe Wu. Selective sampling and imitation learning via online regression. *Advances in Neural Information Processing Systems*, 36, 2024.

[56] Burr Settles, Mark Craven, and Lewis Friedland. Active learning with real annotation costs. In *Proceedings of the NIPS workshop on cost-sensitive learning*, volume 1. Vancouver, CA:, 2008.

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

[58] Nihal Sharma, Soumya Basu, Karthikeyan Shanmugam, and Sanjay Shakkottai. Warm starting bandits with side information from confounded data. *arXiv preprint arXiv:2002.08405*, 2020.

[59] Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at your robot: Improving on-the-fly from language corrections. *arXiv preprint arXiv:2403.12910*, 2024.

[60] Max Simchowitz, Daniel Pfrommer, and Ali Jadbabaie. The pitfalls of imitation learning when actions are continuous. *arXiv preprint arXiv:2503.09722*, 2025.

[61] Yuda Song, Yifei Zhou, Ayush Sekhari, Drew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. In *The Eleventh International Conference on Learning Representations*.

[62] 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.

[63] Jonathan Spencer, Sanjiban Choudhury, Matthew Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Siddhartha Srinivasa. Learning from interventions: Human-robot interaction as both explicit and implicit feedback. In *16th robotics: science and systems, RSS 2020*. MIT Press Journals, 2020.

[64] 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.

[65] Wen Sun, J Andrew Bagnell, and Byron Boots. Truncated horizon policy search: Combining reinforcement learning & imitation learning. *arXiv preprint arXiv:1805.11240*, 2018.

[66] 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.

[67] 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.

[68] 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.

[69] Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. *Advances in neural information processing systems*, 20, 2007.

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

[71] Kevin Tan, Wei Fan, and Yuting Wei. Hybrid reinforcement learning breaks sample size barriers in linear mdps. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[72] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[73] Mitchell Torok, Mohammad Deghat, and Yang Song. Greedy-dagger-a student rollout efficient imitation learning algorithm. *IEEE Robotics and Automation Letters*, 2025.

[74] Yuanyu Wan, Wei-Wei Tu, and Lijun Zhang. Online strongly convex optimization with unknown delays. *Machine Learning*, 111(3):871–893, 2022.

[75] Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. When demonstrations meet generative world models: A maximum likelihood framework for offline inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 36:65531–65565, 2023.

[76] Chicheng Zhang, Alekh Agarwal, Hal Daumé Iii, John Langford, and Sahand Negahban. Warm-starting contextual bandits: Robustly combining supervised and bandit feedback. In *International Conference on Machine Learning*, pages 7335–7344. PMLR, 2019.

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

[78] Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end simulated driving. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[79] Zhouqiao Zhao, Xishun Liao, Amr Abdelraouf, Kyungtae Han, Rohit Gupta, Matthew J Barth, and Guoyuan Wu. Real-time learning of driving gap preference for personalized adaptive cruise control. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4675–4682. IEEE, 2023.

[80] Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. Modeling interaction via the principle of maximum causal entropy. 2010.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's contributions—proposing STAGGER and WARM-STAGGER, establishing their sample efficiency, and validating them empirically. These claims align with the paper's content and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The paper discusses limitations such as the assumption of a realizable and deterministic expert, and that environment interaction cost is not modeled. These are noted in the theoretical setup and discussed as directions for future work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Each theorem explicitly states its assumptions, such as realizability and recoverability. Full, self-contained proofs are provided in the appendix, including intermediate lemmas and detailed derivations, to the best of our knowledge.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: The paper includes all necessary details to reproduce the main experimental results, including model architecture, training hyperparameters, evaluation protocols, and data collection methods. Additional implementation details and results are provided in Appendix G.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper includes a link in the appendix to a public GitHub repository containing the code and instructions necessary to reproduce the main experimental results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies training and evaluation protocols, including model architecture, learning rate, batch size, optimizer, number of updates, and trajectory length. Additional details and implementation-specific settings are provided in Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports error bars using bootstrap confidence bounds, computed over 10 independent runs, and clearly states the methodology in the caption of each relevant figure.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Details on compute resources used for experiments, including GPU type, memory configuration, and experiment running time, are provided in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research adheres to the NeurIPS Code of Ethics. It involves standard simulation environments and pretrained MLP expert policy annotations, without the use of sensitive data, human subjects, or deployments with societal implications.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper presents foundational theoretical work in imitation learning, aimed at improving sample efficiency through state-wise annotation and hybrid learning algorithms. To the best of our knowledge, the work poses no direct negative societal impact. Its potential applications, such as in robotics or autonomous systems, are mentioned to motivate the study, but the contributions themselves are purely algorithmic and theoretical.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release pretrained models, large-scale datasets, or tools with high risk of misuse. The work is theoretical and algorithmic in nature, focusing on sample efficiency in imitation learning, and thus does not raise direct concerns requiring safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets used in this work, including MuJoCo environments and pretrained experts, are properly cited (e.g., [72, 52]). MuJoCo is used under its standard academic license, and the pretrained models are referenced with proper attribution. No proprietary data or code was used without credit.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce new simulation scripts and code for our proposed algorithms (STAGGER and WARM-STAGGER), which are documented and included via a GitHub link in the appendix. Instructions for running experiments and reproducing results are also provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve any crowdsourcing or research with human subjects. All experiments are conducted in simulated environments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve human subjects or participant-based studies; all experiments are performed in simulation, so IRB approval is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Large language models (LLMs) were only used for grammar checking and editing. They were not involved in the development or evaluation of the core methodology, and did not influence the scientific contributions of the work.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Additional Related Work

First-step mixing and each-step mixing policies. The emergence of first-step mixing policies originated from technical considerations. In may interactive IL methods [51, 50], the returned policy was not a uniform first-step mixture but rather the best policy selected through validation. However, performing such validation in an interactive setting often requires additional expert annotations. Subsequent works [46, 33, 34, 17] circumvented the need for validation by employing a uniform first-step mixture of policies across learning rounds, thereby directly translating online regret guarantees into performance differences. Our TRAGGER algorithm (Algorithm 3 in Appendix E) also employs a first-step mixing policy at each iteration, and has state-wise sample complexity on par with behavior cloning.

On the other hand, each-step mixing between the learned policy across rounds and the expert policy has been a prevalent strategy in interactive IL approaches [14, 49, 51, 50]. For each-step mixture policies, [33] was the first to explicitly distinguish this approach from first-step mixing. In other works [46, 17], each-step mixing can be interpreted as learning H separate mixture policies, one for each step within an episode.

Alternative algorithm designs and practical implementations. Though this work follows [17] and focuses on log loss, we believe our $O(1/N)$ rate is not exclusive to log loss. Despite requiring an additional supervision oracle, [31] suggests that trajectory-wise annotation complexity similar to Theorem 3 (and Theorem 31) can be achieved using Halving [57] and 0-1 loss.

From an algorithmic perspective, we explored trajectory-wise annotation with first-step mixing (Algorithm 3 in Appendix E) and state-wise annotation with each-step mixing (Algorithm 1). For trajectory-wise annotation with each-step mixing, naively learning a parameter-sharing policy may encounter a batch-summed log loss, introducing an (undesirable) additional H factor to the sample complexity [21, 74]. Analyzing state-wise annotation with first-step mixing remains an open question.

For practical implementations, it is worth noting that even with oracle-efficient implementations (e.g. [33, 34]), interactive IL may require multiple supervised learning oracle calls per iteration. In contrast, offline IL requires only a single oracle call to obtain the returned policy, which provides a clear computational advantage. We also note that real-world experts can be suboptimal; in some applications it may be preferable to combine imitation and reinforcement learning signals (e.g., [50, 65, 3]).

Information-theoretic lower bounds for interactive imitation learning. A line of works [47, 46, 17] provides lower bounds for the sample complexity of imitation learning under the realizable setting and considers μ-recoverability. [46] is the first to demonstrate a gap between the lower bounds of offline IL and interactive IL in trajectory-wise annotation, focusing on the tabular and non-parameter-sharing setting. [17] establishes a $\Omega\left(\frac{H}{\epsilon}\right)$ sample-complexity lower bound for trajectory-wise annotation in the parameter-sharing setting.

We observe that the proof of [17, Theorem 2.2] also implicitly implies a $\Omega(\frac{H}{\epsilon})$ sample complexity lower bound for the state-wise annotation setting. Their proof relies on an MDP consisting only of self-absorbing states, where annotating a full trajectory gives the same amount of information as annotating a single state. In that MDP (which is 1-recoverable), Algorithm 1 achieves $\tilde{O}(\frac{H\log(B)}{\epsilon})$ state-wise sample complexity, which does not contradict this lower bound. Nonetheless, obtaining lower bounds for state-wise sample complexity for general MDPs, policy classes, and general recoverability constants remains an open question.

B Proof for STAGGER

We first present two useful distance measures for pair of policies.

Definition 9 (Trajectory-wise L_1-divergence)**.** *For a pair of Markovian policies π and π', define their trajectory-wise L_1-divergence as*

$$\lambda(\pi \parallel \pi') := \mathbb{E}^{\pi}\mathbb{E}_{a'_1\sim\pi'(\cdot|s_1),\ldots,a'_H\sim\pi'(\cdot|s_H)}\left[\sum_{h=1}^{H}\mathbb{I}(a_h \neq a'_h)\right].$$

$\lambda(\pi \parallel \pi')$ is the expected total number of actions taken by π' that deviates from actions in trajectories induced by π. Note that $\lambda(\cdot||\cdot)$ is asymmetric, while the same concept is applied in offline and interactive IL [49, 51] with different guarantees for $\lambda(\pi \parallel \pi^{\mathrm{E}})$ and $\lambda(\pi^{\mathrm{E}} \parallel \pi)$ (see Lemma 39).

Definition 10 (State-wise Hellinger distance)**.** *For a pair of policies π and π', define their state-wise Hellinger distance as $\mathbb{E}_{s\sim d^{\pi}}\left[D^2_H(\pi(\cdot \mid s), \pi'(\cdot \mid s))\right]$.*

State-wise Hellinger distance represents the expected Hellinger distance between the action distribution of π and π' on $s \sim d^{\pi}$. One notable feature here is that the distance is evaluated between $\pi(\cdot \mid s)$ and $\pi'(\cdot \mid s)$, unrelated to the original action a taken by by π when visiting s. By Lemma 38, state-wise Hellinger distance with the expert policy is a constant factor equivalent to trajectory-wise L_1-divergence in the deterministic realizable expert setting.

In the following lemma, we show that the performance difference between the policy $\hat{\pi}$ returned by STAGGER (Algorithm 1) and the expert policy π^{E} can be bounded by the online state-wise Hellinger estimation error:

$$\mathrm{OnEst}^{\mathrm{State}}_N := \sum_{n=1}^{N} \mathbb{E}_{s\sim d^{\pi^n}}\left[D^2_{\mathrm{H}}(\pi^n(\cdot \mid s), \pi^{\mathrm{E}}(\cdot \mid s))\right],$$

where $\pi^n(\cdot \mid s)$ and $\pi^E(\cdot \mid s)$ denote the action distributions produced by the policies π^n and π^E at state s.

We are ready to prove the state-wise annotation complexity of Algorithm 1:

Lemma 11. *For any MDP $\mathcal{M}$, deterministic expert π^{E}, and sequence of policies $\{\pi^n\}_{n=1}^N$, then $\hat{\pi}$, the first-step uniform mixture of $\{\pi^1, \dots, \pi^N\}$ satisfies:*

$$J(\pi^{\mathrm{E}}) - J(\hat{\pi}) \le \mu H \cdot \frac{\mathrm{OnEst}^{\mathrm{State}}_N}{N}.$$

Proof. By Lemma 39, under the assumption of recoverability, the performance difference between $\hat{\pi}$ and the expert is bounded by

$$J(\pi^{\mathrm{E}}) - J(\hat{\pi}) \le \mu \cdot \lambda(\hat{\pi} \parallel \pi^{\mathrm{E}}),$$

where we recall the notation that

$$\lambda(\pi \parallel \pi^{\mathrm{E}}) = \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} \mathbb{I}(a_h \neq \pi^{\mathrm{E}}(s_h))\right] = \frac{1}{2}\sum_{h=1}^{H} \mathbb{E}^{\pi}\|\pi(\cdot \mid s_h) - \pi^{\mathrm{E}}(\cdot \mid s_h)\|_1.$$

The proof follows by upper-bounding $\sum_{n=1}^{N} \lambda(\pi^n \parallel \pi^{\mathrm{E}})$ by $H \cdot \mathrm{OnEst}^{\mathrm{State}}_N$. To this end, it suffices to show that for any stationary policy π,

$$H \cdot \mathbb{E}_{s\sim d^{\pi}}\left[D^2_H(\pi(\cdot \mid s), \pi^{\mathrm{E}}(\cdot \mid s))\right] \ge \frac{1}{2}\sum_{h=1}^{H} \mathbb{E}^{\pi}\|\pi(\cdot \mid s) - \pi^{\mathrm{E}}(\cdot \mid s)\|_1.$$

Observe that $H \cdot \mathbb{E}_{s\sim d^{\pi}}\left[D^2_H(\pi(\cdot \mid s), \pi^{\mathrm{E}}(\cdot \mid s))\right] = \sum_{h=1}^{H} \mathbb{E}^{\pi}\left[D^2_H(\pi(\cdot \mid s_h), \pi^{\mathrm{E}}(\cdot \mid s_h))\right]$, we conclude the proof by applying Lemma 38 with $p = \pi(\cdot \mid s_h)$ and $q = \pi^{\mathrm{E}}(\cdot \mid s_h)$, which gives

$$D^2_{\mathrm{H}}(\pi(\cdot \mid s_h), \pi^{\mathrm{E}}(\cdot \mid s_h)) \ge \frac{1}{2}\|\pi(\cdot \mid s_h) - \pi^{\mathrm{E}}(\cdot \mid s_h)\|_1.$$

□

Theorem 12 (Theorem 3 Restated)**.** *If* STAGGER *(Algorithm 1) is run with a state-wise expert annotation oracle $\mathcal{O}^{\mathrm{State}}$, an MDP $\mathcal{M}$ where $(\mathcal{M}, \pi^{\mathrm{E}})$ is μ-recoverable, a policy class $\mathcal{B}$ such that deterministic realizability (Assumption 1) holds, and the online learning oracle $\mathbb{A}$ set as the exponential weight algorithm, then it returns $\hat{\pi}$ such that, with probability at least $1-\delta$,*

$$\mathrm{OnEst}^{\mathrm{State}}_{N_{\mathrm{int}}} \le \log(B) + 2\log(1/\delta),$$

and furthermore, the returned $\hat{\pi}$ satisfies

$$J(\hat{\pi}) - J(\pi^{\mathrm{E}}) \le \mu H \frac{\log(B) + 2\log(1/\delta)}{N_{\mathrm{int}}}.$$

Proof. Recall the each-step mixing in Definition 1: since $\bar{\pi}_u$ is a each-step mixing policy, we have that $\forall h \in [H], s \in \mathcal{S}, \bar{\pi}_u(a|s) = \sum_{\pi \in \mathcal{B}} u(\pi)\pi(a|s)$.

The loss function at each round that passed through online learning oracle $\mathbb{A}$, evaluated at $\bar{\pi}_u \in \bar{\Pi}_{\mathcal{B}}$, $\ell^n(\bar{\pi}_u)$, is of the form

$$\ell^n(\bar{\pi}_u) = \log\left(\frac{1}{\bar{\pi}_u(a^{n,*}|s^n)}\right) = \log\left(\frac{1}{\sum_{\pi \in \mathcal{B}} u(\pi)\pi(a^{n,*}|s^n)}\right),$$

which is 1-exp-concave with respect to u. Thus, implementing $\mathbb{A}$ using the exponential weights algorithm (Proposition 41) achieves:

$$\sum_{n=1}^{N_{\mathrm{int}}} \log(1/\pi^n(a^{*,n} \mid s^n)) \le \sum_{n=1}^{N_{\mathrm{int}}} \log(1/\pi^{\mathrm{E}}(a^{*,n} \mid s^n)) + \log(B) = \log(B).$$

Then, Lemma 42, a standard online-to-batch conversion argument with $x^n = (s^n, h^n)$, $y^n = a^{*,n}$, $g_* = \pi^{\mathrm{E}}$, and $\mathcal{H}^n = \{o^{n'}\}_{n'=1}^n$, where $o^n = (s^n, a^n, a^{*,n})$, implies that with probability at least $1-\delta$,

$$\mathrm{OnEst}^{\mathrm{State}}_{N_{\mathrm{int}}} = \sum_{n=1}^{N_{\mathrm{int}}} \mathbb{E}_{s^n \sim d^{\pi^n}}\left[D^2_{\mathrm{H}}(\pi^n(\cdot \mid s^n), \pi^{\mathrm{E}}(\cdot \mid s^n))\right] \le \log(B) + 2\log(1/\delta).$$

The second part of the theorem follows by applying Lemma 11. □

C Proof for WARM-STAGGER

In this section, we analyze the guarantees of WARM-STAGGER under the realizable and deterministic expert assumption. We show that all intermediate policies, as well as the final returned mixture policy, induces distributions over trajectories that enjoy small Hellinger distance to the expert's trajectory distribution, due to their agreement on the offline dataset. Our analysis builds on generalization guarantees of the maximum likelihood estimator (MLE).

C.1 Preliminaries: First-step mixing and causally conditioned probabilities

Our main analysis leverages the fact that the each-step mixing policies π^n's maintained by WARM-STAGGER can be viewed as first-step mixtures of policies in $\tilde{\mathcal{B}}$ (recall Definition 5) – this subsection is dedicated to prove this result (Lemma 16). We first recall the definition of first-step mixing of policies, adapted to our notations:

Definition 13 (First-step mixing, e.g. [69])**.** *Given a distribution u over a set of (possibly nonstationary) policies Π, denote π_u as the first-step mixing policy induced by u: when rolling out π_u, first sample policy $\pi \sim u$, and follow π for the entire episode.* [4]

Importantly, even when the policies in Π are all Markovian, π_u may no longer be a Markovian policy: for example, the random draw of action a_2 does not only depend on the Markovian state s_2 but also the random policy π drawn, which in turn may correlate with a_1.

To simplify the notations in our development below, following [80], we define causally conditional probabilities, when the agent uses Markovian policies or their first-step mixing:

Definition 14. *Given an MDP $\mathcal{M}$, the causally conditional probability of state sequence $s_{1:H}$ given action sequence $a_{1:H-1}$, is defined as:*

$$\mathbb{P}^{\mathcal{M}}(s_{1:H} \parallel a_{1:H-1}) = \rho(s_1) \prod_{h=1}^{H-1} P(s_{h+1} \mid s_h, a_h).$$

Given a (first-step mixture) of Markovian policy, its causally conditional probability of state sequence $a_{1:H}$ given action sequence $s_{1:H}$ is defined as:

[4] Note that we use a notation π_u different from every-step mixing policy notation $\bar{\pi}_u$.

- *For Markovian policy* $\pi = \pi_{1:H}$, $\pi(a_{1:H} \parallel s_{1:H}) := \prod_{h=1}^{H} \pi_h(a_h|s_h)$.
- *For first-step mixing of Markovian policies* π_u, $\pi_u(\cdot \parallel s_{1:H}) := \sum_{\pi \in \mathcal{B}} u(\pi)\pi(\cdot \parallel s_{1:H})$.

Note that $\pi(\cdot\|s_{1:H})$ and $\pi_u(\cdot\|s_{1:H})$ are valid probability distributions (e.g., $\sum_{a_{1:H}} \pi(a_{1:H}\|s_{1:H}) = 1$), however, the use of '$\|$' highlights its distinction from standard conditioning. For example, when executing Markovian policy π, the conditional probability of the actions given the states $\mathbb{P}^{\pi}(a_{1:H}|s_{1:H}) \propto \prod_{h=1}^{H} \pi_h(a_h|s_h) \prod_{h=1}^{H-1} P(s_{h+1} \mid s_h, a_h)$, which is clearly different from its causally conditional counterpart.

We have the following (perhaps folklore) lemma for causally conditional probability (e.g. [80]).

Lemma 15. *For any Markovian policy* π*,*

$$\mathbb{P}^{\pi}(s_{1:H}, a_{1:H}) = \mathbb{P}^{\mathcal{M}}(s_{1:H} \parallel a_{1:H-1}) \cdot \pi(a_{1:H} \parallel s_{1:H}), \tag{4}$$

and for any first-step mixing of Markovian policies π_u*,*

$$\mathbb{P}^{\pi_u}(s_{1:H}, a_{1:H}) = \mathbb{P}^{\mathcal{M}}(s_{1:H} \parallel a_{1:H-1}) \cdot \pi_u(a_{1:H} \parallel s_{1:H}). \tag{5}$$

Proof. Eq. (4) follows by noting that both sides are equal to

$$\rho(s_1) \prod_{h=1}^{H-1} P(s_{h+1} \mid s_h, a_h) \prod_{h=1}^{H} \pi_h(a_h \mid s_h).$$

Eq. (5) follows by noting that both sides are equal to

$$\sum_{\nu} u(\nu)\rho(s_1) \prod_{h=1}^{H-1} P(s_{h+1} \mid s_h, a_h) \prod_{h=1}^{H} \nu_h(a_h \mid s_h). \qquad \square$$

Lemma 16. *If* $\bar{\pi}_u$ *is an each-step policy in* $\bar{\Pi}_{\mathcal{B}}$*, then there exists some first-step mixture of policies in* $\tilde{\mathcal{B}}$ *that is equivalent to* $\bar{\pi}_u$*, i.e., they induce the same distribution over all length-*H *trajectories.*

Proof. Define μ as a distribution over class $\tilde{\mathcal{B}}$ with $\mu(\nu) := \prod_{h=1}^{H} u(\nu_h)$, for every $\nu = \nu_{1:H}$ in $\tilde{\mathcal{B}}$. Consider the joint action distribution under $\bar{\pi}_u$, which samples $\pi_h \sim u$ independently for each step and executes $a_h \sim \pi_h(\cdot \mid s_h)$. The resulting causally conditional distribution over actions given state sequence $s_{1:H}$ is

$$\bar{\pi}_u(a_{1:H}\|s_{1:H}) = \prod_{h=1}^{H} \left(\sum_{\pi_h \in \mathcal{B}} u(\pi_h)\, \pi_h(a_h \mid s_h) \right).$$

On the other hand, under the first-step mixture policy π_μ over $\tilde{\mathcal{B}}$, a full tuple $\nu = (\nu_1, \ldots, \nu_H)$ is sampled once from μ, and actions are drawn as $a_h \sim \nu_h(\cdot \mid s_h)$. The resulting action distribution is

$$\pi_\mu(a_{1:H}\|s_{1:H}) = \sum_{\nu \in \tilde{\mathcal{B}}} \mu(\nu) \prod_{h=1}^{H} \nu_h(a_h \mid s_h).$$

Expanding the sum yields

$$\sum_{(\nu_1,\ldots,\nu_H) \in \mathcal{B}^H} \left(\prod_{h=1}^{H} u(\nu_h)\nu_h(a_h \mid s_h) \right) = \prod_{h=1}^{H} \sum_{\pi_h \in \mathcal{B}} u(\nu_h)\, \nu_h(a_h \mid s_h),$$

by the distributive property and independence of the product.

Therefore, $\bar{\pi}_u(a_{1:H}\|s_{1:H}) = \pi_\mu(a_{1:H}\|s_{1:H})$, and both policies induce the same trajectory distribution by Lemma 15. $\square$

C.2 Proof of Theorem 6

Lemma 17. *Let $\tilde{\mathcal{B}}_{\mathrm{bc}} := \{\pi = \pi_{1:H} \in \tilde{\mathcal{B}} : \pi_h(s_h) = a_h,\ \forall h \in [H], \forall (s,a)_{1:H} \in D_{\mathrm{off}}\}$ be the set of policies in $\tilde{\mathcal{B}}$ that agree with the expert on the offline dataset of N_{off} iid expert trajectories. Assume the expert policy π^E is deterministic and realizable. Then, with probability at least $1-\delta$: for all π in $\tilde{\mathcal{B}}_{\mathrm{bc}}$,*

$$J(\pi^{\mathrm{E}}) - J(\pi) \le O\left(\frac{R\log(\tilde{B}/\delta)}{N_{\mathrm{off}}}\right). \tag{6}$$

Consequently, for all π^n's computed in WARM-STAGGER *(Algorithm 2), it holds that:*

$$J(\pi^{\mathrm{E}}) - J(\pi^n) \le O\left(\frac{R\log(\tilde{B}/\delta)}{N_{\mathrm{off}}}\right). \tag{7}$$

Proof. Eq. (6) is a direct consequence of behavior cloning's guarantee applied to Markovian policy class $\tilde{\mathcal{B}}$ ([17, Corollary 2.1]).[5]

Eq. (7) follows from Lemma 16 that $\mathbb{P}^{\pi^i}$ is a convex combination of $\mathbb{P}^{\pi}$'s for π's in $\tilde{\mathcal{B}}_{\mathrm{bc}}$, as well as the fact that the expected return function $J(\pi)$ is linear in the trajectory distribution $\mathbb{P}^{\pi}$. □

Theorem 18 (Theorem 6 restated)**.** *If Algorithm 2 is run with a deterministic expert policy π^E, an MDP $\mathcal{M}$ such that $(\mathcal{M}, \pi^E)$ is μ-recoverable, a policy class $\mathcal{B}$ such that deterministic realizability holds, and the online learning oracle $\mathbb{A}$ set as the exponential weight algorithm, then it returns $\hat{\pi}$ such that, with probability at least $1-\delta$,*

$$J(\pi^{\mathrm{E}}) - J(\hat{\pi}) \le O\left(\min\left(\frac{R\log(\tilde{B}/\delta)}{N_{\mathrm{off}}}, \frac{\mu H\log(B_{\mathrm{bc}}/\delta)}{N_{on}}\right)\right), \tag{8}$$

Proof. We bound $J(\pi^{\mathrm{E}}) - J(\hat{\pi})$ by the two terms in the minimum expression on the right hand side of Eq. (8) respectively.

For the first term, we recall from Lemma 17 that with probability $1-\delta/2$, for all policies π^n's, $J(\pi^{\mathrm{E}}) - J(\pi^n) \le O\left(\frac{R\log(\tilde{B}/\delta)}{N_{\mathrm{off}}}\right)$. Since $J(\hat{\pi}) = \frac{1}{N}\sum_{n=1}^{N} J(\pi^n)$, we have,

$$J(\pi^{\mathrm{E}}) - J(\hat{\pi}) \le O\left(\frac{R\log(\tilde{B}/\delta)}{N_{\mathrm{off}}}\right).$$

For the second term, we note that by definition $\pi^{\mathrm{E}} \in \mathcal{B}_{\mathrm{bc}}$. Thus, by applying Theorem 3, we conclude that with probability at least $1-\delta/2$, the returned $\hat{\pi}$ satisfies

$$J(\hat{\pi}) - J(\pi^{\mathrm{E}}) \le O\left(\frac{\mu H\log(B_{\mathrm{bc}}/\delta)}{N_{\mathrm{int}}}\right).$$

Together, we conclude our proof by applying a union bound. □

D Proof for Theorem 8

We formally define the MDP $\mathcal{M}$ and the expert policy from Section 4.2, where the expert policy π^{E} is deterministic

- **State Space** $\mathcal{S} = \mathbf{E} \cup \mathbf{E}' \cup \mathbf{B} \cup \mathbf{B}'$, where:
  - $\mathbf{E}$: ideal expert states, $|\mathbf{E}| = N_0$;
  - $\mathbf{E}'$: recoverable expert states, $|\mathbf{E}'| = N_1$;
  - $\mathbf{B} = \{\mathbf{b}\}$: absorbing failure state (unrecoverable);

[5] Our presentation of Theorem 2 uses a less general interpretation of that corollary, by restricting the policy class to be stationary.

- $\mathbf{B}' = \{\mathbf{b}'\}$: recoverable reset state.

- **Action Space** $\mathcal{A}$: contains $A = |\mathcal{A}|$ discrete actions. For each state $s \in \mathcal{S}$, there is a unique action $\pi^{\mathrm{E}}(s)$ taken by the expert.
- **Episode length** H**.**
- **Initial State Distribution** ρ:

$$\rho(s) = \frac{1}{(1+\beta)N_0} \text{ for all } s \in \mathbf{E}, \quad \rho(s) = \frac{\beta}{(1+\beta)N_1} \text{ for all } s \in \mathbf{E}'.$$

- **Transition Dynamics**:
  - $s \in \mathbf{E}$:
    - $a = \pi^{\mathrm{E}}(s)$: with probability $1-\beta$, transitions to a uniformly random $s' \in \mathbf{E}$; with probability β, transitions to a uniformly random $s' \in \mathbf{E}'$;
    - $a \neq \pi^{\mathrm{E}}(s)$: transitions to $\mathbf{b}$.
  - $s \in \mathbf{E}'$:
    - $a = \pi^{\mathrm{E}}(s)$: transitions to a uniformly random $s' \in \mathbf{E}$;
    - $a \neq \pi^{\mathrm{E}}(s)$: transitions to $\mathbf{b}'$.
  - $s \in \mathbf{B} = \{\mathbf{b}\}$: absorbing for all actions. Specifically, $P(s' = \mathbf{b} | s = \mathbf{b}, a) = 1$.
  - $s \in \mathbf{B}' = \{\mathbf{b}'\}$:
    - $a = \pi^{\mathrm{E}}(\mathbf{b}')$: transitions to a uniformly random $s' \in \mathbf{E}$;
    - $a \neq \pi^{\mathrm{E}}(\mathbf{b}')$: remains in $\mathbf{b}'$.
- **Reward Function**: For theoretical analysis, we consider the following reward function R_1:

$$R_1(s,a) = \begin{cases} 1 & \text{if } s \in \mathbf{E} \cup \mathbf{E}' \\ 1 & \text{if } s = \mathbf{b}' \text{ and } a = \pi^{\mathrm{E}}(s) \\ 0 & \text{otherwise} \end{cases}$$

- **Specification of Parameters**: In the following proofs, we let $H \geq \max(50, \frac{5}{4}\log(10N_0))$, $A \geq 10H$, $\beta = \frac{8}{H-8}$, $N_1 \geq \max(500, 160N_0)$.

We also make the following assumption on the policies produced in our learning algorithms (BC, STAGGER, and WARM-STAGGER). At any stage of learning, denote the set of states that are annotated by the expert by $\mathcal{S}_{\text{annotated}}$. Given the annotated (state, expert action) pairs, the learner calls some offline or online learning oracle $\mathbb{A}$ to obtain a policy π. We require π's behavior as follows:

$$\pi(\cdot \mid s) = \begin{cases} \pi^{\mathrm{E}}(\cdot \mid s), & s \in \mathcal{S}_{\text{annotated}}, \\ (\frac{1}{A}, \ldots, \frac{1}{A}), & s \notin \mathcal{S}_{\text{annotated}}. \end{cases}$$

In other words, π follows the expert's action whenever such information is available; otherwise, it takes an action uniformly at random.

Denote by $d_h^\pi(s) = \mathbb{P}^\pi(s_h = s)$ policy π's state visitation distribution in $\mathcal{M}$ at step h. We next make a simple observation that by the construction of $\mathcal{M}$, the expert policy's state-visitation distribution in $\mathcal{M}$ is stationary over all steps:

Observation 1. *For MDP* $\mathcal{M}$*,* $d_h^{\pi^E}$ *equals* ρ*, for all* $h \in [H]$.

Proof. Recall the initial state distribution:

$$\rho(s) = \begin{cases} \frac{1}{N_0(1+\beta)}, & s \in \mathbf{E}, \\ \frac{\beta}{N_1(1+\beta)}, & s \in \mathbf{E}', \\ 0, & \text{otherwise.} \end{cases}$$

Under π^{E}, states $\mathbf{b}$ and $\mathbf{b}'$ are never reached. And the induced transition kernel on $\mathbf{E} \cup \mathbf{E}'$ satisfies:

$$P(s'|s, \pi^E(s)) = \begin{cases} \frac{1-\beta}{N_0}, & s \in \mathbf{E},\ s' \in \mathbf{E}, \\ \frac{\beta}{N_1}, & s \in \mathbf{E},\ s' \in \mathbf{E}', \\ \frac{1}{N_0}, & s \in \mathbf{E}',\ s' \in \mathbf{E}, \\ 0, & \text{otherwise.} \end{cases}$$

For any fixed $s \in \mathbf{E}$, using the kernel above,

$$\sum_{s'} \rho(s')P(s \mid s', \pi^E(s')) = \sum_{s' \in \mathbf{E}} \frac{1}{N_0(1+\beta)} \cdot \frac{1-\beta}{N_0} + \sum_{s' \in \mathbf{E}'} \frac{\beta}{N_1(1+\beta)} \cdot \frac{1}{N_0} = \frac{1}{N_0(1+\beta)} = \rho(s).$$

Similarly, for any fixed $s \in \mathbf{E}'$,

$$\sum_{s'} \rho(s')P(s \mid s', \pi^E(s')) = \sum_{s' \in \mathbf{E}} \frac{1}{N_0(1+\beta)} \cdot \frac{\beta}{N_1} = \frac{\beta}{N_1(1+\beta)} = \rho(s).$$

Thus, for any $s \in \mathbf{E} \cup \mathbf{E}'$,

$$\rho(s) = \sum_{s'} \rho(s')P(s \mid s', \pi^E(s')).$$

Hence ρ is a stationary distribution for the Markov chain induced by rolloing out π^E in $\mathcal{M}$. Since $d_1^{\pi^E} = \rho$ and the dynamics are time-homogeneous, induction gives that $\forall h \in [H]$, $d_h^{\pi^E} = \rho$.

□

The following is the main theorem of this section; setting $N_0 = \Theta(N_1)$ (in which case both N_0 and N_1 are $\Theta(S)$) gives Theorem 8 in our main paper.

Theorem 19 (Strengthening of Theorem 8)**.** *To achieve smaller than $\frac{H}{2}$ suboptimality compared to expert in MDP $\mathcal{M}$ with probability $\frac{1}{2}$:*

- *Behavior Cloning (BC) using offline expert trajectories requires*
$$N_{\text{off}} = \Omega(N_1) \quad \textit{with total annotation cost } \Omega(HN_1).$$
- STAGGER *that collects interactive state-wise annotations requires*
$$N_{\text{int}} = \Omega(HN_0) \quad \textit{with total annotation cost } \Omega(CHN_0).$$

In contrast, WARM-STAGGER *learns a policy that achieves expert performance with probability at least $\frac{1}{2}$, using*

$$\begin{aligned} &N_{\text{off}} = O(\frac{N_0}{H}\log(N_0)) \quad \textit{expert trajectories, and} \\ &N_{\text{int}} \leq 3 \quad \textit{interactive annotations,} \\ &\qquad \textit{with total annotation cost } \tilde{O}(N_0 + C). \end{aligned} \tag{9}$$

Proof. The proof is divided into three parts:

First, by Lemma 20, we show that in $\mathcal{M}$, Behavior Cloning requires $\Omega(HN_1)$ expert trajectories to achieve suboptimality $H/2$ with probability $\frac{1}{2}$.

Next, in Lemma 24, we show that STAGGER, which rolls out the learner policy and queries the expert on only one state sampled uniformly from its learned policy's rollout, requires $N_{\text{int}} = \Omega(HN_0)$ interactive annotations to achieve suboptimality no greater than $H/2$ with probability $\frac{1}{2}$.

Finally, by Lemma 26, we demonstrate that WARM-STAGGER achieves expert performance using $O(\frac{N_0}{H}\log(N_0))$ offline expert tarjectories and 3 interactive annotations with probability $\frac{1}{2}$. □

D.1 Lower Bound for Behavior Cloning

Throughout this subsection, we denote by $\mathbf{E}'_{\text{annotated}}$ the set of states in $\mathbf{E}'$ that are visited and annotated by the expert's N_{off} offline trajectories.

Lemma 20 (BC suboptimality lower bound)**.** *Consider the MDP $\mathcal{M}$ and the expert policy π^{E} constructed as above. If Behavior Cloning uses*

$$N_{\text{off}} < \frac{N_1}{160}$$

iid expert trajectories, then with probability at least $\frac{1}{2}$, the suboptimality of its returned policy $\hat{\pi}$ is lower bounded by:

$$J(\pi^{\text{E}}) - J(\hat{\pi}) \geq \frac{H}{2}.$$

Proof. First, given policy $\hat{\pi}$, we define a modified policy $\tilde{\pi}$ that agrees with $\hat{\pi}$ everywhere except that it always take the expert's action on $\mathbf{E}$. Note that if $\hat{\pi}$ ever takes a wrong action at a state in $\mathbf{E}$, the trajectory deterministically falls into the absorbing bad state $\mathbf{b}$, yielding even smaller return than $\tilde{\pi}$. Thus, $J(\tilde{\pi}) \geq J(\hat{\pi})$, so it suffices to prove the claimed lower bound for $\tilde{\pi}$, which we show for the remainder of this proof.

In the following, we show that an insufficient number of expert trajectories leads to small $|\mathbf{E}'_{\text{annotated}}|$ i.e., poor coverage on $\mathbf{E}'$. This, in turn, causes policy $\tilde{\pi}$ to frequently fail to recover and get trapped in the absorbing bad state $\mathbf{b}'$, incurring a large suboptimality compared to the expert.

By Lemma 21, when N_{off} is below the stated threshold, with probability $\frac{1}{2}$,

$$|\mathbf{E}'_{\text{annotated}}| \leq \frac{1}{10}|\mathbf{E}'|$$

We henceforth condition on this happening. In this case, recall that our offline learning oracle is such that $\tilde{\pi}$ takes actions uniformly at random in states in $\mathbf{E}' \setminus \mathbf{E}'_{\text{annotated}}$. Suppose we roll out the policy $\tilde{\pi}$ in $\mathcal{M}$; let τ be the first step such that $s_\tau \in \mathbf{E}'$ (and $\tau := H+1$ if no such step exists).

By Lemma 22,

$$\Pr_{\mathcal{M},\tilde{\pi}}(\tau \leq H/5) \geq 0.79, \text{ and } \Pr_{\mathcal{M},\tilde{\pi}}\left(s_\tau \notin \mathbf{E}'_{\text{annotated}} \mid \tau \leq H/5\right) \geq 0.9.$$

We henceforth condition on the event $\{\tau \leq H/5,\ s_\tau \notin \mathbf{E}'_{\text{annotated}}\}$ when rolling out $\tilde{\pi}$. Applying Lemma 23, we have, with probability at least 0.9, $\tilde{\pi}$ takes a wrong action at s_τ and transitions to $\mathbf{b}'$, and subsequently never takes the expert recovery action at $\mathbf{b}'$. Therefore the trajectory remains in $\mathbf{b}'$ from step $H/5$ onward, yielding zero reward for at least $\frac{4H}{5}$ steps.

Multiplying all factors, with probability greater than $\frac{1}{2}$,

$$J(\pi^E) - J(\tilde{\pi}) \geq \underbrace{0.79}_{\text{reach } \mathbf{E}'} \times \underbrace{0.9}_{\text{unannotated } s_\tau} \times \underbrace{0.9}_{\text{action errors}} \times \underbrace{0.8H}_{\text{zero reward}} > \frac{H}{2},$$

which concludes the proof. □

Lemma 21 (Bounded $\mathbf{E}'$ coverage)**.** *If the number of expert trajectories satisfies:*

$$N_{\text{off}} \leq \frac{N_1}{160},$$

then, with probability at least $\frac{1}{2}$*,*

$$|\mathbf{E}'_{\text{annotated}}| \ \leq \ \frac{N_1}{10}.$$

Proof. Recall that by Observation 1, $d_h^{\pi^E}(s) = \frac{\beta}{N_1(1+\beta)} = \frac{8}{HN_1}$ for $s \in \mathbf{E}'$, where $\beta = \frac{8}{H-8}$. Denote $X := |\mathbf{E}'_{\text{annotated}}|$ to be the number of annotated states in $\mathbf{E}'$. Consider N_{off} expert trajectories, each of length H, and let $s_{i,h}$ be the state at step h of trajectory i. Fix any $s \in \mathbf{E}'$. By a union bound over all steps and available offline expert trajectories,

$$\Pr(s \in \mathbf{E}'_{\text{annotated}}) = \Pr\Big(\bigcup_{i=1}^{N_{\text{off}}}\bigcup_{h=1}^{H}\{s_{i,h} = s\}\Big) \leq \sum_{i=1}^{N_{\text{off}}}\sum_{h=1}^{H}\Pr(s_{i,h} = s) = \sum_{i=1}^{N_{\text{off}}}\sum_{h=1}^{H} d_h^{\pi^E}(s) = \frac{8N_{\text{off}}}{N_1}.$$

Under the assumption $N_{\text{off}} \leq N_1/160$, this gives

$$\Pr(s \in \mathbf{E}'_{\text{annotated}}) \leq \frac{1}{20}.$$

Let Z_s be the indicator that state s appears in the expert trajectories, by linearity of expectation,

$$\mathbb{E}[X] = \sum_{s \in \mathbf{E}'} \mathbb{E}[Z_s] = \sum_{s \in \mathbf{E}'} \Pr(s \in \mathbf{E}'_{\text{annotated}}) \leq \frac{N_1}{20}.$$

Applying Markov's inequality at threshold $N_1/10$,

$$\Pr\left(X > \frac{N_1}{10}\right) \le \frac{\mathbb{E}[X]}{N_1/10} \le \frac{N_1/20}{N_1/10} = \frac{1}{2}.$$

Equivalently,

$$\Pr\left(X \le \frac{N_1}{10}\right) \ge \frac{1}{2}.$$ □

Lemma 22 (First $\mathbf{E}'$ visit)**.** *For the MDP $\mathcal{M}$, and any policy π that agrees with π^{E} on $\mathbf{E}$:*

$$\Pr_{\mathcal{M},\pi}\left(\exists h \in [H/5], s_t \in \mathbf{E}'\right) \ge 0.79$$

Proof. Since π agrees with π^{E} on $\mathbf{E}$,

$$\Pr_{\mathcal{M},\pi}\left(\exists h \in [H/5], s_t \in \mathbf{E}'\right) = \Pr_{\mathcal{M},\pi^{\mathrm{E}}}\left(\exists h \in [H/5], s_t \in \mathbf{E}'\right).$$

It suffices to consider the expert policy π^{E}'s visitation. By Observation 1, the state visitation distribution for π^{E} satisfies that for all $h \in [H]$,

$$d_h^{\pi^E}(s) = \begin{cases} \frac{1}{N_0(1+\beta)}, & s' \in \mathbf{E}, \\ \frac{\beta}{N_1(1+\beta)}, & s' \in \mathbf{E}, \\ 0, & \text{otherwise.} \end{cases}$$

The probability that *no* $s \in \mathbf{E}'$ is visited in $\frac{H}{5}$ steps is:

$$\begin{aligned} \frac{1}{1+\beta} \cdot (1-\beta)^{\frac{H}{5}-1} =& \frac{1}{(1+\beta)(1-\beta)} \left(\frac{1}{1+\beta}\right)^{\frac{H}{5}} \\ =& \frac{1}{1-\beta^2}\left(1-\frac{8}{H}\right)^{\frac{H}{5}} \\ \le& \frac{1}{1-\frac{64}{(H-8)^2}} e^{-\frac{8}{H}\cdot\frac{H}{5}} \\ <& 1.038 \cdot e^{-\frac{8}{5}} < 0.21, \end{aligned} \tag{10}$$

where we apply $\beta = \frac{8}{H-8}$, $1 - x \le e^{-x}$, and $H \ge 50$. Thus:

$$\Pr_{\mathcal{M},\pi^{\mathrm{E}}}\left(\exists h \in [H/5], s_t \in \mathbf{E}'\right) > 1 - 0.2095 = 0.79$$ □

Lemma 23 (Action Errors on Unannotated States)**.** *Consider MDP $\mathcal{M}$ with action space size $A \ge 10H$.*

$$\Pr_{\mathcal{M},\tilde{\pi}}\left(\forall h \in [H] : s_h \in (\mathbf{E}' \setminus \mathbf{E}'_{\text{annotated}}) \cup \mathbf{B}' \implies a_h \ne \pi^E(s_h)\right) \ge 0.9.$$

Proof. Since for any state in $(\mathbf{E}' \setminus \mathbf{E}'_{\text{annotated}}) \cup \mathbf{B}'$, policy $\tilde{\pi}$ selects the expert action with probability exactly $\frac{1}{A}$, thus,

$$\begin{aligned} &\Pr_{\mathcal{M},\tilde{\pi}}\left(\exists h \in [H] : \ s_h \in (\mathbf{E}' \setminus \mathbf{E}'_{\text{annotated}}) \cup \mathbf{B}' \text{ and } a_h = \pi^E(s_h)\right) \\ \le& \sum_{h=1}^{H} \Pr_{\mathcal{M},\tilde{\pi}}\left(s_h \in (\mathbf{E}' \setminus \mathbf{E}'_{\text{annotated}}) \cup \mathbf{B}' \text{ and } a_h = \pi^E(s_h)\right) \\ \le& \sum_{h=1}^{H} \Pr_{\mathcal{M},\tilde{\pi}}\left(a_h = \pi^E(s_h) \mid s_h \in (\mathbf{E}' \setminus \mathbf{E}'_{\text{annotated}}) \cup \mathbf{B}'\right) \le \frac{H}{A} \le 0.1, \end{aligned}$$

where we recall that $A \ge 10H$ in our construction. The lemma now follows by taking the complement of the probability above. □

D.2 Lower Bound for STAGGER

Throughout the proof, denote by $\mathbf{E}_{\text{annotated}}$ the final set of states in $\mathbf{E}$ on which STAGGER have requested expert annotations.

Lemma 24 (STAGGER suboptimality lower bound)**.** *Consider the MDP $\mathcal{M}$ from Section 4.2 with $H \geq 50$, $A \geq 10H$, and $\beta = \frac{8}{H-8}$. If* STAGGER *collects no more than*

$$N_{\text{int}} \leq \frac{HN_0}{12}$$

interactive state-wise annotations, then, with probability at least $\frac{1}{2}$, the returned policy $\hat{\pi}$ suffers suboptimality at least

$$J(\pi^{\text{E}}) - J(\hat{\pi}) \geq \frac{H}{2}.$$

Proof. By Lemma 25, if STAGGER collects at most $\frac{HN_0}{12}$ interactive state-wise annotations as above, then with probability at least $\frac{1}{2}$, $|\mathbf{E}_{\text{annotated}}|$, the number of distinct states in $\mathbf{E}$ annotated is fewer than $N_0/3$. Consider a random rollout of $\hat{\pi}$, and define the following events:

$$F_1 := \{s_1 \notin \mathbf{E}_{\text{annotated}}\},$$
$$F_2 := \{a_1 \neq \pi^{\text{E}}(s_1)\}.$$

We now lower bound their probabilities:

$$\Pr_{\hat{\pi}}(F_1) \geq \frac{2}{3(1+\beta)},$$
$$\Pr_{\hat{\pi}}(F_2 \mid F_1) \geq 1 - \frac{1}{A} \geq 1 - \frac{1}{10H}.$$

Conditioned on the two events, the agent will get trapped at state $\mathbf{b}$ from step 2 on, and thus its conditional expected return satisfies

$$\mathbb{E}_{\hat{\pi}}\left[\sum_{h=1}^{H} r_h \mid F_1, F_2\right] \leq 1.$$

Also, by the definition of the reward function R_1, $J(\pi^E) = H$. Thus,

$$\begin{aligned} J(\pi^E) - J(\hat{\pi}) =& \mathbb{E}_{\hat{\pi}}\left[H - \sum_{h=1}^{H} r_h\right] \\ \geq& \mathbb{P}_{\hat{\pi}}(F_1, F_2) \cdot \mathbb{E}_{\hat{\pi}}\left[H - \sum_{h=1}^{H} r_h \mid F_1, F_2\right] \\ \geq& \frac{2}{3} \cdot \frac{H-8}{H} \cdot \frac{10H-1}{10H} \cdot (H-1) \geq \frac{H}{2}, \end{aligned}$$

where we use our setting of $\beta = \frac{8}{H-8}$ and apply $H \geq 50$. □

Lemma 25 (Bounded $\mathbf{E}$ coverage under STAGGER)**.** *Suppose* STAGGER *collects at most*

$$N_{\text{int}} \leq \frac{HN_0}{12}$$

interactive annotations. Then,

$$\Pr\left(|\mathbf{E}_{\text{annotated}}| \geq \frac{N_0}{3}\right) \leq \frac{1}{2}.$$

Proof. In this proof, we say that state s is *annotated at iteration i*, if it has been annotated by the expert before iteration i (excluding iteration i). Since each iteration of STAGGER samples one state uniformly from the current policy's rollout for annotation, we denote the indicator of expert

annotating an unannotated state from $\mathbf{E}$ at iteration i as $X_i \in \{0, 1\}$. With this notation, we have the total number of annotated states by the end of iteration N_{int} as

$$|\mathbf{E}_{\text{annotated}}| = \sum_{i=1}^{N_{\text{int}}} X_i.$$

Let $\mathcal{F}_j$ be the sigma-algebra generated by all information seen by STAGGER up to iteration j. We now upper bound the expected value of X_i conditioned on $\mathcal{F}_{i-1}$ for each i.

Denote by Y_i the number of unannotated states in $\mathbf{E}$ visited by round i's rollout (by policy π^i). We claim that conditioned on $\mathcal{F}_{i-1}$, Y_i is stochastically dominated by a geometric distribution with parameter $\frac{A-1}{A}$. Indeed, whenever an unannotated state in $\mathbf{E}$ is encountered when rolling out π^i, the probability that the agent takes a wrong action is $\frac{A-1}{A}$; if so, the agent gets absorbed to $\mathbf{b}$ immediately, and thus never sees any new unannotated states in this episode. In summary,

$$\mathbb{E}[Y_i \mid \mathcal{F}_{i-1}] \leq \mathbb{E}_{Z \sim \text{Geometric}(\frac{A-1}{A})}[Z] = \frac{A}{A-1} \leq 2.$$

Since the state sampled for expert annotation is uniformly at random from the trajectory, conditioned on Y_i, the probability that it lands on a an unannotated state in $\mathbf{E}$ is at most $\frac{Y_i}{H}$. Hence,

$$\mathbb{E}[X_i \mid \mathcal{F}_{i-1}] = \mathbb{E}\left[\mathbb{E}[X_i \mid Y_i, \mathcal{F}_{i-1}] \mid \mathcal{F}_{i-1}\right] = \mathbb{E}\left[\frac{Y_i}{H} \mid \mathcal{F}_{i-1}\right] \leq \frac{2}{H}.$$

By linearity of expectation:

$$\mathbb{E}[|\mathbf{E}_{\text{annotated}}|] = \sum_{i=1}^{N_{\text{int}}} \mathbb{E}[X_i] \leq \frac{2N_{\text{int}}}{H}.$$

Applying Markov's inequality:

$$\Pr(|\mathbf{E}_{\text{annotated}}| \geq N_0/3) \leq \frac{\mathbb{E}[X]}{N_0/3} \leq \frac{6N_{\text{int}}}{HN_0} \leq \frac{1}{2},$$

where the last inequality is by our assumption that $N_{\text{int}} \leq \frac{HN_0}{12}$. This completes the proof. □

D.3 Upper Bound for WARM-STAGGER

Lemma 26 (Hybrid IL achieves expert performance under R_1)**.** *Consider the MDP $\mathcal{M}$ and expert policy π^{E} as above, then, with probability at least $1/2$,* WARM-STAGGER *outputs a policy that achieves expert performance using $N_{\text{off}} = O\left(\frac{N_0}{H}\log(N_0)\right)$ offline expert trajectories and $N_{\text{int}} \leq 3$ interactive annotations.*

Proof. We divide the proof into four steps. Throughout, we denote by $\mathbf{E}_{\text{annotated}}$ and $\mathbf{E}'_{\text{annotated}}$ the subsets of $\mathbf{E}$ and $\mathbf{E}'$, respectively, that are annotated by the N_{off} offline expert demonstration trajectories.

First, we state a high probability event for the N_{off} offline expert demonstration trajectories to provide annotations on all states in $\mathbf{E}$. Define event

$$F_3 := \{\mathbf{E}_{\text{annotated}} = \mathbf{E}\}.$$

By Lemma 27, the choice

$$N_{\text{off}} = \frac{N_0}{(1-\beta)H}\log\Big(10N_0\Big)$$

ensures that $\Pr(F_3) \geq 0.9$. On event F_3, the learner takes the correct action on every state in $\mathbf{E}$.

Next, we define the event under which only a small fraction of $\mathbf{E}'$ is covered by the expert. With our setting of MDP parameters, the number of offline trajectories satisfies

$$N_{\text{off}} = \frac{N_0}{(1-\beta)H}\log\Big(10N_0\Big) \leq \frac{4}{5(1-\beta)}N_0 = \frac{4(H-8)}{5(H-16)}N_0 \leq N_0 < \frac{N_1}{160},$$

where in the first inequality we apply $H \geq \frac{5}{4}\log(10N_0)$; in the second inequality, we apply $\frac{H-16}{H-8} < \frac{4}{5}$ when $H \geq 50$; and we apply $N_0 < \frac{N_1}{160}$ in the third inequality by the MDP setting. Thus, by Lemma 21,

$$\mathbb{E}[|\mathbf{E}'_{\text{annotated}}|] \leq \frac{N_1}{20}.$$

Applying Markov's inequality at threshold $\frac{N_1}{4} - 2$ gives

$$\Pr\left(|\mathbf{E}'_{\text{annotated}}| > \frac{N_1}{4} - 2\right) \leq \Pr\left(|\mathbf{E}'_{\text{annotated}}| > \frac{N_1}{4} - \frac{2N_1}{500}\right) \leq \frac{25}{121} < 0.21.$$

Define event

$$F_4 := \left\{|\mathbf{E}'_{\text{annotated}}| \leq \frac{N_1}{4} - 2\right\}.$$

Then $\Pr(F_4) \geq 0.79$.

Third, we show that under events F_3 and F_4, the interactive phase of WARM-STAGGER gets $\mathbf{b}'$ annotated with probability greater than 0.4 in each of the first three rollouts, such that $\mathbf{b}'$ is annotated within three iterations with good probability.

First, we observe that by union bound,

$$\Pr(F_3, F_4) \geq \Pr(F_3) + \Pr(F_4) - 1 \geq 0.69.$$

We henceforth condition on $F_3 \cap F_4$ happening. By the definition of F_3, the policies π^n produced by WARM-STAGGER acts optimally on all states in $\mathbf{E}$.

Denote by $\mathbf{E}'^{,n}_{\text{annotated}}$ the set of annotated states in $\mathbf{E}$ after WARM-STAGGER's iteration n; with this notation, $\mathbf{E}'^{,0}_{\text{annotated}} = \mathbf{E}'_{\text{annotated}}$, and $|\mathbf{E}'^{,n+1}_{\text{annotated}}| \leq |\mathbf{E}'^{,n}_{\text{annotated}}| + 1$ for all n. Thus, by the definition of F_4, for every $n = 0, 1, 2$, $\mathbf{E}'^{,n+1}_{\text{annotated}} \leq \frac{N_1}{4} - 2 + 2 = \frac{N_1}{4}$. In other words, at least $\frac{3}{4}$ of the states in $\mathbf{E}'$ are unannotated for each of the first three iterations.

During the rollout of a π^n, let τ be the first step such that $s_\tau \in \mathbf{E}'$. By Lemma 22, for every $n \in \{1, 2, 3\}$,

$$\Pr_{\mathcal{M},\pi^n}(\tau \leq H/5) \geq 0.79.$$

Since by the definition of $\mathcal{M}$, s_τ is drawn uniformly from $\mathbf{E}'$ conditioned on $\tau \leq H/5$,

$$\Pr_{\mathcal{M},\pi^n}(s_\tau \notin \mathbf{E}'^{,n}_{\text{annotated}} \mid \tau \leq H/5) \geq 0.75.$$

Therefore,

$$\Pr_{\mathcal{M},\pi^n}(s_\tau \notin \mathbf{E}'^{,n}_{\text{annotated}}, \tau \leq H/5) \geq 0.79 \times 0.75 \geq 0.59 \tag{11}$$

With $A \geq 10H$, by Lemma 23,

$$\Pr_{\mathcal{M},\pi^n}\left(\forall h \in [H] : s_h \in (\mathbf{E}' \setminus \mathbf{E}'^{,n}_{\text{annotated}}) \cup \mathbf{B}' \implies a_h \neq \pi^E(s_h)\right) \geq 0.9. \tag{12}$$

Thus, when the events in Eqs. (11) and (12) happen simultaneously (which happens with probability $\geq 0.59 + 0.9 - 1 \geq 0.49$ by union bound), the trajectory rolled out by π^n transitions to $\mathbf{b}'$ at step $\tau + 1$ and stays there till the end of episode, accumulating no less than $0.8H$ $\mathbf{b}'$ states. Since WARM-STAGGER samples one state uniformly from each rollout for annotation, this gives 0.8 probability of $\mathbf{b}'$ being annotated with probability ≥ 0.49.

Denote by $\mathcal{F}_n$ the sigma-algebra generated by observations up to iteration n of WARM-STAGGER. Our reasoning above implies that for all history $\mathcal{F}_{n-1}$ such that $F_3 \cap F_4$ happens, for all $n \in \{1, 2, 3\}$,

$$\Pr(\mathbf{b}' \text{ is annotated at iteration } n \mid \mathcal{F}_{n-1}) \geq 0.49 \times 0.8 \geq 0.39.$$

Define event

$$F_5 := \{\mathbf{b}' \text{ is annotated within 3 iterations}\}.$$

Hence

$$\Pr(F_5 \mid F_3, F_4) \geq 1 - (1 - 0.39)^3 = 1 - 0.61^3 > 0.77.$$

Combining the three events, we have

$$\Pr(F_3 \cap F_4 \cap F_5) \;\geq\; \Pr(F_3 \cap F_4)\,\Pr(F_5 \mid F_3, F_4) \;\geq\; 0.69 \times 0.77 > 0.5.$$

Finally, on $F_3 \cap F_4 \cap F_5$, all states in $\mathbf{E}$ and the reset state $\mathbf{b}'$ are annotated, under reward function R_1, the learner receives reward 1 at any step if it is in $\mathbf{E}$ or $\mathbf{E}'$ or it is in $\mathbf{b}'$ and takes the recovery action same as the expert. Since learned policy now behaves identical to π^{E} on all states in $\mathbf{E}$ and can successfully recover in $\mathbf{B}'$, its total return is the same as the expert, which concludes the proof. □

Lemma 27 (Coverage of states in $\mathbf{E}$ with N_{off} trajectories)**.** *Consider the MDP $\mathcal{M}$. For N_{off} trajectories of length H rolled out by expert policy π^{E}, all N_0 states in $\mathbf{E}$ are annotated with probability $\geq 1-\delta$ if:*

$$N_{\mathrm{off}} \;\geq\; \frac{N_0}{(1-\beta)H}\log\Big(\frac{N_0}{\delta}\Big).$$

Proof. Recall that we denote by $\mathbf{E}_{\mathrm{annotated}}$ the set of states in $\mathbf{E}$ visited and annotated by offline expert demonstrations. Fix any $s \in \mathbf{E}$ and concatenate all expert trajectories into a single sequence $\{s_t\}_{t=1}^{T}$ of length $T := HN_{\mathrm{off}}$, ordered from the first state of the first trajectory to the last state of the last trajectory. Let $\mathcal{F}_t$ be the sigma-algebra generated by all states $s_1, \ldots, s_t$, and define the set of initial indices

$$I_0 := \{\,1,\; H+1,\; 2H+1,\; \ldots,\; (N_{\mathrm{off}}-1)H+1\,\}.$$

For any $t+1 \in I_0$, s_{t+1} is drawn from ρ, so

$$\Pr(s_{t+1} = s \mid \mathcal{F}_t) = \rho(s) = \frac{1}{(1+\beta)N_0}.$$

For $t+1 \notin I_0$, we have $s_t \in \mathbf{E} \cup \mathbf{E}'$ and under π^E:

$$\Pr(s_{t+1} = s \mid s_t) = \begin{cases} \dfrac{1-\beta}{N_0}, & s_t \in \mathbf{E}, \\ \dfrac{1}{N_0}, & s_t \in \mathbf{E}'. \end{cases}$$

Therefore, in all cases,

$$\Pr(s_{t+1} = s \mid \mathcal{F}_t) \;\geq\; \frac{1-\beta}{N_0}, \qquad \Pr(s_{t+1} \neq s \mid \mathcal{F}_t) \;\leq\; 1 - \frac{1-\beta}{N_0}.$$

Let $A_t := \{s_1 \neq s, \ldots, s_t \neq s\}$. Then

$$\Pr(A_{t+1}) \leq \Big(1 - \frac{1-\beta}{N_0}\Big)\Pr(A_t),$$

Thus,

$$\Pr\left(s \notin \mathbf{E}_{\mathrm{annotated}}\right) = \Pr(A_T) \leq \Big(1 - \frac{1-\beta}{N_0}\Big)^{T} \leq \exp\Big(-\frac{(1-\beta)T}{N_0}\Big) = \exp\Big(-\frac{(1-\beta)HN_{\mathrm{off}}}{N_0}\Big).$$

By a union bound over all $s \in \mathbf{E}$,

$$\Pr(\exists s \in \mathbf{E}, s \notin \mathbf{E}_{\mathrm{annotated}}) \leq N_0 \exp\Big(-\frac{(1-\beta)HN_{\mathrm{off}}}{N_0}\Big),$$

which is at most δ whenever

$$N_{\mathrm{off}} \;\geq\; \frac{N_0}{(1-\beta)H}\log\Big(\frac{N_0}{\delta}\Big).$$

□

E Additional Guarantees for a Trajectory-wise DAgger Variant without Recoverability Assumption

In this section, we revisit and conduct a refined analysis of another variant of DAgger with trajectory-wise annotations. We show that without the recoverability assumption, an interactive IL algorithm has sample complexity no worse than that of behavior cloning. This result complements [17] that analyzes a different version of trajectory-wise DAgger, which they proved to have a worse sample complexity guarantee than behavior cloning.

E.1 Additional Notations and Useful Distance Measures

Recall that we have defined first-step mixing of policies in Section C.1. We instantiate it to define a mixture of policies in $\mathcal{B}$, which induces a useful policy class below:

Definition 28 (First-step mixing of $\mathcal{B}$)**.** *Define $\Pi_{\mathcal{B}} := \{\pi_u : u \in \Delta(\mathcal{B})\}$, where policy π_u is executed in an an episode of an MDP $\mathcal{M}$ by: draw $\pi \sim u$ at the beginning of the episode, and execute policy π throughout the episode.*

In the following, we present another two useful distance measures for a pair of policies.

Definition 29 (Trajectory-wise L_∞-semi-metric [17])**.** *For a pair of Markovian policies π and π', define their trajectory-wise L_∞-semi-metric as*

$$\rho(\pi \parallel \pi') := \mathbb{E}^{\pi}\mathbb{E}_{a'_{1:H}\sim\pi'(\cdot\|s_{1:H})}\left[\mathbb{I}\left\{\exists h : a_h \neq a'_h\right\}\right].$$

$\rho(\pi \parallel \pi')$ is the probability of any action taken by π' deviating from actions in trajectories induced by π, which is symmetric [17]. A bound on $\rho(\pi \parallel \pi^{\mathrm{E}})$ leads to straightforward performance difference guarantee: $J(\pi^{\mathrm{E}}) - J(\pi) \leq R \cdot \rho(\pi \parallel \pi^{\mathrm{E}})$ [17] (Lemma 40).

Recall that we have defined causually conditional probabilities in Section C.1. Built upon it, we introduce the following definition:

Definition 30 (Decoupled Hellinger distance)**.** *For a pair of Markovian policies π and π', define their decoupled Hellinger distance as $\mathbb{E}^{\pi}\left[D^2_H(\pi(\cdot\|s_{1:H}), \pi'(\cdot\|s_{1:H}))\right]$.*

Similarly, $\mathbb{E}^{\pi}\left[D^2_{\mathrm{H}}(\pi(\cdot\|s_{1:H}), \pi'(\cdot\|s_{1:H}))\right]$ denotes the expected Hellinger distance between the causal distribution of actions $\pi(\cdot\|s_{1:H})$ and $\pi'(\cdot\|s_{1:H})$ on state sequence $s_{1:H}$ visited by π. This allows decoupled analysis for state and action sequences, which is useful for our analysis below.

E.2 Interactive IL Matches Offline IL on Trajectory-wise Annotation Sample Complexity

We present TRAGGER (Algorithm 3), another variant of DAgger that operates in the trajectory-wise sampling model, and provide its sample complexity bounds.

Algorithm 3 TRAGGER: DAgger with trajectory-wise annotation oracle

1: **Input:** MDP $\mathcal{M}$, deterministic expert π^{E}, stationary policy class $\mathcal{B}$, online learning oracle $\mathbb{A}$ with decision space $\Pi_{\mathcal{B}}$.
2: **for** $n = 1, \ldots, N$ **do**
3: Query $\mathbb{A}$ and receive $\pi^n \in \Pi_{\mathcal{B}}$.
4: Roll out π^n and sample $s^n_{1:H}$ following $\mathbb{P}^{\pi^n}$. Query $\mathcal{O}^{\mathrm{Traj}}$ for $a^{*,n}_{1:H} = \pi^{\mathrm{E}}(s^n_{1:H})$.
5: Update $\mathbb{A}$ with loss function

$$\ell^n(\pi) := \log\left(\frac{1}{\pi(a^{*,n}_{1:H} \parallel s^n_{1:H})}\right). \tag{13}$$

6: **end for**
7: Output $\hat{\pi}$, the first-step uniform mixture of policies in $\left\{\pi^1, \ldots, \pi^n\right\}$.

Algorithm 3 uses first-step mixing policies $\pi_u \in \Pi_{\mathcal{B}}$ (recall Definition 28). At round n, it rolls out $\pi^n = \pi_{u^n}$ obtained from an online learning oracle $\mathbb{A}$ and samples a full state sequence $s^n_{1:H}$. Similar to Algorithm 1, Algorithm 3 also requires $\mathbb{A}$ to have decision space $\Pi_{\mathcal{B}}$ (cf. $\bar{\Pi}_{\mathcal{B}}$ in Algorithm 1). It then requests expert's trajectory-wise annotation $a^{*,n}_{1:H}$ and updates $\mathbb{A}$ by $\ell^n(\pi)$ (Eq. (13)). At the end of iteration N, the uniform first-step mixing of $\{\pi^n\}_{n=1}^N$ is returned, which is equivalent to returning $\pi_{\hat{u}}$, where $\hat{u} := \frac{1}{N}\sum_{n=1}^N u^n$. We provide the following performance guarantee of Algorithm 3:

Theorem 31. *If Algorithm 3 is run with a deterministic expert policy π^E, a policy class $\mathcal{B}$ such that realizability holds, and the online learning oracle $\mathbb{A}$ set as the exponential weight algorithm, then it returns $\hat{\pi}$ such that, with probability at least $1-\delta$,*

$$J(\pi^{\mathrm{E}}) - J(\hat{\pi}) \leq 2R\frac{\log(B) + 2\log(1/\delta)}{N}.$$

Theorem 31 shows that in the deterministic realizable setting, the interactive IL Algorithm 3 has a trajectory-wise sample complexity matching that of behavior cloning [17]. In contrast, prior state-of-the-art analysis of interactive IL algorithms [17, Appendix C.2] gives sample complexity results that are in general worse than behavior cloning. [6]

For the proof of Theorem 31, we introduce a new notion called decoupled Hellinger estimation error:

$$\mathrm{OnEst}_N^{\mathrm{Traj}} := \sum_{n=1}^{N} \mathbb{E}^{\pi^n} \left[D^2_{\mathrm{H}}(\pi^n(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H})) \right].$$

$\mathrm{OnEst}_N^{\mathrm{Traj}}$ decouples the dependence between the state sequence and the distribution of action sequence induced by the learner. Perhaps surprisingly, as we show below, it is compatible with non-Markovian first-step mixing of policies, while still being well-behaved enough to be translated to a policy suboptimality guarantee, which could be of independent interest.

E.3 Decoupling State and Action Sequences by Decoupled Hellinger Distance

In this section, we demonstrate that similar to $D^2_{\mathrm{H}}\left(\mathbb{P}^{\pi}, \mathbb{P}^{\pi^{\mathrm{E}}}\right)$ [17], the decoupled Hellinger distance $\mathbb{E}^{\pi}\left[D^2_{\mathrm{H}}(\pi(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H}))\right]$ is also proportionally lower bounded by a constant factor of $\rho(\pi \parallel \pi^{\mathrm{E}})$. The following two lemmas show that such relationship holds for both Markovian policies and their first-step mixings.

Lemma 32. *Let π^{E} be a deterministic policy, and let π be a Markovian policy. Then we have*

$$\frac{1}{2} \cdot \rho(\pi \parallel \pi^{\mathrm{E}}) \le \mathbb{E}^{\pi}\left[D^2_H(\pi(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H}))\right].$$

Lemma 33. *Let π^{E} be a deterministic policy, and let π_u be a first-step mixing of Markovian policies. Then we have*

$$\frac{1}{2} \cdot \rho(\pi_u \parallel \pi^{\mathrm{E}}) \le \mathbb{E}^{\pi_u}\left[D^2_H(\pi_u(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H}))\right].$$

The cornerstone of the above two lemmas is the following special case about first-step mixing of deterministic policies. Given an MDP with finite state space size S and action space size A, we denote the set of all deterministic, Markovian policies, as $\mathcal{B}^{\mathrm{Det}}$. $\mathcal{B}^{\mathrm{Det}}$ contains A^{SH} deterministic policies, which can be indexed by a tuple of actions $(a_{h,s})_{h\in[H],\, s\in\mathcal{S}}$.

Lemma 34. *Let π^{E} be a deterministic Markovian policy, and let π_u be a first-step mixing of deterministic Markovian policies (i.e., elements of $\mathcal{B}^{\mathrm{Det}}$). Then we have that*

$$\frac{1}{2} \cdot \rho(\pi_u \parallel \pi^{\mathrm{E}}) \le \mathbb{E}^{\pi_u}\left[D^2_H(\pi_u(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H}))\right].$$

We now quickly conclude Lemmas 32 and 33 using Lemma 34 in the next subsection, then come back to the proof of Lemma 34 in the subsection after.

E.3.1 Proofs of Lemmas 32 and 33

Proof of Lemma 32. We show the following simple claim: any Markovian policy π is equivalent to a first-step mixing of a set of deterministic Markovian policies. This allows us to apply guarantees for mixtures of deterministic Markovian policies in Lemma 34 to conclude the proof. □

Claim 35. *For a Markovian policy $\pi = (\pi_1, \ldots, \pi_H)$, there exists a first-step mixing of deterministic policy π_u such that for any $s_{1:H} \in \mathcal{S}^H$, 1. $\pi(\cdot\|s_{1:H}) = \pi_u(\cdot\|s_{1:H})$, and 2. $\mathbb{P}^{\pi}(s_{1:H}) = \mathbb{P}^{\pi_u}(s_{1:H})$.*

[6] For [17, Appendix C.2]'s sample complexity to improve over behavior cloning, we need $\mu H \max_{h\in[H]} \log|\mathcal{B}_h|$ to be significantly smaller $R\log|\mathcal{B}|$ (where $\mathcal{B}_h$ is the projection of $\mathcal{B}$ onto step h). This may require the strong condition that $\mu \ll R/H \le 1$ in the more practical parameter-sharing settings ($|\mathcal{B}_h| = |\mathcal{B}|$).

Proof of Claim 35. To construct policy π_u, we will set the weight vector u (over $\mathcal{B}^{\mathrm{Det}}$) such that its weight on policy ν indexed by $(a_{h,s})_{h\in[H],s\in\mathcal{S}}$ as:

$$u(\nu) = \prod_{h=1}^{H}\prod_{s\in\mathcal{S}} \pi_h(a_{h,s}|s) \tag{14}$$

It can be easily verified that $\sum_{\nu\in\mathcal{B}^{\mathrm{Det}}} u(\nu) = 1$.

We now verify the first item. By first-step mixing, we rewrite $\pi_u(a_{1:H} \parallel s_{1:H})$ as

$$\begin{aligned}
\pi_u(a_{1:H} \parallel s_{1:H}) &= \sum_{\nu\in\mathcal{B}^{\mathrm{Det}}} u(\nu)\prod_{h=1}^{H}\nu_h(a_h|s_h) \\
&= \sum_{(a'_{h,s})_{h\in[H],s\in\mathcal{S}}} \prod_{h=1}^{H}\prod_{s\in\mathcal{S}} \pi_h(a'_{h,s}|s)\prod_{h=1}^{H}\mathbb{I}\left[a'_{h,s_h}=a_h\right] \\
&= \sum_{(a'_{h,s})_{h\in[H],s\neq s_h}} \prod_{h=1}^{H}\prod_{s\neq s_h} \pi_h(a'_{h,s}|s) \sum_{(a'_{h,s})_{h\in[H],s=s_h}} \prod_{h=1}^{H}\pi_h(a'_{h,s_h}|s_h)\prod_{h=1}^{H}\mathbb{I}\left[a'_{h,s_h}=a_h\right] \\
&= \sum_{(a'_{h,s})_{h\in[H],s\neq s_h}} \prod_{h=1}^{H}\prod_{s\neq s_h} \pi_h(a'_{h,s}|s)\prod_{h=1}^{H}\pi_h(a_h|s_h) \\
&= \prod_{h=1}^{H}\pi_h(a_h|s_h) = \pi(a_{1:H} \parallel s_{1:H}).
\end{aligned} \tag{15}$$

Since this holds for any action sequence $a_{1:H} \in \mathcal{A}^H$, we derive the first part of Claim 35 that $\pi(\cdot\|s_{1:H}) = \pi_u(\cdot\|s_{1:H})$. The second item follows from the first item combined with Lemma 15. □

Proof of Lemma 33. By Claim 35, any Markovian policy can be viewed as a first-step mixing of A^{SH} deterministic policies from $\mathcal{B}^{\mathrm{Det}}$, thus, any first-step mixing of Markovian policies π_u can also be viewed as a first-step mixing of A^{SH} deterministic policies from $\mathcal{B}^{\mathrm{Det}}$. The proof again follows by applying Lemma 34. □

E.3.2 Proof of Lemma 34

To facilitate the proof of Lemma 34, we introduce the following additional notations:

- Recall that $\mathcal{B}^{\mathrm{Det}}$ is the set of all deterministic, Markovian policies. We will use ν, ν' to denote members of $\mathcal{B}^{\mathrm{Det}}$ and $\nu_h(s)$ to denote the action ν takes at state s at step h when it is clear from the context.
- Let $\mathcal{B}^{\mathrm{E}}(s_{1:h})$ represent the subset of $\mathcal{B}^{\mathrm{Det}}$ that agrees with π^{E} on the state sequence $s_{1:h}$.
- Define $F(\nu;\nu';\pi^{\mathrm{E}}) := \sum_{s_{1:H}} \mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right]$, which evaluates the probability that ν' disagrees with π^{E} over the distribution of H-step state sequences induced by π.

Our key idea is to lower bound $\mathbb{E}^{\pi_u}\left[D^2_{\mathrm{H}}(\pi_u(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H}))\right]$, which reflects the asymmetric roles of the two appearances of π_u's, using a symmetric formulation via function F (as shown in (19) below).

Proof. Recall the first-step mixing policy in Definition 28, we start by rewriting

$$\begin{aligned}\rho(\pi_u \parallel \pi^{\mathrm{E}}) =& \mathbb{E}^{\pi_u}\left[\mathbb{I}\left\{\exists h : a_h \neq \pi_h^{\mathrm{E}}(s_h)\right\}\right] \\ =& \sum_{\nu \in \mathcal{B}^{\mathrm{Det}}} u(\nu) \sum_{s_{1:H}} \mathbb{P}^{\nu}(s_{1:H}, a_{1:H}) \mathbb{I}\left\{\exists h : a_h \neq \pi_h^{\mathrm{E}}(s_h)\right\} \\ =& \sum_{\nu \in \mathcal{B}^{\mathrm{Det}}} u(\nu) \rho(\nu \parallel \pi^{\mathrm{E}}),\end{aligned} \tag{16}$$

which is a weighted combination of $\rho(\nu \parallel \pi^{\mathrm{E}})$ for $\nu \in \mathcal{B}^{\mathrm{Det}}$.

Next, we turn to analyzing $D_{\mathrm{H}}^2(\pi_u(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H}))$. Since the deterministic expert induces a delta mass distribution over actions, we apply the elementary fact about the Hellinger distance with delta mass distribution stated in Lemma 38, yielding:

$$\frac{1}{2} \parallel \pi_u(\cdot\|s_{1:H}) - \pi^{\mathrm{E}}(\cdot\|s_{1:H}) \parallel_1 \leq D_{\mathrm{H}}^2(\pi_u(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H})).$$

We recall that $\mathcal{B}^{\mathrm{E}}(s_{1:H})$ denotes the subset of $\mathcal{B}^{\mathrm{Det}}$ that agrees with π^{E} on $s_{1:H}$ and define the total weight assigned by u on it as $u(\mathcal{B}^{\mathrm{E}}(s_{1:H})) := \sum_{\nu \in \mathcal{B}^{\mathrm{E}}(s_{1:H})} u(\nu)$. Then,

$$\frac{1}{2} \parallel \pi_u(\cdot\|s_{1:H}) - \pi^{\mathrm{E}}(\cdot\|s_{1:H}) \parallel_1 = 1 - u(\mathcal{B}^{\mathrm{E}}(s_{1:H})),$$

which implies:

$$1 - u(\mathcal{B}^{\mathrm{E}}(s_{1:H})) \leq D_{\mathrm{H}}^2(\pi_u(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H})). \tag{17}$$

Therefore, by taking expectation over $s_{1:H} \sim \mathbb{P}^{\pi_u}$ in Eq. (17),

$$\sum_{s_{1:H}} \mathbb{P}^{\pi_u}(s_{1:H})(1 - u(\mathcal{B}^{\mathrm{E}}(s_{1:H}))) \leq \mathbb{E}^{\pi_u}\left[D_{\mathrm{H}}^2(\pi_u(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H}))\right]. \tag{18}$$

We now examine the expression

$$\sum_{s_{1:H}} \mathbb{P}^{\pi_u}(s_{1:H})(1 - u(\mathcal{B}^{\mathrm{E}}(s_{1:H}))). \tag{$*$}$$

Since π_u is a first-step mixing of policies in $\mathcal{B}^{\mathrm{Det}}$ with weight u, we have $\mathbb{P}^{\pi_u}(s_{1:H}) = \sum_{\nu \in \mathcal{B}^{\mathrm{Det}}} u(\nu) \mathbb{P}^{\nu}(s_{1:H})$. This allows us to rewrite $(*)$ using the definition of $F(\nu; \nu', \pi^{\mathrm{E}})$ as:

$$\begin{aligned}(*) =& \sum_{s_{1:H}} \sum_{\nu \in \mathcal{B}^{\mathrm{Det}}} u(\nu) \mathbb{P}^{\nu}(s_{1:H}) \sum_{\nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu') \mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right] \\ =& \sum_{\nu, \nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu) u(\nu') \sum_{s_{1:H}} \mathbb{P}^{\nu}(s_{1:H}) \mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right] \\ =& \sum_{\nu, \nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu) u(\nu') F(\nu; \nu'; \pi^E) \\ =& \frac{1}{2} \sum_{\nu, \nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu) u(\nu') \left(F(\nu; \nu'; \pi^E) + F(\nu; \nu'; \pi^E)\right),\end{aligned} \tag{19}$$

where the first three equalities are by algebra and the definition of $F(\nu; \nu; \pi^E)$. In the last equality, we use the observation that

$$\sum_{\nu, \nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu) u(\nu') F(\nu; \nu'; \pi^E) = \sum_{\nu, \nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu) u(\nu') F(\nu'; \nu; \pi^E).$$

By Lemma 36 (stated below),

$$\begin{aligned}(*) \geq& \frac{1}{2} \cdot \frac{1}{2} \cdot \sum_{\nu, \nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu) u(\nu') \left(\rho(\nu \parallel \pi^{\mathrm{E}}) + \rho(\nu' \parallel \pi^{\mathrm{E}})\right) \\ =& \frac{1}{2} \cdot \sum_{\nu \in \mathcal{B}^{\mathrm{Det}}} u(\nu) \rho(\nu \parallel \pi^{\mathrm{E}}) = \frac{1}{2} \cdot \rho(\pi_u \parallel \pi^{\mathrm{E}}).\end{aligned}$$

Combining the above two inequalities with Eq (18) we conclude the proof by

$$\frac{1}{2} \cdot \rho(\pi_u \parallel \pi^{\mathrm{E}}) \leq (*) \leq \mathbb{E}^{\pi_u}\left[D^2_{\mathrm{H}}(\pi_u(\cdot\|s_{1:H}), \pi^{\mathrm{E}}(\cdot\|s_{1:H}))\right]. \qquad \square$$

Lemma 36 (Symmetric Evaluation Lemma)**.** *Given deterministic Markovian policies ν, ν', and π^{E}, the following holds:*

$$\frac{1}{2} \cdot \big(\rho(\nu \parallel \pi^{\mathrm{E}}) + \rho(\nu' \parallel \pi^{\mathrm{E}})\big) \leq F(\nu; \nu'; \pi^{\mathrm{E}}) + F(\nu'; \nu; \pi^{\mathrm{E}}). \tag{20}$$

Proof. Recall that

$$F(\nu; \nu'; \pi^{\mathrm{E}}) + F(\nu'; \nu; \pi^{\mathrm{E}}) = \sum_{s_{1:H}} \left(\mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right] + \mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left[\nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right]\right).$$

Throughout the proof, we say that ν makes a *mistake* at step h, if $\nu_h(s_h) \neq \pi^{\mathrm{E}}_h(s_h)$. Then, we can partition all state sequences $s_{1:H} \in \mathcal{S}^H$ into 4 subsets, $\mathcal{X}_i$, indexed by $i \in \{1, 2, 3, 4\}$:

1. $\mathcal{X}_1 := \{s_{1:H} \mid \nu, \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:H})\}$;
2. $\mathcal{X}_2 := \{s_{1:H} \mid \exists h, s.t. \nu \in \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1})\}$;
3. $\mathcal{X}_3 := \{s_{1:H} \mid \exists h, s.t. \nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1})\}$;
4. $\mathcal{X}_4 := \{s_{1:H} \mid \exists h, s.t. \nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1}), \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1})\}\}$.

In words, the four subsets divide state sequences into cases where: (1) both ν, ν' agree with the π^{E} throughout, (2)&(3) one of ν, ν' makes its first mistake earlier than the other, and (4) ν, ν' make their first mistake at the same time. It can now be easily seen that each $s_{1:H} \in \mathcal{S}^H$ lies in exactly one of such $\mathcal{X}_i$, and

$$\mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{X}_3 \cup \mathcal{X}_4 = \mathcal{S}^H.$$

To see this, consider h^{err}, the first time step h such that one of ν and ν' disagree with π^{E}. If h^{err} does not exist, then $s_{1:H} \in \mathcal{X}_1$. Otherwise, $s_{1:H}$ lies in one of $\mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4$ depending on whether ν and ν' makes mistakes at step h^{err}.

By definition, subset $\mathcal{X}_1$ denotes trajectories $s_{1:H}$ where $\nu, \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:H})$, meaning that

$$\sum_{s_{1:H} \in \mathcal{X}_1} \left(\mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right] + \mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left[\nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right]\right) = 0\,.$$

For the other 3 sets, i.e. $\mathcal{X}_i$ for $i \in \{2, 3, 4\}$, we can further divide each set based on the time step where the first error occurs, formally:

$$\begin{aligned}
\mathcal{X}_2^h :=& \{s_{1:H} \mid \nu \in \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1})\};\\
\mathcal{X}_3^h :=& \{s_{1:H} \mid \nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1})\};\\
\mathcal{X}_4^h :=& \{s_{1:H} \mid \nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1}), \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1})\}.
\end{aligned} \tag{21}$$

By definition, each pair of subsets is disjoint and $\cup_{h\in[H]}\mathcal{X}_i^h = \mathcal{X}_i$, for $i = 2, 3, 4$. Note that the determination of whether $s_{1:H} \in \mathcal{X}_i^h$ only depends on $s_{1:h}$; therefore, $\mathcal{X}_i^h$ can be represented as $\tilde{\mathcal{X}}_i^h \times \mathcal{S}^{H-h}$, where

$$\tilde{\mathcal{X}}_i^h := \{s_{1:h} \mid s_{1:H} \in \mathcal{X}_i^h\}.$$

Based on this observation, we have

$$\sum_{s_{1:H} \in \mathcal{X}_i^h} \mathbb{P}^{\nu}(s_{1:H}) = \sum_{s_{1:h} \in \tilde{\mathcal{X}}_i^h, s_{h+1:H} \in \mathcal{S}^{H-h}} \mathbb{P}^{\nu}(s_{1:H}) = \sum_{s_{1:h} \in \tilde{\mathcal{X}}_i^h} \mathbb{P}^{\nu}(s_{1:h}).$$

Furthermore, since deterministic policies $\nu, \nu', \pi^{\mathrm{E}}$ agrees with each other for all $\{s_{1:h-1} | s_{1:h} \in \tilde{\mathcal{X}}_i^h\}$,

$$\begin{aligned}\sum_{s_{1:h}\in\tilde{\mathcal{X}}_i^h}\mathbb{P}^{\nu}(s_{1:h}) &= \sum_{s_{1:h}\in\tilde{\mathcal{X}}_i^h}\rho(s_1)\prod_{h'=1}^{h-1}P_h(s_{h+1}|s_h,\nu_h(s_h)) \\ &= \sum_{s_{1:h}\in\tilde{\mathcal{X}}_i^h}\rho(s_1)\prod_{h'=1}^{h-1}P_h(s_{h+1}|s_h,\nu'_h(s_h)) = \sum_{s_{1:h}\in\tilde{\mathcal{X}}_i^h}\mathbb{P}^{\nu'}(s_{1:h}).\end{aligned} \tag{22}$$

This implies that

$$\sum_{s_{1:H}\in\mathcal{X}_i^h}\mathbb{P}^{\nu}(s_{1:H}) = \sum_{s_{1:H}\in\mathcal{X}_i^h}\mathbb{P}^{\nu'}(s_{1:H}),$$

and therefore, summing over all $h\in[H]$,

$$\sum_{s_{1:H}\in\mathcal{X}_i}\mathbb{P}^{\nu}(s_{1:H}) = \sum_{s_{1:H}\in\mathcal{X}_i}\mathbb{P}^{\nu'}(s_{1:H}).$$

Now, for $\mathcal{X}_2$, we have

$$\sum_{s_{1:H}\in\mathcal{X}_2}\left(\mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left[\nu'\notin\mathcal{B}^{\mathrm{E}}(s_{1:H})\right]+\mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left[\nu\notin\mathcal{B}^{\mathrm{E}}(s_{1:H})\right]\right)\geq\sum_{s_{1:H}\in\mathcal{X}_2}\mathbb{P}^{\nu}(s_{1:H})\,, \tag{23}$$

where we apply the fact that for all $s_{1:H}\in\mathcal{X}_2$, $\nu'\notin\mathcal{B}^{\mathrm{E}}(s_{1:H})$, and dropping the second term which is nonnegative.

Similarly, for $\mathcal{X}_3$, we have that

$$\sum_{s_{1:H}\in\mathcal{X}_3}\left(\mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left[\nu'\notin\mathcal{B}^{\mathrm{E}}(s_{1:H})\right]+\mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left[\nu\notin\mathcal{B}^{\mathrm{E}}(s_{1:H})\right]\right)\geq\sum_{s_{1:H}\in\mathcal{X}_3}\mathbb{P}^{\nu'}(s_{1:H})=\sum_{s_{1:H}\in\mathcal{X}_3}\mathbb{P}^{\nu}(s_{1:H}). \tag{24}$$

Finally, for $\mathcal{X}_4$, we use the fact that for $s_{1:H}\in\mathcal{X}_4$, $\nu,\nu'\notin\mathcal{B}^{\mathrm{E}}(s_{1:H})$ and obtain

$$\begin{aligned}&\sum_{s_{1:H}\in\mathcal{X}_4}\left(\mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left[\nu'\notin\mathcal{B}^{\mathrm{E}}(s_{1:H})\right]+\mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left[\nu\notin\mathcal{B}^{\mathrm{E}}(s_{1:H})\right]\right) \\ &=\sum_{s_{1:H}\in\mathcal{X}_4}\left(\mathbb{P}^{\nu}(s_{1:H})+\mathbb{P}^{\nu'}(s_{1:H})\right)\geq\sum_{s_{1:H}\in\mathcal{X}_4}\mathbb{P}^{\nu}(s_{1:H}).\end{aligned} \tag{25}$$

Now, we combine Eqs. (23), (24), (25) and observe that

$$\sum_{s_{1:H}\in\mathcal{X}_2}\mathbb{P}^{\nu}(s_{1:H})+\sum_{s_{1:H}\in\mathcal{X}_3}\mathbb{P}^{\nu}(s_{1:H})+\sum_{s_{1:H}\in\mathcal{X}_4}\mathbb{P}^{\nu}(s_{1:H})\geq\frac{1}{2}\sum_{s_{1:H}\in\mathcal{X}_2\cup\mathcal{X}_3\cup\mathcal{X}_4}\left(\mathbb{P}^{\nu}(s_{1:H})+\mathbb{P}^{\nu'}(s_{1:H})\right), \tag{26}$$

which implies

$$F(\nu;\nu';\pi^{\mathrm{E}})+F(\nu';\nu;\pi^{\mathrm{E}})\geq\frac{1}{2}\sum_{s_{1:H}\in\mathcal{X}_2\cup\mathcal{X}_3\cup\mathcal{X}_4}\left(\mathbb{P}^{\nu}(s_{1:H})+\mathbb{P}^{\nu'}(s_{1:H})\right). \tag{27}$$

Based on the definitions of $\mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4$ and $\rho(\cdot \parallel \cdot)$,

$$\begin{aligned}\sum_{s_{1:H}\in\mathcal{X}_2\cup\mathcal{X}_3\cup\mathcal{X}_4}\left(\mathbb{P}^{\nu}(s_{1:H})+\mathbb{P}^{\nu'}(s_{1:H})\right) &= \sum_{s_{1:H}}\mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left\{\exists h:\nu_h(s_h)\neq\pi^{\mathrm{E}}_h(s_h) \text{ or } \nu'_h(s_h)\neq\pi^{\mathrm{E}}_h(s_h)\right\}\\ &\quad+\sum_{s_{1:H}}\mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left\{\exists h:\nu_h(s_h)\neq\pi^{\mathrm{E}}_h(s_h) \text{ or } \nu'_h(s_h)\neq\pi^{\mathrm{E}}_h(s_h)\right\}\\ &\geq\sum_{s_{1:H}}\mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left\{\exists h:\nu_h(s_h)\neq\pi^{\mathrm{E}}_h(s_h)\right\}\\ &\quad+\sum_{s_{1:H}}\mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left\{\exists h:\nu'_h(s_h)\neq\pi^{\mathrm{E}}_h(s_h)\right\}\\ &=\rho(\nu\parallel\pi^{\mathrm{E}})+\rho(\nu'\parallel\pi^{\mathrm{E}}),\end{aligned} \tag{28}$$

where $s_{1:H}\in\mathcal{X}_2\cup\mathcal{X}_3\cup\mathcal{X}_4$ implies either ν or ν' disagrees with π^{E}, while the inequality relaxes the condition by splitting it into separate contributions for ν and ν'.

We conclude the proof by plugging (28) into (27). □

E.4 Proof of Theorem 31

We first demonstrate that the performance difference between expert and the the uniform first-step mixing of any Markovian policy sequence $\{\pi^n\}_{n=1}^N$ is upper bounded by $2R\,\mathrm{OnEst}_N^{\mathrm{Traj}}/N$, and then show the trajectory-wise sample complexity of Algorithm 3 in Theorem 31.

Lemma 37. *For any MDP $\mathcal{M}$, deterministic expert π^{E}, and sequence of policies $\{\pi^n\}_{n=1}^N$, each of which can be Markovian or a first-step mixing of Markovian policies, their first step uniform mixture policy $\hat{\pi}$ satisfies.*

$$J(\pi^{\mathrm{E}})-J(\hat{\pi})\leq 2R\cdot\frac{\mathrm{OnEst}_N^{\mathrm{Traj}}}{N}.$$

Proof. By Lemma 33, for each π^n, which is a first-step mixing of Markovian policies:

$$\mathbb{E}^{\pi^n}\left[D^2_{\mathrm{H}}(\pi^n(\cdot\|s_{1:H}),\pi^{\mathrm{E}}(\cdot\|s_{1:H}))\right]\geq\frac{1}{2}\rho(\pi^n\parallel\pi^{\mathrm{E}}).$$

Then, by the definition of $\mathrm{OnEst}_N^{\mathrm{Traj}}$,

$$\frac{\mathrm{OnEst}_N^{\mathrm{Traj}}}{N}=\frac{1}{N}\sum_{n=1}^N\mathbb{E}^{\pi^n}\left[D^2_{\mathrm{H}}(\pi^n(\cdot\|s^n_{1:H}),\pi^{\mathrm{E}}(\cdot\|s^n_{1:H}))\right]\geq\frac{1}{2N}\sum_{n=1}^N\rho(\pi^n\parallel\pi^{\mathrm{E}})=\frac{1}{2}\rho(\hat{\pi}\parallel\pi^{\mathrm{E}}),$$

where we apply the fact that $\hat{\pi}$ is a first-step uniform mixing of $\{\pi^n\}_{n=1}^N$. Finally, we conclude the proof by applying Lemma 40. □

Proof of Theorem 31. The proof closely follows Proposition C.2 in [17],which was tailored to another variant of DAgger. Different from that analysis, here we leverage the distribution of the state sequence $s_{1:H}$ instead of the per-step state distribution.

Observe that the log loss functions passed through online learning oracle $\mathbb{A}$, $\ell^n(\pi)$ is of the form

$$\ell^n(\pi)=\log\left(\frac{1}{\pi(a^{n,*}_{1:H}\parallel s^n_{1:H})}\right)$$

It can be observed that $\ell^n(\pi_u)$'s are 1-exp-concave in $u\in\Delta(\mathcal{B})$. Therefore, setting $\mathbb{A}$ as the exponential weight algorithm (Proposition 41) ensures that the following bound holds almost surely:

$$\sum_{n=1}^N\log(1/\pi^n(a^{*,n}_{1:H}\parallel s^n_{1:H}))\leq\sum_{n=1}^N\log(1/\pi^{\mathrm{E}}(a^{*,n}_{1:H}\parallel s^n_{1:H}))+\log(B)=\log(B).$$

Now, Lemma 42 with $x^n = s^n_{1:H}$, $y^n = a^{*,n}_{1:H}$, $g_* = \pi^{\mathrm{E}}(\cdot\|\cdot)$, and $\mathcal{H}^n = \{o^{n'}\}_{n'=1}^n$, where $o^n = \left(s^n_1, a^n_1, a^{*,n}_1, \ldots, s^n_H, a^n_H, a^{*,n}_H\right)$, implies that with probability at least $1-\delta$,

$$\mathrm{OnEst}^{\mathrm{Traj}}_N = \sum_{n=1}^N \mathbb{E}^{\pi^n}\left[D^2_{\mathrm{H}}(\pi^n(\cdot\|s^n_{1:H}), \pi^{\mathrm{E}}(\cdot\|s^n_{1:H}))\right] \le \log(B) + 2\log(1/\delta).$$

The second part of the theorem follows by applying Lemma 37. □

F Auxiliary Results

Lemma 38. *If p, q are two distributions over some discrete domain $\mathcal{Z}$, and q is* a delta mass *on an element in $\mathcal{Z}$. Then*

$$\frac{1}{2} \parallel p - q \parallel_1 \le D^2_H(p,q) \le \parallel p - q \parallel_1$$

Proof. Without loss of generality, assume that q is a delta mass on z_0. Therefore, $\|p-q\|_1 = 2(1-p(z_0))$, and

$$D^2_H(p,q) = (1-\sqrt{p(z_0)})^2 + (1-p_{z_0}) = 2(1-\sqrt{p(z_0)}).$$

Thus,

$$\frac{D^2_H(p,q)}{\|p-q\|_1} = \frac{1}{1+\sqrt{p(z_0)}} \in [\frac{1}{2}, 1].$$

□

Lemma 39 (Performance Difference Lemma [23, 49])**.** *For two Markovian policies π and π^{E} : $\mathcal{S} \to \Delta(\mathcal{A})$, we have*

$$J(\pi^{\mathrm{E}}) - J(\pi) = \mathbb{E}^{\pi}\left[\sum_{h=1}^H A^E_h(s_h, a_h)\right],$$

where $A^E_h(s_h,a_h) := Q^{\pi^{\mathrm{E}}}_h(s_h,a_h) - V^{\pi^{\mathrm{E}}}_h(s_h)$. Furthermore:

- *It holds that (recall Definition 9)*
$$J(\pi) - J(\pi^E) \le H \cdot \lambda(\pi^{\mathrm{E}} \parallel \pi).$$
- *Suppose $(\mathcal{M}, \pi^{\mathrm{E}})$ is μ-recoverable, then*
$$J(\pi) - J(\pi^E) \le \mu \cdot \lambda(\pi \parallel \pi^{\mathrm{E}}).$$

Lemma 40 (Lemma D.2. of [17])**.** *For all (potentially stochastic) policies π and π', it holds that*

$$J(\pi) - J(\pi') \le R \cdot \rho(\pi \parallel \pi').$$

Proposition 41 (Proposition 3.1 of [9])**.** *Suppose $\{\ell^n(\cdot)\}_{n=1}^N$ is a sequence of η-exp-concave functions from $\Delta(\mathcal{X})$ to $\mathbb{R}$. For all $x \in \mathcal{X}$, define the weights $w^{n-1}(x)$ and probabilities $u^n(x)$ as follows:*

$$w^{n-1}(x) = e^{-\eta\sum_{i=1}^{n-1}\ell_i(e_x)}, \quad u^n(x) = \frac{w^{n-1}(x)}{\sum_{x'\in\mathcal{X}} w^{n-1}(x')},$$

where e_x is the x-th standard basis vector in $\mathbb{R}^{|\mathcal{X}|}$. Then, choosing $u^n = (u^n(x))_{x\in\mathcal{X}}$ (exponential weights used with learning rate η) satisfies:

$$\sum_{n=1}^N \ell^n(u^n) \le \min_{x\in\mathcal{X}} \sum_{n=1}^N \ell^n(e_x) + \frac{\log|\mathcal{X}|}{\eta}.$$

Lemma 42 (Corollary of Lemma A.14 in [18])**.** *Under the realizbility assumption, where there exists $g_\star \in \mathcal{G}$ such that for all $n \in [N]$,*

$$y^n \mid x^n, \mathcal{H}^{n-1} \sim g_\star(\cdot \mid x^n),$$

where $\mathcal{H}^{n-1}$ denotes all histories at the beginning of round n. Then, for any estimation algorithm that outputs $\{\hat{g}^n\}_{n=1}^N$ online and any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\sum_{n=1}^N \mathbb{E}_{n-1}\left[D^2_{\mathcal{H}}\left(\hat{g}^n(\cdot \mid x^n), g_\star(\cdot \mid x^n)\right)\right] \le \sum_{n=1}^N \left(\log\frac{1}{\hat{g}^n(y^n \mid x^n)} - \log\frac{1}{g_\star(y^n \mid x^n)}\right) + 2\log(\delta^{-1}).$$

where $\mathbb{E}_n[\cdot] := \mathbb{E}[\cdot \mid \mathcal{H}^n]$.

G Experiment Details

We compare WARM-STAGGER with Behavior Cloning (BC) and STAGGER on continuous-control tasks from OpenAI Gym MuJoCo [72, 8] with episode length $H = 1000$.

Infrastructure and Implementation. All experiments were conducted on a Linux workstation equipped with an Intel Core i9 CPU (3.3GHz) and four NVIDIA GeForce RTX 2080 Ti GPUs. Our implementation builds on the publicly available DRIL repository [7] (https://github.com/xkianteb/dril), with modifications to support interactive learning. The continuous control environments used in our experiments are: "HalfCheetahBulletEnv-v0", "AntBulletEnv-v0", "Walker2DBulletEnv-v0", and "HopperBulletEnv-v0". We include link to our implementation here: `https://github.com/liyichen1998/Interactive-and-Hybrid-Imitation-Learning-Provably-Beating-Behavior-Cloning`.

Environments and Expert Policies. We use four MuJoCo environments: Ant, Hopper, HalfCheetah, and Walker2D. The expert policy is a deterministic MLP pretrained via TRPO [52, 53], with two hidden layers of size 64.

Model Architecture used by Learner. The learner uses the same MLP architecture as the expert. Following [17], we use a diagonal Gaussian policy:

$$\pi(a \mid s) = \mathcal{N}\left(f_\theta(s), \mathrm{diag}(\sigma^2)\right),$$

where $f_\theta(s) \in \mathbb{R}^{d_\mathcal{A}}$ is the learned mean, and $\sigma \in \mathbb{R}^{d_\mathcal{A}}$ is a learnable log-standard deviation vector.

Each model is trained from random initialization using a batch size of 100, a learning rate of 10^{-3}, and up to 2000 passes over the dataset, with early stopping evaluated every 250 passes using a 20% held-out validation set.

Learning Protocols. To evaluate the performance of BC against the number of states annotated, we reveal expert state-action pairs sequentially along expert trajectories until the annotation budget is reached. For STAGGER, at each iteration, it rolls out the latest policy, samples a state uniformly from the trajectory, queries it for the expert action, and updates immediately.

For WARM-STAGGER, we begin with BC and switch to STAGGER after a predefined number of offline expert state-action pairs has been used, denoted as N. We set N to be 100, 200, or 400 for easier tasks (e.g., Hopper, Ant) and 200, 400, or 800 for harder tasks (e.g., HalfCheetah, Walker2D).

Cost Model and Evaluation. We assign a cost of 1 to each offline state-action pair and a cost of $C = 1$ or 2 to each interactive query. We run each method for 10 random seeds. For every 50 new state-action pairs collected, we evaluate the current policy by running 25 full-episode rollouts and reporting the average return.

Though the nonrealizable setting is beyond the scope of this work, we expect that some variant of our algorithm can still give reasonable performance, provided that the policy class is expressive enough (so that the approximation error is nonzero but small). For example, [34] observed that with nonrealizable stochastic experts, DAgger variants outperform BC, and exhibit learning curves similar to ours.

G.1 Additional Experiment Plots

We present extended experiment results with larger cost budgets. As shown in Figure 4, we allocate a total annotation cost budget of 2000 for Hopper and Ant, and 4000 for HalfCheetah and Walker. This complements Figure 3 in the main paper by showing the full training curves without zooming into the stage with small cost budget. The trends are consistent with our earlier observations: WARM-STAGGER achieves similar or better sample efficiency compared to STAGGER when $C = 1$, and clearly outperforms both baselines under the cost-aware setting where $C = 2$.

G.2 Experiment with MSE Loss

We additionally evaluate our algorithms using mean squared error (MSE) as the loss function for optimization. All training settings remain identical to the main experiments with log loss, except that we use a learning rate of 2.5×10^{-4}. As shown in Figure 5, we observe qualitatively similar results to those under log loss shown in Figure 3, consistent with prior observations in [17], with perhaps more stable learning curves.

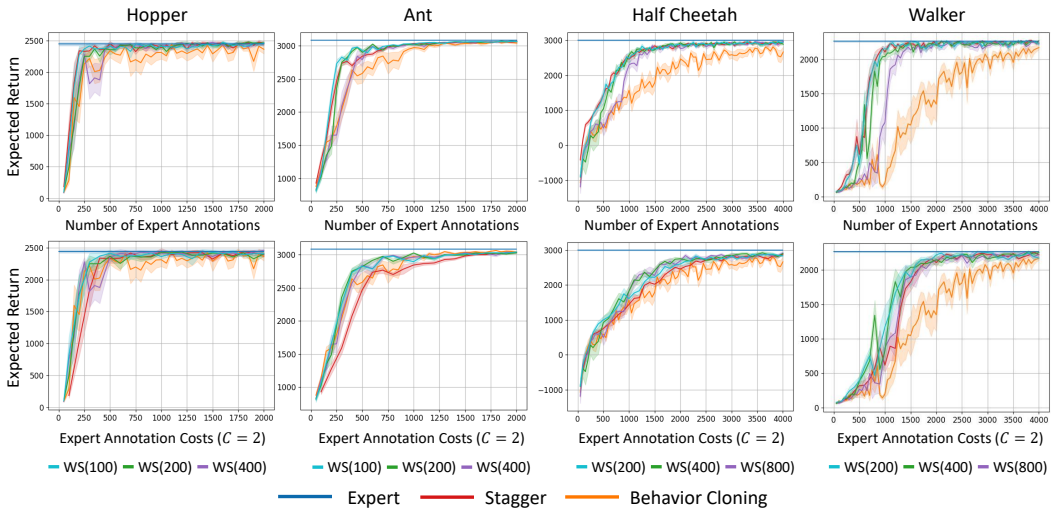

Figure 4: Sample and cost efficiency on MuJoCo tasks. The top row shows expected return vs. number of annotations ($C = 1$); the bottom row shows performance under a cost-aware setting ($C = 2$). WARM-STAGGER (WS) is initialized with 1/20, 1/10, or 1/5 of the samples as offline demonstrations. It matches STAGGER in sample efficiency and outperforms the baselines when $C = 2$, especially WS(1/5).

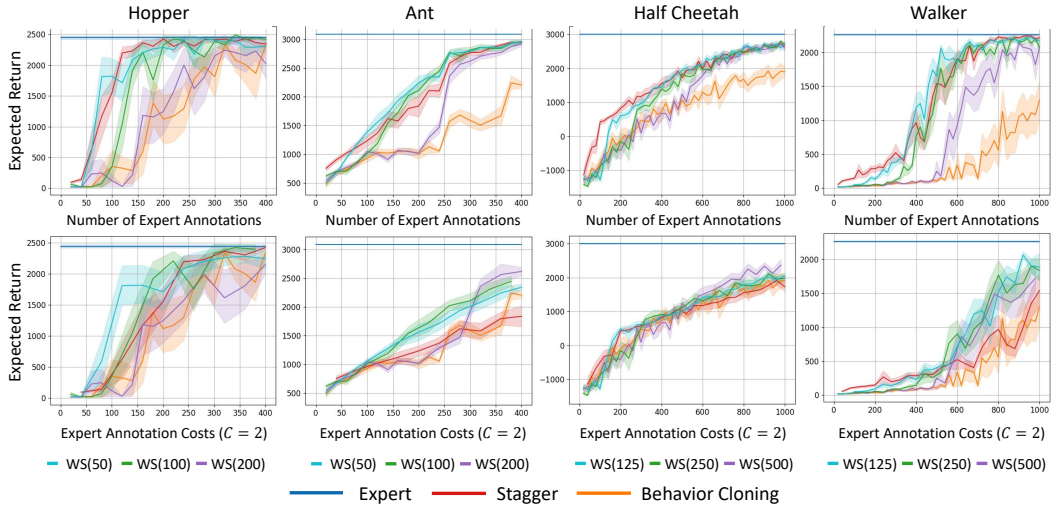

Figure 5: Performance comparison under MSE loss across MuJoCo tasks. Results show that WARM-STAGGER (WS) achieves comparable sample efficiency and performance to the log loss setting, with improved training stability. Each curve represents the average over 10 seeds.

G.3 Additional Experiments with TRAGGER

For completeness, we evaluate TRAGGER and its warm-start variant (WARM-TRAGGER, Algorithm 4), on continuous control tasks and the toy MDP as in Figure 2. The key distinction between WARM-TRAGGER and WARM-STAGGER lies in the annotation mode: the former employs trajectory-wise oracle feedback instead of state-wise annotation, leading to notably different behaviors, as shown in Figure 6.

In particular, for Ant and HalfCheetah, the state-wise cost efficiency ($C = 1$) of TRAGGER and WARM-TRAGGER is significantly worse than that of STAGGER due to the cold-start problem: early

Algorithm 4 WARM-TRAGGER: Warm-start TRAGGER with offline demonstrations

Input: MDP $\mathcal{M}$, trajectory-wise expert annotation oracle $\mathcal{O}^{\mathrm{Traj}}$, Stationary policy class $\mathcal{B}$, online learning oracle $\mathbb{A}$, offline expert dataset D_{off} of size N_{off}, interaction budget (in terms of number of states) N_{int}
Initialize $\mathbb{A}$ with policy class $\mathcal{B}_{\mathrm{bc}} := \{\pi \in \mathcal{B} : \pi(s_h) = a_h,\ \forall h \in [H], \forall (s,a)_{1:H} \in D_{\mathrm{off}}\}$
for $n = 1, \dots, N_{\mathrm{int}}/H$ **do**
    Query $\mathbb{A}$ and receive π^n.
    Execute π^n and sample $s^n_{1:H}$ following $\mathbb{P}^{\pi^n}$. Query $\mathcal{O}^{\mathrm{Traj}}$ for $a^{*,n}_{1:H} = \pi^{\mathrm{E}}(s^n_{1:H})$.
    Update $\mathbb{A}$ with loss function

$$\ell^n(\pi) := \log\left(\frac{1}{\pi(a^{*,n}_{1:H} \parallel s^n_{1:H})}\right). \tag{29}$$

end for
Output: $\hat{\pi}$, a first-step uniform mixture of $\{\pi^1, \dots, \pi^N\}$.

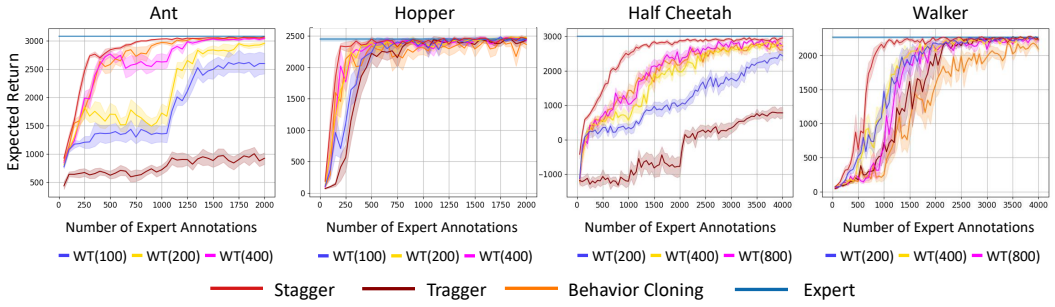

Figure 6: Sample efficiency of algorithms on MuJoCo tasks, showing expected return vs. number of annotations ($C = 1$). WARM-TRAGGER (WT) is initialized with 1/20, 1/10, or 1/5 of the total annotation budget as offline demonstrations. Specifically, WT(n) refers to WT with offline demonstrations of total length n. Although the performance of WT improves with more offline demonstrations, both TRAGGER and WARM-TRAGGER remain inferior to STAGGER and, in many cases, even underperform Behavior Cloning, confirming the advantage of state-wise over trajectory-wise annotations.

DAgger rollouts have poor state coverage but must still proceed until the end of each trajectory. In contrast, STAGGER samples only one state per trajectory, thereby better leveraging interaction to get timely feedback. For Hopper and Walker, however, TRAGGER and WARM-TRAGGER achieve performance closer to STAGGER. These environments feature hard resets when the agent's state becomes unhealthy (unlikely in Ant and never in HalfCheetah), which terminates ('truncate' has a specific meaning of ending at a prespecified step in openai gym) poor trajectories and consequently improve sample efficiency.

Overall, these observations suggest a natural middle ground between full-trajectory and single-state annotation—namely, batch queries (e.g., sampling 50 states per trajectory), as explored by [34] with comparable results.

A head-to-head comparison between TRAGGER and STAGGER, as well as between WARM-TRAGGER and WARM-STAGGER, is shown in Figure 7, highlighting the advantage of state-wise over trajectory-wise annotation.

However, this advantage does not hold in general: in the toy MDP in Figure 2, TRAGGER and WARM-TRAGGER achieve performance nearly identical to STAGGER and WARM-STAGGER, as shown in Figure 8.

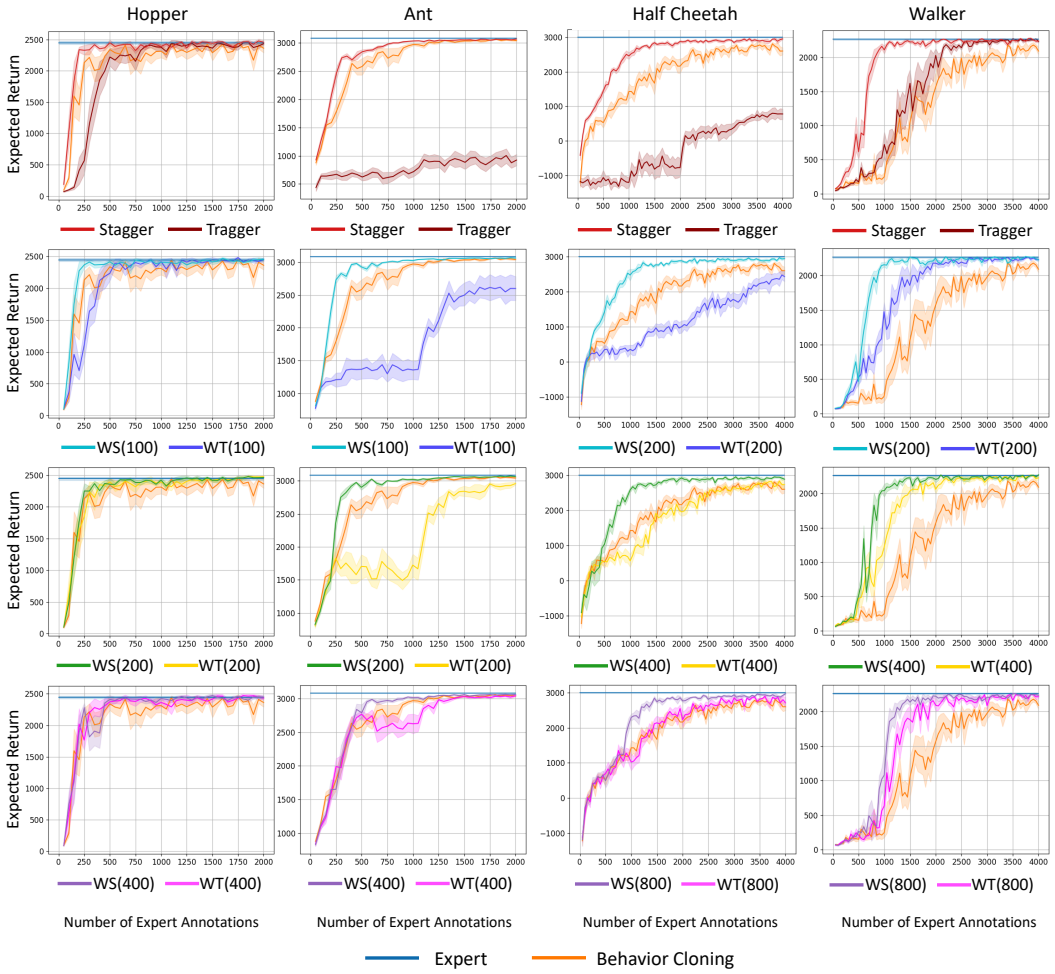

Figure 7: Head-to-head sample efficiency comparison between TRAGGER and STAGGER, and between WARM-TRAGGER and WARM-STAGGER under equal (since we are talking about comparison here) offline demonstration budgets. STAGGER and WARM-STAGGER consistently outperform TRAGGER and WARM-TRAGGER. The performance gap narrows as the offline budget increases, effectively alleviating the cold-start problem suffered by TRAGGER.

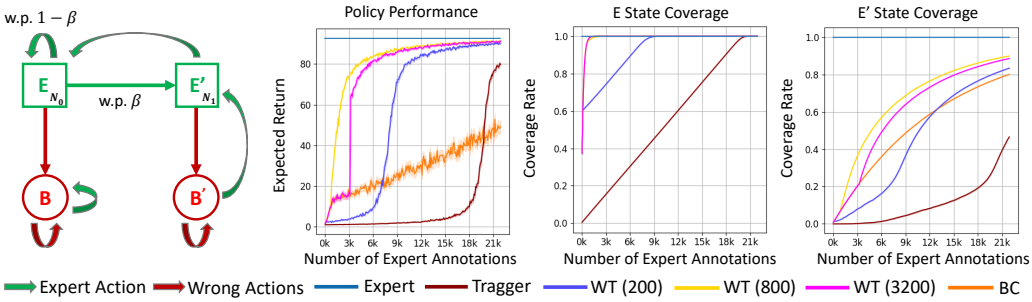

Figure 8: Similar to Figure 2, we evaluate TRAGGER and WARM-TRAGGER (WT) with 200, 800, 3200 offline (state, expert action) pairs in the toy MDP therein. All methods are evaluated under equal total annotation cost with $C = 1$. With 800 offline (state, expert action) pairs, WT significantly improves the sample efficiency over the baselines and explores $\mathbf{E}'$ more effectively. The performance of TRAGGER and WARM-TRAGGER is almost the same as STAGGER and WARM-STAGGER in Figure 2.