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 training 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, or learning from demonstrations, is a widely applied and practical paradigm for learning sequential decision-making policies [43, 4, 3]. 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 [40, 64].

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 [49, 48, 64]. The most basic and well-known approach for offline imitation learning is Behavior Cloning [47, 15], 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 is known to be limited [43]. 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 [15], via a sharp theoretical 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 [47] can enjoy sample complexity benefits, but so far the benefits are only exhibited in limited examples, with the most general examples in the tabular setting [44]. This leaves open the question:

> *Can interaction provide sample efficiency benefit for imitation learning under a broad range of settings, especially with 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 [26, 34], 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 [49] in the environment, 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 has the additional ability to 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 compunding error [43] and cold start problem [32, 39], two practical challenges in imitation learning, 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 simulation results that verify this theoretical 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 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. An episodic MDP $\mathcal{M}$ is defined as a tuple $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, H)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P := \{P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})\}_{h=1}^H$ denotes the transition dynamics, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \Delta([0, 1])$ denotes the reward distribution, and H denotes episode length. A Markovian policy (policy) is a collection of H mappings from states to probability distributions over actions $\pi = \{\pi_h : \mathcal{S} \to \Delta(\mathcal{A})\}_{h=1}^H$. A policy induces a distribution over trajectories $\tau = (s_h, a_h, r_h)_{h=1}^H$ by first drawing the initial state $s_1 \sim P_0(\varnothing)$, and then iteratively taking actions $a_h \sim \pi_h(s_h)$, receiving rewards $r_h \sim \mathcal{R}(s_h, a_h)$, and transitioning to the next state $s_{h+1} \sim P_h(s_h, a_h)$ (except at step H, where $P_H = \varnothing$). 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 $d^\pi(s, h) := \frac{1}{H}\mathbb{P}^\pi(s_h = s)$ its (state, step) visitation distribution, and $d^\pi(h) := \frac{1}{H}$. 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 $Q_h^\pi(s, a) := \mathbb{E}^\pi \left[\sum_{h'=h}^H r_{h'} \mid s_h = s, a_h = a\right]$.

Additional policy-related notations. Throughout, we assume the access to an Markovian policy class $\mathcal{B}$ of finite size B, which contains the deterministic expert demonstrator policy $\pi^{\mathrm{E}} := \{\pi_h^{\mathrm{E}} : \mathcal{S} \to \mathcal{A}\}_{h=1}^H$. A $(\mathrm{MDP}, \mathrm{Expert})$ pair $(\mathcal{M}, \pi^{\mathrm{E}})$ is said to be μ-recoverable if for all $h \in [H]$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $Q_h^{\pi^{\mathrm{E}}}(s,a) - V_h^{\pi^{\mathrm{E}}}(s) \le \mu$. Additionally, we assume normalized return [15], where for any trajectory, $\sum_{h=1}^H r_h \in [0, R]$. 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 of $\mathcal{B}$)**.** $\bar{\Pi}_{\mathcal{B}} = \{\bar{\pi}_{u,h}(a|s) := \sum_{\pi \in \mathcal{B}} u(\pi)\pi_h(a|s) : u \in \Delta(\mathcal{B})\}$.

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 takes action $a \sim \pi_h(\cdot|s)$ (e.g. [30, 31]). Observe that $\bar{\pi}_u$ is a Markovian policy.

Offline imitation learning and Behavior Cloning. In offline imitation learning, the agent is given a collection of expert trajectories $\mathcal{D} = \{\tau_1, \ldots, \tau_{N_{\text{off}}}\}$, 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 a policy $\pi \in \mathcal{B}$ that minimizes its disagreement with expert's actions on the seen states, i.e.,

$$\hat{\pi} = \operatorname*{argmin}_{\pi \in \mathcal{B}} \sum_{i=1}^{N_{\text{off}}} \sum_{h=1}^{H} I(\pi_h(s_{i,h}) \neq a_{i,h}),$$

where $I(\cdot)$ is the indicator function. Recent result of [15] establishes a horizon-independent analysis of Behavior Cloning, which we recall its guarantees here:

Theorem 2 (Guarantee of BC [15])**.** *Suppose Assumption* **??** *holds, then with probability $1 - \delta$, the policy returned by BC $\hat{\pi}$ satisfies:*

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

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* $\mathcal{O}^{\mathrm{Traj}}$ [49, 15]: given a state sequence $(s_h)_{h=1}^H$, return $(a_h)_{h=1}^H$ such that $a_h = \pi_h^{\mathrm{E}}(s_h)$ for all h. Subsequent works have considered modeling the expert as a *state-wise demonstration oracle* [20, 5, 37, 53] $\mathcal{O}^{\mathrm{State}}$: given a state s_h and step h, return $a_h = \pi_h^{\mathrm{E}}(s_h)$. 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 rollout in $\mathcal{M}$, observing 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 [26]. 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 to be more expensive than obtaining an additional offline (state, expert action) pair. We denote the ratio between these two costs as C, where $C \ge 1$ is an application-dependent constant. [1]

[1]For practical settings such as human-in-the-loop learning with expert interventions [34, 61], obtaining a short segment of corrective demonstrations may be cheaper than querying $\mathcal{O}^{\mathrm{State}}$ for each state therein individually. Here, we focus on a simplified setting and leave detailed cost modeling for such settings as interesting future work.

Algorithm 1 STAGGER: DAgger with State-wise annotation oracle

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

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

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

3 State-wise Annotation in Interactive Imitation Learning

Recent work [15] on refined analysis of Behavior Cloning (BC) casts doubt in the utility of interaction in imitation learning: when measuring sample complexity in the number of trajectories annotated, BC is shown to provide guarantees no worse than interactive approaches. Although benefits of interactive approaches have been shown in specific examples, progresses so far have been relatively sparse [15, 44], with the most general results in the tabular setting [44]. In this section, we show that interaction benefits imitation learning in a general sense: when measuring sample complexity using the number of state-wise annotations, we design an interactive algorithm that achieves a lower sample complexity than BC, as long as the expert 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 $\mathcal{O}^{\text{State}}$. Similar to the original DAgger [49], it requires base policy class $\mathcal{B}$ and reduces interactive imitation learning to no-regret online learning. At round n, it rolls out the current policy π^n obtained from an online learning oracle $\mathbb{A}$ and samples (s^n, h^n) from d^{π^n}. A classical example of $\mathbb{A}$ is the exponential weight algorithm that chooses policies from $\bar{\Pi}_{\mathcal{B}}$ ([7]; see Appendix F). It then queries $\mathcal{O}^{\text{State}}$ 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 $\{\pi^n\}_{n=1}^{N_{\text{int}}}$, i.e., sample one π^n uniformly at random and execute it for the episode. In contrast to the DAgger variant analyzed in [15], which trains a distinct policy at each step—yielding H policies in total—and employs trajectory-level annotations, our algorithm utilizes parameter sharing and uses state-wise annotations.

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 $\mathcal{O}^{\text{State}}$, an MDP $\mathcal{M}$ where $(\mathcal{M}, \pi^{\text{E}})$ is μ-recoverable, a policy class $\mathcal{B}$ such that deterministic realizability (Assumption* **??***) holds, and the online learning oracle $\mathbb{A}$ set as the exponential weight algorithm with decision space $\Delta(\mathcal{B})$ and returns each-step mixing policies $\bar{\pi}_u \in \bar{\Pi}_{\mathcal{B}}$. Then it returns $\hat{\pi}$ such that, with probability at least $1 - \delta$,*

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

Theorem 3 shows that STAGGER returns a policy of suboptimality $O(\frac{\mu H \log B}{N_{\text{int}}})$ using N_{int} interactive state-wise annotations from the expert. In comparison, with the cost of N_{int} state-wise annotations, one can obtain $\frac{CN_{\text{int}}}{H}$ trajectory-wise annotations; [15]'s analysis shows that Behavior Cloning with this number of trajectories from π^{E} returns a policy of suboptimality $O(\frac{RH \log B}{CN_{\text{int}}})$; recall Theorem 2. Thus, if $C \ll \frac{R}{\mu}$, Algorithm 1 has a better cost-efficiency guarantee than Behavior Cloning.

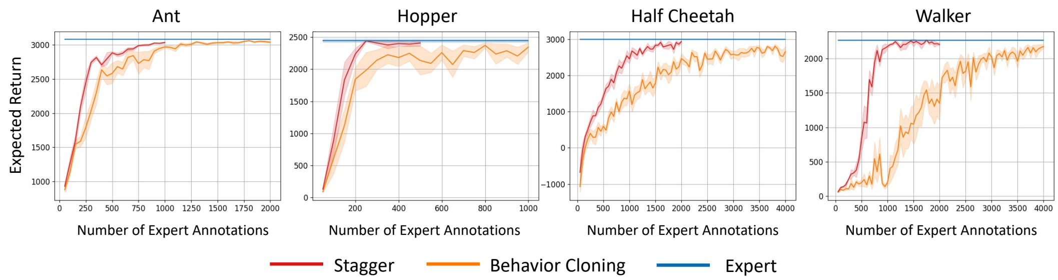

Figure 1: State-wise sample complexity comparison between Behavior Cloning and STAGGER. Shaded areas show the 10th–90th percentile bootstrap confidence intervals [13] over 10 runs. STAGGER matches or exceeds BC with 50% fewer annotations, achieving better state-wise annotation efficiency.

We now sketch the proof of Theorem 3. In line with [15], we define the online, on-policy state-wise estimation error as

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

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 [15], their definition requires all H states per trajectory, while ours allows unbiased approximation from a single state. This enables each labeled state to serve as immediate online feedback, fully utilizing the adaptivity of online learning. In contrast, trajectory-wise annotations may cause the online learning oracle to fall into a delayed feedback setting [19, 72], which incurs a fundamental extra factor of H in state-wise sample complexity compared to our approach.

3.2 Experimental Comparison

We conduct a simple simulation study comparing the sample efficiency of log-loss Behavior Cloning [15] and STAGGER in four MuJoCo [70, 6] continuous control tasks with $H = 1000$ and pretrained deterministic MLP experts [50, 51]. Considering MuJoCo's low sensitivity to horizon length [15], we reveal expert states one by one along consecutive trajectories for BC to allow fine-grained state-level sample complexity comparison, while STAGGER queries exactly one state per round by sampling from the latest rollout and updating immediately with the expert action. In STAGGER, we implement the online learning oracle $\mathbb{A}$ so that it outputs a policy that approximately minimizes the log-loss over historical (state, expert action) pairs. In addition to log-loss, we also include results with online learning oracle minimizing historical samples' 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. Overall we see that, 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 oftentimes combine offline and interactive feedback modalities [24, 18]: for example, in autonomous driving [76, 2, 77], the learner has access

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

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

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

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

to some offline expert demonstrations to start with, and also receives state-wise interactive expert demonstration feedback in subsequent finetuning phase. Motivated by this common practice, we formulate the following problem setup:

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

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

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

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

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 $\mathcal{B}_{\text{bc}}$, 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 $\mathcal{B}_{\text{bc}}$ 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 $\mathcal{O}^{\text{State}}$. For the purpose of analysis, we introduce the following technical definition.

Definition 4 (Each-step policy completion)**.** *Given a base policy class $\mathcal{B}$, define for each step $h \in [H]$*

$$\mathcal{B}_h := \left\{ \pi_h \;\middle|\; \pi = (\pi_1, \ldots, \pi_H) \in \mathcal{B} \right\}.$$

Then the each-step completion of $\mathcal{B}$ is defined as

$$\tilde{\mathcal{B}} := \left\{ \pi = (\pi_1, \ldots, \pi_H) \;\middle|\; \pi_h \in \mathcal{B}_h \textit{ for all } h \in [H] \right\}.$$

In words, each $\pi \in \tilde{\mathcal{B}}$ uses a possibly distinct policy π_h from $\mathcal{B}_h$ to take action at step h. By definition, $\tilde{B} := |\tilde{\mathcal{B}}|$ is at most B^H, since $|\mathcal{B}_h| \leq B$. Under non-parameter-sharing settings [47, 45, 44, 15], where the base policy class $\mathcal{B}$ allows the policies used at each step to be chosen independently, $\tilde{B} = B$.

Theorem 5. *If* WARM-STAGGER *is run with a state-wise expert annotation oracle $\mathcal{O}^{\text{State}}$, an MDP $\mathcal{M}$ where $(\mathcal{M}, \pi^{\text{E}})$ is μ-recoverable, a policy class $\mathcal{B}$ such that deterministic realizability (Assumption* **??***) holds, and the online learning oracle $\mathbb{A}$ set as the exponential weight algorithm with each-step mixing policies $\bar{\pi}_u \in \bar{\Pi}_{\mathcal{B}}$, then it returns $\hat{\pi}$ such that, with probability at least $1 - \delta$,*

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

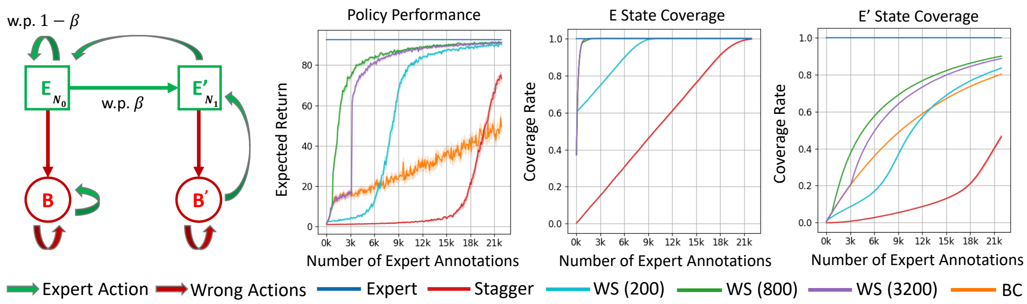

Figure 2: MDP construction and simulation results of algorithms with rewards assigned only in $\mathbf{E}$. We evaluate WARM-STAGGER (WS) with 200, 800, 3200 offline (state, expert action) pairs. All methods are evaluated under equal total annotation cost with $C = 1$. With 800 offline (state, expert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

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

Theorem 5 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 of $O\left(\frac{RH\log(B/\delta)}{N_{\text{off}}}\right)$ (cf. Theorem 2); second, STAGGER without using offline data has a suboptimality of $O\left(\frac{\mu H\log(B/\delta)}{N_{\text{int}}}\right)$ (cf. Theorem 3). We conjecture that the $\log\tilde{B}$ dependence may be sharpened to $\log B$; we leave this as an interesting open question.

Remark 6. *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 our* 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 5 is perhaps best viewed as a fall-back guarantee for WARM-STAGGER: its performance is not much worse than either of the baselines. In this section, we demonstrate that the benefit of hybrid imitation learning can go beyond this: we construct an MDP motivated by practical challenges, 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 7. *For large enough S, H, there exists an MDP $\mathcal{M}$ with S states, 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 $\tilde{O}\left(S/H\right)$ offline trajectories and $\tilde{O}(1)$ expert interactions,* WARM-STAGGER *learns a policy $\hat{\pi}$ such that $J(\hat{\pi}) = J(\pi^E)$.*

Theorem 7 suggests that when $HS \gg \max(1, C)$, 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 action space of cardinality greater than 20. 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 is partitioned to the following subsets:

- Expert ideal states $\mathbf{E}$, where $|\mathbf{E}| = N_0$: this can model for example, the agent driving stably on the edge of a cliff [49], where any incorrect action transitions the agent to the unrecoverable absorbing state $\mathbf{B}$ (e.g., 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).
- Unrecoverable state $\mathbf{B}$: a special absorbing state that is unrecoverable by any action (dead).
- Expert recoverable states $\mathbf{E}'$: this models the agent getting off from the edge of the cliff to a safe slope. When in $\mathbf{E}'$, taking the expert action allows the agent to return to a uniformly sampled state in $\mathbf{E}$. Taking a wrong action from $\mathbf{E}'$ leads to $\mathbf{B}'$ (e.g., rest area).
- Recoverable state $\mathbf{B}'$: Not knowing how to act in $\mathbf{B}'$ will result in the agent getting trapped in $\mathbf{B}'$ for the episode.

In the following proof sketch, we briefly justify each baseline's performance as stated in Theorem 7. (1) 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. (2) 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 unseen states in $\mathbf{E}$ even with $\Omega(HS)$ queries. (3) WARM-STAGGER benefits from offline data that fully covers $\mathbf{E}$, and uses limited interaction 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 a more challenging 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, which corresponds to a special case of the log loss under a deterministic learner policy class 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 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 $\mathbf{E}$ states, consistent with the cold-start intuition; (3) WARM-STAGGER (WS), when initialized with limited 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 continuous-control 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 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 equal 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.

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 [43], was reduced to offline, state-wise supervised learning in early works [47, 68] and named Behavior

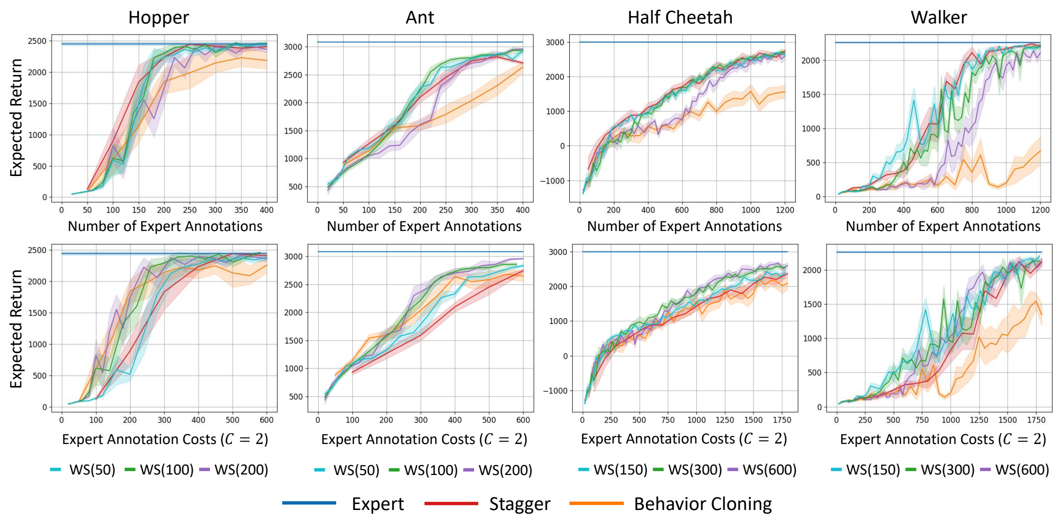

Figure 3: Sample and cost efficiency on MuJoCo tasks. The top row shows expected return vs. number of annotations ($C = 1$); the bottom row shows performance under a cost-aware setting ($C = 2$). WARM-STAGGER (WS) is initialized with 1/8, 1/4, or 1/2 of the total annotation budget as offline demonstrations.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

Cloning (BC). A recent analysis by [15] 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 [43], arises when imperfect imitation drives the learner to unseen states, resulting in a compounding sequence of mistakes. From a data collection perspective, this can be mitigated by noise-injection approaches such as [27, 23]. By leveraging additional environment interactions, generative-adversarial IL methods [17, 63, 22, 62] frame learning as a two-player game that 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, 67], which also starts from offline demonstrations and assumes access to the MDP dynamics or interactive rollouts. Quantitative comparisons with these methods are beyond our scope, as they rely on extensive interactions with the MDP, while we focus on the advantages of state-wise interactive annotations. Recent work [46] introduces Hybrid Inverse Reinforcement Learning, which leverages hybrid Reinforcement Learning to accelerate its inner loop of policy search; in contrast, our "hybrid" setting focus on utilizing heterogeneous data modalities. Recent offline imitation learning approaches [8, 73] do not require MDP access but still require access to strong offline datasets, 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 incorporate environment interaction cost as future work.

Imitation Learning with interactive demonstrations, first proposed by [47], allows the expert to provide corrective feedback to the learner's action retroactively. Assuming low costs of expert recovery from mistakes, termed recoverability, DAgger [49], and following works [25, 48, 64, 9, 10, 44] outperform traditional BC both theoretically and empirically. However, this efficiency demands substantial annotation effort [34]. Although DAgger [49] and some subsequent works [64, 44, 65, 15] popularized the convention of annotating full state sequences, there has also been growing interest in state-wise annotation [30, 52, 31], which appeared as early as [47, 20]. In fact, practical applications of DAgger often adopt state-wise annotation in expert-in-the-loop [34, 60, 33] designs, as seen in [75, 24, 18, 71], where issues such as inconsistencies caused by retroactive relabeling [26] can be mitigated. These methods often leverage human- or machine-gated expert interventions to ensure safety during data collection [76, 35], provide more targeted feedback [36, 11], and enable on-the-fly learning [57]. The common 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 theoretical foundation for this increasingly popular paradigm of state-wise annotation.

Utlizing Offline Data for Interactive Learning. Many practical deployments of interactive learning systems do not start from tabula rasa; instead, prior knowledge of various forms are oftentimes available. Combining offline and interactive feedback has recently gained much popularity such as instruction finetuning large language models [14, 42], and bandit machine translation [38]. Many recent theoretical works in reinforcement learning try to quantify the computational and statistical benefit of combining offline and online feedback: for example, [29, 69] shows provable reduction of sample complexity using hybrid reinforcement learning, using novel notions of partial coverage; [59] shows that under some structural assumptions on the MDP, hybrid RL achieves computational savings. Many works also quantify the benefit of utilizing additional offline data sources in the contextual bandit domain; for example, [39, 56, 74] study warm-starting contextual bandits using offline bandit data and supervised learning data, respectively. While some DAgger variants [76, 18] also operate in a hybrid setting, our work focuses on a fundamental formulation that explicitly accounts for practical cost asymmetry between offline and interactive annotations [54], and, to the best of our knowledge, is the first to provide a rigorous framework with provable sample efficiency guarantees.

6 Conclusion

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

Limitations: Our theory provide sample complexity guarantees for the discrete-action setting with deterministic and realizable expert. When such assumptions are relaxed, additional challenges arise [58]. 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 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] 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.

[3] 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.

[4] 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.

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

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

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

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

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

[18] 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.

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

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

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

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

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

[33] 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.

[34] 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.

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

[36] 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.

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

[38] 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.

[39] 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.

[40] 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.

[41] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. Advances in neural information processing systems, 29, 2016.

[42] 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.

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

[44] 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.

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

[50] 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.

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

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

[53] 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.

[54] 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.

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

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

[57] 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.

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

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] Umar Syed, Michael Bowling, and Robert E Schapire. Apprenticeship learning using linear programming. In Proceedings of the 25th international conference on Machine learning, pages 1032–1039, 2008.

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

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

[69] 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.

[70] 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.

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

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

[73] 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.

[74] 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.

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

[76] 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.

[77] 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.

[78] 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, hyper-parameters, 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., [70, 50]). 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.

Interactive and Hybrid Imitation Learning: Provably Beating Behavior Cloning — Supplementary Materials

A Additional Notations and Related Work

A.1 Additional Notations

In line with [15], we introduce another annotation oracle that models interaction with the demonstration expert: the *trajectory-wise demonstration oracle* $\mathcal{O}^{\mathrm{Traj}}$, which takes as input a state sequence $s_{1:H}$ and returns $a^*_{1:H} \sim \pi^{\mathrm{E}}(\cdot \mid s_{1:H})$. We also provide a formal definition of first-step policy mixing—used in the definition of $\hat{\pi}$ (see Algorithm 1, Algorithm 2), which is a common technique (e.g., [64][66, 41]), defined as follows:

Definition 8 (First-step mixing of $\mathcal{B}$)**.** $\Pi_{\mathcal{B}} := \{\pi_u : u \in \Delta(\mathcal{B})\}$*, 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.*

Importantly, π_u is not a stationary policy; as a result, $a_{1:H}$ are *dependent* conditioned on $s_{1:H}$, while $a_{1:H}$ are only conditionally independent given $s_{1:H}$ and the random policy π drawn.

Additionally, we use $\pi(\cdot \parallel s_{1:H})$ to denote the causally-conditioned probability of action sequence $a_{1:H}$ induced by π, given state sequence $s_{1:H}$ [78]. [2] To elaborate:

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

It is well-known that the trajectory distribution induced by Markovian policies and their first-step mixings π can be factorized to the product of $\pi(a_{1:H} \parallel s_{1:H})$ and the causally-conditioned probability of the state sequence given the action sequence (Definition 39 and Lemma 40). When it is clear from context, we use shorthand $\pi(s_{1:H})$ for $\pi(\cdot \parallel s_{1:H})$.

A.2 Useful Distance Measures

In the following, we present 4 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*

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

$\lambda(\pi \parallel \pi')$ 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 [47, 49] with different guarantees for $\lambda(\pi \parallel \pi^{\mathrm{E}})$ and $\lambda(\pi^{\mathrm{E}} \parallel \pi)$ (Lemma 35).

Definition 10 (Trajectory-wise L_∞-semi-metric [15])**.** *For a pair of Markovian policies π and π', define their trajectory-wise L_∞-semi-metric as*

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

$\rho(\pi \parallel \pi')$ is the probability of any action taken by π' deviating from actions in trajectories induced by π, which is symmetric [15]. A bound on $\rho(\pi \parallel \pi^{\mathrm{E}})$ leads to straightforward performance difference guarantee: $J(\pi^{\mathrm{E}}) - J(\pi) \leq R \cdot \rho(\pi \parallel \pi^{\mathrm{E}})$ [15] (Lemma 36).

Definition 11 (State-wise Hellinger distance)**.** *For a pair of Markovian policies π and π', define their state-wise Hellinger distance as* $\mathbb{E}_{(s,h)\sim d^{\pi}}\left[D_H^2(\pi_h(s), \pi'_h(s))\right]$.

[2] The use of $\parallel$ highlights its distinction from standard conditioning on $s_{1:H}$.

State-wise Hellinger distance represents the expected Hellinger distance between the action distribution of π and π' on $(s, h) \sim d^\pi$. One notable feature here is that the distance is evaluated between $\pi_h(\cdot \mid s)$ and $\pi'_h(\cdot \mid s)$, independent of the original action a taken by π when visiting s. By Lemma 35, state-wise Hellinger distance can relate to trajectory-wise L_1-divergence.

Definition 12 (Decoupled Hellinger distance)**.** *For a pair of Markovian policies π and π', define their decoupled Hellinger distance as $\mathbb{E}^\pi \left[D_H^2(\pi(s_{1:H}), \pi'(s_{1:H}))\right]$.*

Similarly, $\mathbb{E}^\pi \left[D_{\mathrm{H}}^2(\pi(s_{1:H}), \pi'(s_{1:H}))\right]$ denotes the expected Hellinger distance between the distribution of actions $\pi(s_{1:H})$ and $\pi'(s_{1:H})$ on state sequence $s_{1:H}$ visited by π. This allows decoupled analysis for state and action sequences, which is useful for the proof of Theorem 26.

A.3 Additional Related Work

State-wise v.s. trajectory-wise annotations in imitation learning. The debate over using state-wise versus trajectory-wise annotations traces back to the reduction from imitation learning to supervised learning [47]. While the sample complexity of BC can be interpreted in a state-wise manner (e.g. [9, 30]), BC conventionally relies on full trajectories of expert demonstrations. Recent advancements [15] seem to settle the debate in favor of trajectory-wise annotations through more refined analysis. However, the question "Is trajectory-wise annotation all you need?" remains unresolved in the interactive setting. To the best of our knowledge, we are the first to explicitly address this debate, systematically developing efficient algorithms tailored to both state-wise and trajectory-wise annotations (See Section E.3).

First-step mixing and each-step mixing policies. The emergence of first-step mixing originated from technical considerations. In may interactive IL methods [49, 48], 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 [44, 30, 31, 15] 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. While [64] implies the usage of a first-step mixture policy class, to the best of our knowledge, we are the first to establish a concrete online regret guarantee this policy class.

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 [12][47, 49, 48, 64]. However, when selecting the mixture policy class for each learning round, ambiguity arises regarding the choice between first-step mixing and each-step mixing [64]. For each-step mixture policies, [30] was the first to explicitly distinguish this approach from first-step mixing. In other works [44, 15], 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 [15] and focuses on log loss, we believe the $1/n$ rate is not exclusive to log loss. Despite requiring an additional supervision oracle, [28] suggests that trajectory-wise annotation complexity similar to Theorem 26 can be achieved using Halving [55] and 0-1 loss.

From an algorithmic perspective, we explored trajectory-wise annotation with first-step mixing (Algorithm 3) 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 additional H factor to the sample complexity, which is undesirable. State-wise annotation with each-step mixing remains an open question for future research.

In terms of practical implementation, it is worth noting that even with oracle-efficient implementations (e.g. [30, 31]), interactive IL requires multiple computational oracle calls per learning round. 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., [48]).

Lower bounds in interactive imitation leanring. From an information-theoretic perspective, a line of work [45][44, 15] provides lower bounds for imitation learning under the realizable setting and considers μ-recoverability. [44] 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. [15] shifts attention back to the general setting and establishes a $\Omega(\frac{H}{\epsilon})$ sample complexity lower bound for trajectory-wise annotation.

We argue that the result in [15] also 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 is equivalent to annotating a single state. In their special MDP case (1-recoverable), Algorithm 1 achieves $\tilde{O}(\frac{H\log(B)}{\epsilon})$ state-wise sample complexity, which does not contradict existing lower bounds. Nonetheless, obtaining lower bounds for state-wise sample complexity for general MDPs and policy classes remains an open question.

B Proof for STAGGER

In the following, 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 state-wise Hellinger estimation error:

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

where $\pi_h^n(s)$ and $\pi_h^E(s)$ denote the action distributions over $\mathcal{A}$ produced by the policies π^n and π^E at step h in state s.

We first prove this in Lemma 13, and then prove the state-wise annotation complexity of Algorithm 1 in Theorem 14.

Different from [15], where access to full action demonstrations is assumed, we consider a more restrictive model where, at each round n, only a single state (s^n, h^n) from the trajectory induced by π^n is sampled and annotated by the expert.

Lemma 13. *For any MDP $\mathcal{M}$, deterministic expert π^{E}, and sequence of policies $\{\pi^n\}_{n=1}^N$, each of which Markovian, then $\hat{\pi}$, the first-step uniform mixture of $\{\pi^1, \dots, \pi^N\}$ satisfies:*

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

Proof. By Lemma 35, under the assumption of recoverability, the performance difference between $\hat{\pi}$ and the expert is bounded by

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

where we recall the notation that

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

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

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

Observe that $H \cdot \mathbb{E}_{(s,h)\sim d^{\pi}} \left[D_H^2(\pi_h(s), \pi_h^{\mathrm{E}}(s)) \right] = \sum_{h=1}^{H} \mathbb{E}^{\pi} \left[D_H^2(\pi_h(s_h), \pi_h^{\mathrm{E}}(s_h)) \right]$, we conclude the proof by applying Lemma 34 with $p = \pi_h(s_h)$ and $q = \pi_h^{\mathrm{E}}(s_h)$, which gives

$$D_{\mathrm{H}}^2(\pi_h(s), \pi_h^{\mathrm{E}}(s)) \geq \frac{1}{2}\|\pi_h(s) - \pi_h^{\mathrm{E}}(s)\|_1.$$

Theorem 14 (Theorem 3 Restated)**.** *If* STAGGER *(Algorithm 1) is run with a state-wise expert annotation oracle $\mathcal{O}^{\text{State}}$, an MDP $\mathcal{M}$ where $(\mathcal{M}, \pi^{\text{E}})$ is μ-recoverable, a policy class $\mathcal{B}$ such that realizability (Assumption* **??***) 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 $1-\delta$,*

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

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

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

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

By using $\bar{\pi}_u$, the loss functions at each round that passed through online learning oracle $\mathbb{A}$, $\ell^n(\pi)$ is of the form

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

which is 1-exp-concave with respect to u. Thus, implementing $\mathbb{A}$ using the exponential weights algorithm (Proposition 37) achieves:

$$\sum_{n=1}^{N_{\text{int}}} \log(1/\pi_{h^n}^n(a^{*,n} \mid s^n)) \leq \sum_{n=1}^{N_{\text{int}}} \log(1/\pi_{h^n}^{\text{E}}(a^{*,n} \mid s^n)) + \log(B) = \log(B).$$

Then, Lemma 38, a standard online-to-batch conversion argument with $x^n = (s^n, h^n)$, $y^n = a^{*,n}$, $g_* = \pi^{\text{E}}$, and $\mathcal{H}^n = \{o^{n'}\}_{n'=1}^{n}$, where $o^n = (s^n, h^n, a^n, a^{*,n})$, implies that with probability at least $1-\delta$,

$$\text{OnEst}_{N_{\text{int}}}^{\text{State}} = \sum_{n=1}^{N_{\text{int}}} \mathbb{E}_{(s^n,h^n)\sim d^{\pi^n}} \left[D_{\text{H}}^2(\pi_{h^n}^n(s^n), \pi_{h^n}^{\text{E}}(s^n))\right] \leq \log(B) + 2\log(1/\delta).$$

The second part of the theorem follows by applying Lemma 13. ∎

C Proof for WARM-STAGGER

In this section, we analyze the theoretical guarantees of WARM-STAGGER under the realizability with deterministic expert. We show that all intermediate policies, as well as the final returned mixture policy, enjoy small Hellinger distance to the expert's trajectory distribution, due to their agreement on the offline dataset. Our analysis builds on maximum likelihood estimator (MLE) generalization guarantees under log-loss minimization, and leverages the fact that each each-step mixing policy in WARM-STAGGER can be viewed as a first-step policy mixture. The following structural claim shows that any Markovian policy π can be represented as a first-step mixing over deterministic Markovian policies from a finite class.

Lemma 15. *Let $\bar{\pi}_u$ be a each-step mixing policy that, at each step $h \in [H]$, samples a base policy $\pi_h \in \mathcal{B}$ independently according to $u \in \Delta(\mathcal{B})$. Then, its induced trajectory distribution is equivalent to a first-step mixing the each-step policy completion of $\mathcal{B}$, denoted by $\tilde{\mathcal{B}} := \{(\pi_1, \ldots, \pi_H) : \pi_h \in \mathcal{B}\}$ (see Definition 4).*

Proof. Let $\nu = (\pi_1, \ldots, \pi_H) \in \tilde{\mathcal{B}}$. Define $u(\nu) := \prod_{h=1}^{H} u(\pi_h)$, which is a valid distribution over $\tilde{\mathcal{B}}$. 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 conditional distribution over actions given the state sequence is

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

Under the first-step mixture policy $\pi_{u(\nu)}$ over $\tilde{\mathcal{B}}$, a full tuple $\nu = (\pi_1, \ldots, \pi_H)$ is sampled once from $u(\nu)$, and actions are drawn as $a_h \sim \pi_h(\cdot \mid s_h)$. The resulting action distribution is

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

Expanding the sum yields

$$\sum_{(\pi_1,\ldots,\pi_H)} \left(\prod_{h=1}^{H} u(\pi_h)\pi_h(a_h \mid s_h) \right) = \prod_{h=1}^{H} \sum_{\pi_h \in \mathcal{B}} u(\pi_h)\, \pi_h(a_h \mid s_h),$$

by the distributive property and independence of the product.

Therefore,$\bar{\pi}_u(a_{1:H} \mid s_{1:H}) = \pi_{u(\nu)}(a_{1:H} \mid s_{1:H})$, and both policies induce the same trajectory distribution by Lemma 40.

Lemma 16. *Let $\mathcal{B}_{bc} := \{\pi \in \mathcal{B} : \pi(s) = \pi^E(s)\ \forall s \in D_{off}\}$ be the set of policies that agree with the expert on the offline dataset of N_{off} expert trajectories. Assume the expert π^E is deterministic and realizable. Then, with probability at least $1 - \delta$, for all π_i generated in* WARM-STAGGER *(Algorithm 2), it holds that:*

$$D_H^2(P_{\pi_i}, P_{\pi^E}) \leq O\left(\frac{\log(\tilde{B}/\delta)}{N_{off}} \right).$$

Furthermore, the returned policy $\hat{\pi}$ satisfies:

$$D_H^2(P_{\hat{\pi}}, P_{\pi^E}) \leq O\left(\frac{H\log(\tilde{B}/\delta)}{N_{off}} \right).$$

Proof. Let $\mathcal{B}_{\text{bc}}$ denote the realizable class of policies that agree with π^E on the offline dataset. By the MLE generalization bound [15], for any single policy $\pi \in \mathcal{B}_{\text{bc}}$, we have with probability at least $1 - \delta$:

$$D_H^2(P_{\pi}, P_{\pi^E}) \leq O\left(\frac{\log(B/\delta)}{N_{\text{off}}} \right).$$

However, in WARM-STAGGER, policies π_i are not selected directly from $\mathcal{B}_{\text{bc}}$, but are instead mixtures over $\mathcal{B}_{\text{bc}}$ at each time step. A full trajectory is therefore sampled by first choosing a policy π^h at each step $h \in [H]$, inducing an effective policy over sequences from $\tilde{\mathcal{B}}_{\text{bc}}$. Therefore, by appling $\tilde{\mathcal{B}}$ for Theorem 2, the BC guarantee holds for all $\pi \in \tilde{\mathcal{B}}$, with a factor of $\log|\tilde{\mathcal{B}}| \leq H \log B$ in the worst case instead of $\log B$. Therefore, by Lemma 15, each π_i can be viewed as first-step mixing and satisfies:

$$D_H^2(P_{\pi_i}, P_{\pi^E}) \leq O\left(\frac{H\log(\tilde{B}/\delta)}{N_{\text{off}}} \right).$$

This remains a worst-case bound. If the per-step base policies are drawn from a factored structure (e.g., $\pi = (\pi_1, \ldots, \pi_H)$ with $\pi_h \in \mathcal{B}_{\text{bc}}$), and the support is shared across steps, the effective covering number can be much smaller, reducing the overhead back to $\log B$.

Finally, the returned policy $\hat{\pi}$ is a first-step mixing of $\{\pi_1, \ldots, \pi_N\}$, and thus the same Hellinger bound carries over by convexity.

Theorem 17 (Theorem 5 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 realizability holds, and the online learning oracle $\mathcal{O}$ set as the exponential weight algorithm, then it returns $\hat{\pi}$ such that, with probability at least $1 - \delta$,*

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

Proof. By Lemma 16, with probability at least $1-\delta/2$, for all $\pi \in \{\pi_1, \cdots, \pi_N\}$, it satisfies that $D_H^2(P_\pi, P_{\pi^{\mathrm{E}}}) \le O\left(\frac{\log(\tilde{B}/\delta)}{N_{\text{off}}}\right)$. This implies the first-step uniform mixture of $\{\pi_1, \cdots, \pi_N\}$ satisfies

$$D_H^2(P_{\hat{\pi}}, P_{\pi^{\mathrm{E}}}) = \frac{1}{N}\sum_{i=1}^{N} D_H^2(P_{\pi_i}, P_{\pi^{\mathrm{E}}}) \le O\left(\frac{\log(\tilde{B}/\delta)}{N_{\text{off}}}\right)$$

By invoking Theorem [15][2.1], we have,

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

For the second half of the proof, we notice that by definition $\pi^{\mathrm{E}} \in \mathcal{B}_{\text{bc}}$. Then, by applying Theorem 3, with probability at least $1-\delta/2$, the returned $\hat{\pi}$ satisfies

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

Together, we conclude our proof by applying union bound.

D Proof for Theorem 7

We reintroduce the MDP $\mathcal{M}$ from Section 4.2, where the expert policy π^{E} is deterministic and the transition dynamics are time-homogeneous across all steps $h \in [H]$.

- **State Space** $\mathcal{S} = \mathbf{E} \cup \mathbf{E}' \cup \{\mathbf{B}, \mathbf{B}'\}$, where:
  - $\mathbf{E}$: ideal expert states, $|\mathbf{E}| = N_0$;
  - $\mathbf{E}'$: recoverable expert states, $|\mathbf{E}'| = N_1$;
  - $\mathbf{B}$: absorbing failure state (unrecoverable);
  - $\mathbf{B}'$: recoverable reset state.
- **Action Space** $\mathcal{A}$: $|\mathcal{A}| \ge 20$ discrete actions. For each state $s \in \mathcal{S}$, there is a unique expert action $\pi^{\mathrm{E}}(s)$; all others are incorrect.
- **Transition Dynamics**:
  - Initial Distribution ρ:

$$\rho(\mathbf{E}) = \frac{1}{1+\beta}, \quad \rho(\mathbf{E}') = \frac{\beta}{1+\beta}.$$

  - $s \in \mathbf{E}$:
    - $a = \pi^{\mathrm{E}}(s)$: transition to a uniformly random $s' \in \mathbf{E}$ with probability $1-\beta$, or to $s' \in \mathbf{E}'$ with probability β;
    - $a \ne \pi^{\mathrm{E}}(s)$: transition to $\mathbf{B}$.
  - $s \in \mathbf{E}'$:
    - $a = \pi^{\mathrm{E}}(s)$: return to a uniformly random $s' \in \mathbf{E}$;
    - $a \ne \pi^{\mathrm{E}}(s)$: transition to $\mathbf{B}'$.
  - $\mathbf{B}$: absorbing for all actions.
  - $\mathbf{B}'$:
    - $a = \pi^{\mathrm{E}}(\mathbf{B}')$: return to a uniformly random $s' \in \mathbf{E}$;
    - $a \ne \pi^{\mathrm{E}}(\mathbf{B}')$: remain in $\mathbf{B}'$.
- **Reward Function**: We consider two variants:
  - R_1 (used in theory):

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

– R_2 (used in simulations):

$$R_2(s,a) = \begin{cases} 1 & \text{if } s \in \mathbf{E}, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 18 (Restatement of Theorem 7)**.** *There exists an MDP with state space partitioned into N_0 good states, N_1 recover states, and 2 bad states, episode length $H \geq 50$, action size $A \geq 20$, such that to achieve smaller than $\frac{H}{2}$ suboptimality compared to expert:*

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

In contrast, WARM-STAGGER *achieves expert performance with probability at least $1-\delta$, using*

$$\begin{aligned} N_{off} &= O(\frac{N_0}{H}\log(N_0/\delta)) \quad \textit{expert trajectories,} \\ N_{on} &= O(\log(1/\delta)) \quad \textit{interactive annotations,} \\ &\quad \textit{with total annotation cost } \tilde{O}(N_0). \end{aligned} \tag{3}$$

Proof. The proof is divided into three parts. First, by Lemma 19, we show that in the aforementioned MDP $\mathcal{M}$ with $\beta = \frac{8}{H-8}$ and reward function R_1, Behavior Cloning requires $\Omega(HN_1)$ expert trajectories to achieve suboptimality no greater than $H/2$. Next, in Lemma 23, we show that STAGGER, which rolls out the learner policy and queries the expert on only one state per trajectory (sampled uniformly), requires $N_{\text{on}} = \Omega(HN_0)$ interactive annotations to achieve suboptimality no greater than $H/2$.

Finally, by Lemma 25, we demonstrate that WARM-STAGGER achieves expert performance using $O(N_0 \log(N_0/\delta))$ offline demonstrations and $O(\log(1/\delta))$ interactive annotations.

Lemma 19 (BC suboptimality lower bound)**.** *Consider the MDP specified in Section 4.2 with $H \geq 50$, $A \geq 20$, and $\beta = \frac{8}{H-8}$. If Behavior Cloning collects no more than*

$$N_{off} < \frac{N_1(1+\beta)}{\beta H}\log\left(\frac{1}{1-\delta/10}\right)$$

expert trajectories, then with probability at least $1-\delta$, the expected suboptimality of its returned policy $\hat{\pi}$ is bounded by:

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

Proof. In the following, we show that insufficient expert trajectories leads to poor $\mathbf{E}'$ coverage, which in turn causes the BC policy to frequently fail recovery and get stuck in the absorbing bad state, incurring a large suboptimality compared to the expert.

By Lemma 22, when N_{off} is below the stated threshold, at most $1/10$ of $\mathbf{E}'$ is covered by expert trajectories with probability $1-\delta$. Hence, the policy returned by BC has no information on at least $9/10$ of $\mathbf{E}'$, and fails to act correctly upon reaching them. Lemma 21 further guarantees that with probability at least 0.79, the policy reaches $\mathbf{E}'$ within the first $H/5$ steps. Conditioned on being in $\mathbf{E}'$, the chance of hitting an unseen state is at least 0.9 due to the MDP design, and given $A \geq 20$, the probability of taking the wrong action is at least 0.95. Therefore, with probability at least $0.79 \cdot 0.9 \cdot 0.95 \cdot 0.95 > 0.64$. Under this event, the trajectory enters s_{bad} by time step $0.8H+2$, leading to zero reward for $0.8H-1$ steps, which is no less than $0.78H$ with $H \geq 50$.

Multiplying all factors, with probability no less than one $1-\delta$, the expected suboptimality of the returned policy is lowerbouded by:

$$J(\pi^{\text{E}}) - J(\pi_{\text{BC}}) \geq \underbrace{0.798}_{\text{reach } \mathbf{E}'} \times \underbrace{0.9}_{\text{unseen state}} \times \underbrace{0.95}_{\text{wrong action}} \times \underbrace{0.95}_{\text{reset failure}} \times \underbrace{0.78H}_{\text{0 reward steps}} > \frac{H}{2}$$

which concludes the proof.

Lemma 20 (Coverage of expert with N_{off} trajectories)**.** *Consider an MDP specified in Section 4.2 with stationary distribution ρ: $\rho(\mathbf{E}) = \frac{1}{1+\beta}$, $\rho(\mathbf{E}') = \frac{\beta}{1+\beta}$. For N_{off} trajectories of horizon H:*

1. *All N_0 states in $\mathbf{E}$ are visited with probability $\geq 1 - \delta$ if:*
$$N_{off} \geq \frac{N_0(1+\beta)}{H} \log\left(\frac{N_0}{\delta}\right)$$

2. *All N_1 states in $\mathbf{E}'$ are visited with probability $\geq 1 - \delta$ if:*
$$N_{off} \geq \frac{N_1(1+\beta)}{\beta H} \log\left(\frac{N_1}{\delta}\right)$$

Proof. The stationary distribution gives:

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

It can be verified that for any $h \in [H]$,

$$d_h^{\pi^{\text{E}}}(s \in \mathbf{E}) = \frac{1}{N_0(1+\beta)}, \quad d_h^{\pi^{\text{E}}}(s \in \mathbf{E}') = \frac{\beta}{N_1(1+\beta)}.$$

By this observation, we can view states from an expert trajectory as i.i.d. drawn from ρ.

Then, for the coverage of Expert on $\mathbf{E}$, given N_{off} expert trajectories which equals to HN_{off} states, we have for each state s in $\mathbf{E}$:

$$\mathbb{P}(s \in \mathbf{E} \,\text{not visited}) = \left(1 - \frac{1}{N_0(1+\beta)}\right)^{HN_{\text{off}}}.$$

By taking union bound for all $s \in \mathbf{E}$, we have

$$\mathbb{P}(\exists s \in \mathbf{E}\,, \text{s.t.}\, s \,\text{not visited}) \leq N_0 \left(1 - \frac{1}{N_0(1+\beta)}\right)^{HN_{\text{off}}},$$

which should be smaller than δ.

By solving
$$N_0 \left(1 - \frac{1}{N_0(1+\beta)}\right)^{HN_{\text{off}}} < \delta,$$

We take natural logarithms on both sides and apply logarithm properties,

$$\log N_0 + HN_{\text{off}} \log\left(1 - \frac{1}{N_0(1+\beta)}\right) < \log \delta.$$

Apply the inequality $\log(1-x) \leq -x$ with $x = \frac{1}{N_0(1+\beta)}$:

$$\log N_0 - \frac{HN_{\text{off}}}{N_0(1+\beta)} < \log \delta$$

Rearrange terms and complete the first part of the proof

$$N_{\text{off}} \geq \frac{N_0(1+\beta)}{H} \log\left(\frac{N_0}{\delta}\right)$$

Similarly, for coverage on $\mathbf{E}'$, given HN_{off} total samples, we have

$$\mathbb{P}(\exists s \in \mathbf{E}'\,, \text{s.t.}\, s \,\text{not visited}) \leq N_1 \left(1 - \frac{\beta}{N_1(1+\beta)}\right)^{HN_{\text{off}}} \leq \delta$$

Solving using $\log(1-x) \leq -x$, we obtain

$$N_{\text{off}} \geq \frac{N_1(1+\beta)}{\beta H} \log\left(\frac{N_1}{\delta}\right),$$

which concludes the proof.

Lemma 21 (First $\mathbf{E}'$ visit)**.** *For an MDP specified in Section 4.2 with transition parameter $\beta = \frac{8}{H-8}$ ($H > 16$), any trajectory induced by any policy π that agrees with π^{E} on $\mathbf{E}$ satisfies:*

$$\mathbb{P}\big(\pi \textit{visit } s \in \mathbf{E}' \textit{ within } \frac{H}{5} \textit{ steps}\big) \geq 0.798$$

Proof. Since π agrees with π^{E} on $\mathbf{E}$, we only need to consider the first visit of $\mathbf{E}'$ by the expert. The state visition distribution for π^{E} satisfies that for any $h \in [H]$,

$$d_h^{\pi^{\mathrm{E}}}(s \in \mathbf{E}) = \frac{1}{N_0(1+\beta)}, \quad d_h^{\pi^{\mathrm{E}}}(s \in \mathbf{E}') = \frac{\beta}{N_1(1+\beta)}.$$

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

$$\left(\frac{1}{1+\beta}\right)^{\frac{H}{5}} = \left(1 - \frac{8}{H}\right)^{\frac{H}{5}} \leq e^{-\frac{8}{H}\cdot\frac{H}{5}} = e^{-\frac{8}{5}} < 0.202,$$

where we apply $1 - x \leq e^{-x}$. Thus:

$$\mathbb{P}\big(\pi^{\mathrm{E}}\text{visit } s \in \mathbf{E}' \text{ within } \frac{H}{5} \text{ steps}\big) > 1 - 0.202 = 0.798$$

Lemma 22 (Bounded $\mathbf{E}'$ coverage)**.** *Under the MDP's stationary distribution $\rho(s \in \mathbf{E}') = \frac{\beta}{N_1(1+\beta)}$, if the number of expert trajectories satisfies:*

$$N_{\textit{off}} \leq \frac{N_1(1+\beta)}{\beta} \log\left(\frac{1}{1-\delta/10}\right),$$

then with probability at least $1-\delta$, at most $\frac{1}{10}$ of $\mathbf{E}'$ states are visited:

$$\mathbb{P}\left(|\mathbf{E}'_{\textit{visited}}| > \frac{N_1}{10}\right) \leq \delta.$$

Proof. Recall that $\rho(s \in \mathbf{E}') = \frac{\beta}{N_1(1+\beta)}$. Given HN_{off} states from N_{off} expert trajectories, denote visited states in $\mathbf{E}'$ as $X := |\mathbf{E}'_{\text{visited}}|$. For each $s \in \mathbf{E}'$:

$$\mathbb{P}(s \text{ visited by expert}) = 1 - \left(1 - \frac{\beta}{N_1(1+\beta)}\right)^{HN_{\text{off}}}.$$

By linearity of expectation:

$$\mathbb{E}[X] = N_1\left[1 - \left(1 - \frac{\beta}{N_1(1+\beta)}\right)^{HN_{\text{off}}}\right].$$

Now, we apply Markov's inequality for $\frac{N_1}{10}$:

$$\mathbb{P}\left(X > \frac{N_1}{10}\right) \leq \frac{10\mathbb{E}[X]}{N_1} = 10\left[1 - \left(1 - \frac{\beta}{N_1(1+\beta)}\right)^{HN_{\text{off}}}\right].$$

To ensure $\mathbb{P}\left(X > \frac{N_1}{10}\right) \leq \delta$, it suffice to solve

$$10\left[1 - \left(1 - \frac{\beta}{N_1(1+\beta)}\right)^{HN_{\text{off}}}\right] \leq \delta.$$

By rearranging and taking natural logarithms, we have:

$$HN_{\text{off}} \log\left(1 - \frac{\beta}{N_1(1+\beta)}\right) \geq \log(1 - \frac{\delta}{10})$$

Apply $\log(1-x) \leq -x$:

$$HN_{\text{off}} \left(-\frac{\beta}{N_1(1+\beta)}\right) \geq \log(1 - \frac{\delta}{10}),$$

By rearranging the terms, we conclude that the number of visited states in $\mathbf{E}'$ by expert trajectories is no more than $N_1/10$ with probability at least $1-\delta$ if:

$$N_{\text{off}} \leq \frac{N_1(1+\beta)}{\beta H} \log\left(\frac{1}{1-\delta/10}\right).$$

Lemma 23 (STAGGER suboptimality lower bound)**.** *Consider the MDP $\mathcal{M}$ from Section 4.2 with $H \geq 50$, $A \geq 20$, and $\beta = \frac{8}{H-8}$. If* STAGGER *collects no more than*

$$N_{on} \leq \frac{\delta H N_0}{4}$$

interactive state-wise annotations. Then, with probability at least $1-\delta$, the returned policy $\hat{\pi}$ suffers suboptimality at least

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

Proof. By Lemma 24, if STAGGER collects at most $\frac{\delta H N_0}{4}$ interactive annotations as above, then with probability at least $1-\delta$, the number of distinct expert states $\mathbf{E}$ visited is fewer than $N_0/2$.

Under this event, the returned policy $\hat{\pi}$ agrees with the expert on at most half of $\mathbf{E}$. Starting from $\rho(\mathbf{E}) = \frac{1}{1+\beta}$, the agent begins in a random $s \in \mathbf{E}$, and with probability $\geq 1/2$, enters an unseen state where it takes a wrong action (w.p. ≥ 0.95), transitioning to absorbing $\mathbf{B}$ and earning zero reward for most of the episode.

We consider two-step failure:

With probability $\geq 1/2$, step 1 is already an unseen state, causing immediate failure. Suboptimality is at least $H-1$.

With probability $\leq 1/2$, the first step is safe, but with transition probability $1-\beta$ the agent remains in $\mathbf{E}$ and faces another unseen state at step 2 with probability $\geq 1/2$. Failure leads to at most 2 rewards, and suboptimality $\geq (1-\beta)(H-2)$.

We conclude the proof by combining them together and bring in values

$$J(\pi^{\mathrm{E}}) - J(\hat{\pi}) \geq \frac{1}{2(1+\beta)}(H-1) + \frac{1}{4(1+\beta)}(1-\beta)(H-2) \geq \frac{H}{2}.$$

Lemma 24 (Bounded $\mathbf{E}$ coverage under STAGGER)**.** *Suppose* STAGGER *collects at most*

$$N_{on} \leq \frac{\delta H N_0}{4}$$

interactive annotations. Then with probability at least $1-\delta$, fewer than half of the expert states in $\mathbf{E}$ are visited:

$$\mathbb{P}\left(|\mathbf{E}_{visited}| \geq \frac{N_0}{2}\right) \leq \delta.$$

Proof. Since each annotated trajectory in STAGGER samples only one state uniformly from the current rollout trajectory for annotation, we denote the indicator of whether a new, unseen state from

$\mathbf{E}$ at ruond i as $X_i \in \{0, 1\}$. By this, we have the total number of visited states by the end of round N_{on} as

$$\mathbf{E}_{\text{visited}} := \sum_{i=1}^{N_{\text{on}}} X_i.$$

We now upper bound the expected value of each X_i. Let $\mathcal{F}_{i-1}$ be the filtration up to round $i-1$. For any rollout of STAGGER in the designed MDP given an arbitrary number of already seen states in $\mathbf{E}$, the chance of seeing more than one unseen state in $\mathbf{E}$ decays geometricly, since the learner is forced to guess the correct action and transit to the obsorbing bad state $\mathbf{B}$ if failed. More specificaly, the chance of reaching t distinct $\mathbf{E}$ states before being trapped by $\mathbf{B}$ is upperbounded by $\frac{A-1}{A^t}$. Therefore, for any $A \geq 2$, the expected number of unique $\mathbf{E}$ states in a trajectory is upper bounded by:

$$\sum_{t=1}^{H} t \cdot (A-1) \left(\frac{1}{A}\right)^t \leq \frac{A}{A-1} \leq 2$$

Since the sampled annotation is uniformly from the trajectory, the probability it lands on a new $\mathbf{E}$ state is at most $2/H$. Hence,

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

By linearity of expectation:

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

Apply Markov's inequality:

$$\mathbb{P}(\mathbf{E}_{\text{visited}} \geq N_0/2) \leq \frac{\mathbb{E}[X]}{N_0/2} \leq \frac{4N_{\text{on}}}{HN_0}.$$

To ensure the probability is bounded by δ, we set:

$$\frac{4N_{\text{on}}}{HN_0} \leq \delta,$$

which results in

$$N_{\text{on}} \leq \frac{\delta H N_0}{4}.$$

This completes the proof.

Lemma 25 (Hybrid IL achieves expert performance under R_1)**.** *Consider the MDP $\mathcal{M}$ from Section 4.2 with reward function R_1 and $\beta = \frac{8}{H-8}$. Then with probability at least $1-\delta$,* WARM-STAGGER *achieves expert performance using $N_{\mathit{off}} = O\left(\frac{N_0}{H}\log(N_0/\delta)\right)$ offline expert trajectories and $N_{\mathit{on}} = O\left(\log(1/\delta)\right)$ interactive annotations.*

Proof. We divide the proof into three parts.

First, we state the high probability event for visiting all states in $\mathbf{E}$. By Lemma 20, if WARM-STAGGER uses

$$N_{\text{off}} \geq \frac{N_0(1+\beta)}{H} \log\left(\frac{N_0}{\delta/2}\right),$$

then with probability at least $1-\delta/2$, all states in $\mathbf{E}$ are visited by expert trajectories. Thus, the learner will take correct actions in $\mathbf{E}$.

Next, we analyze the high-probability event of visiting $\mathbf{B}'$. Since the number of offline trajectories is strictly below the threshold in Lemma 22, the learner fails to cover most of $\mathbf{E}'$. By Lemma 21, with probability at least 0.79, the policy visits some state in $\mathbf{E}'$ within the first $H/5$ steps of a rollout. Given insufficient coverage, this state is more than 0.9 probability unseen, and the learner takes the wrong action w.p. 0.95, which causes a transition to $\mathbf{B}'$. Once in $\mathbf{B}'$, the agent accumulates repeated $\mathbf{B}'$ of length $4H/5-1$ w.p. 0.95. As a result, the probability of selecting a $\mathbf{B}'$ state for annotation in each round is at least $1/2$ following the argument in Lemma 19, which requires $O(\log(1/\delta))$ interactive annotations to cover $\mathbf{B}'$.

Finally, after learner sees $\mathbf{B}'$ and all states in $\mathbf{E}$, under reawrd function R_1, the learner receives reward 1 at any step if: (i) it is in $\mathbf{E}$ or $\mathbf{E}'$, or (ii) it is in $\mathbf{B}'$ and takes the recovery action same as the expert. Since the learner now behaves like π^{E} on all states in $\mathbf{E}$ and can successfully recovers in $\mathbf{B}'$, its total return is the same as the expert. With union bound, this guarantee holds with probability at least $1 - \delta$.

E Additional Guarantees for 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, contrary to prior claims [15].

Here, we consider another oracle that models interacting with the demonstration expert: the *trajectory-wise demonstration oracle* $\mathcal{O}^{\mathrm{Traj}}$ that takes into a state-sequence $s_{1:H}$ and returns $a^*_{1:H} \sim \pi^{\mathrm{E}}(\cdot \parallel s_{1:H})$.

Additionally, we use $\pi(\cdot \parallel s_{1:H})$ to denote the causally-conditioned probability of action sequence $a_{1:H}$ induced by π, given state sequence $s_{1:H}$ [78]. [3] To elaborate:

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

It is well-known that the trajectory distribution induced by Markovian policies and their first-step mixings π can be factorized to the product of $\pi(a_{1:H} \parallel s_{1:H})$ and the causally-conditioned probability of the state sequence given the action sequence (Definition 39 and Lemma 40). When it is clear from context, we use shorthand $\pi(s_{1:H})$ for $\pi(\cdot \parallel s_{1:H})$.

E.1 Interactive IL Matches Offline IL on Trajectory-wise Annotation

Next, we consider the trajectory-wise sampling model. We present TRAGGER, another DAgger variant, namely Algorithm 3 and provide its sample complexity bounds.

Algorithm 3 TRAGGER: DAgger with trajectory-wise annotation oracle

Input: MDP $\mathcal{M}$, deterministic expert π^{E}, Markovian policy class $\mathcal{B}$, online learning oracle $\mathbb{A}$ with decision space $\Delta(\mathcal{B})$ and benchmark set $\{e_\pi : \pi \in \mathcal{B}\}$.
for $n = 1, \dots, N$ **do**
  Query $\mathbb{A}$ and receive $u^n \in \Delta(\mathcal{B})$.
  Execute $\pi^n := \pi_{u^n}$ 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

$$\ell^n(\pi) := \log\left(\frac{1}{\pi^n(a^{*,n}_{1:H} \parallel s^n_{1:H})}\right). \tag{4}$$

end for
Output $\hat{\pi}$, the first-step uniform mixture of policies in $\{\pi^1, \dots, \pi^n\}$.

Algorithm 3 uses first-step mixing policies $\pi_u \in \Pi_{\mathcal{B}}$ (recall Definition 8). At round n, it rolls out $\pi^n = \pi_{u^n}$ whose mixing weight u^n is obtained from an online learning oracle $\mathbb{A}$ and samples a full state sequence $s^n_{1:H}$. Same as Algorithm 1, Algorithm 3 also requires $\mathbb{A}$ to have decision space $\Delta(\mathcal{B})$ and benchmark set $\{e_\pi : \pi \in \mathcal{B}\}$. It then requests expert's trajectory-wise annotation $a^{*,n}_{1:H}$ and updates $\mathbb{A}$ by $\ell^n(\pi)$ (Eq. (4)). At the end of round N, the uniform first-step mixing of $\{\pi^n\}_{n=1}^N$ is returned, which is equivalent to returning $\pi_{\hat{u}}$, where $\hat{u} := \frac{1}{H}\sum_{n=1}^N u^n$. We provide the following performance guarantee of Algorithm 3:

[3] The use of $\parallel$ highlights its distinction from standard conditioning on $s_{1:H}$.

Theorem 26. *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 26 shows that the interactive IL Algorithm 3 matches the trajectory-wise sample complexity of behavior cloning in [15]. In contrast, prior state-of-the-art analysis of interactive IL algorithms [15, Appendix C.2] gives sample complexity results that are in general worse than behavior cloning. [4]

For the proof of Theorem 26, we introduce a new notion of decoupled Hellinger estimation error:

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

$\mathrm{OnEst}_N^{\mathrm{Traj}}$ decouples the dependence between the state sequence and the distribution of action sequence induced by the learner. Perhaps surprisingly, 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.2 Decoupling State and Action Sequences by Decoupled Hellinger Distance

In this section, we demonstrate that similar to $D_{\mathrm{H}}^2\left(\mathbb{P}^{\pi}(s_{1:H}, a_{1:H}), \mathbb{P}^{\pi^{\mathrm{E}}}(s_{1:H}, a_{1:H})\right)$ [15], the decoupled Hellinger distance $\mathbb{E}^{\pi}\left[D_{\mathrm{H}}^2(\pi(s_{1:H}), \pi^{\mathrm{E}}(s_{1:H}))\right]$ that decouples states and actions is also proportionally lower bounded by a constant factor of $\rho(\pi \parallel \pi^{\mathrm{E}})$. The following two lemma shows that such relationship holds for both Markovian policies and their first-step mixings.

Lemma 27. *Let π^{E} be a deterministic policy, and let π be an Markovian policy. Then we have*

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

Lemma 28. *Let π^{E} be a deterministic policy, and let π_u be a first-step mixing of Markovian policies. Then we have*

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

To prove these two lemmas, we first prove a special case, i.e. Lemma 29 with first-step mixing of deterministic policies. To facilitate the proofs, we introduce the following additional notations:

- Let $\mathcal{B}^{\mathrm{Det}}$ denote 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 taken by ν at (s, 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.

Lemma 29. *Let π^{E} be a deterministic Markovian policy, and let π_u be a first-step mixing of deterministic Markovian policies (elements of $\mathcal{B}^{\mathrm{Det}}$). Then we have that*

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

[4] For [15, 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 $\mu < R/H < 1$ in the practically-popular parameter sharing settings ($|\mathcal{B}_h| = |\mathcal{B}|$).

The key idea in the following proof is to lower bound $\mathbb{E}^{\pi_u}\left[D_{\mathrm{H}}^2(\pi_u(s_{1:H}), \pi^{\mathrm{E}}(s_{1:H}))\right]$, which reflects the asymmetric roles of the two appearances of π_u's, using a symmetric formulation via function F (as shown in (8)).

Proof. Recall the first-step mixing policy in Definition 8, we start by rewriting

$$\begin{aligned}\rho(\pi_u \parallel \pi^{\mathrm{E}}) =& \mathbb{E}^{\pi_u}\left[\mathbb{I}\left\{\exists h : a_h \neq \pi_h^{\mathrm{E}}(s_h)\right\}\right] \\ =& \sum_{\nu \in \mathcal{B}^{\mathrm{Det}}} u(\nu) \sum_{s_{1:H}} \mathbb{P}^{\nu}(s_{1:H}, a_{1:H}) \mathbb{I}\left\{\exists h : a_h \neq \pi_h^{\mathrm{E}}(s_h)\right\} \\ =& \sum_{\nu \in \mathcal{B}^{\mathrm{Det}}} u(\nu) \rho(\nu \parallel \pi^{\mathrm{E}}),\end{aligned} \tag{5}$$

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

Next, we turn to analyzing $D_{\mathrm{H}}^2(\pi_u(s_{1:H}), \pi^{\mathrm{E}}(s_{1:H}))$. 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 34, yielding:

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

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,

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

which implies:

$$1 - u(\mathcal{B}^{\mathrm{E}}(s_{1:H})) \leq D_{\mathrm{H}}^2(\pi_u(s_{1:H}), \pi^{\mathrm{E}}(s_{1:H})). \tag{6}$$

Therefore, by taking expectation over $s_{1:H} \sim \mathbb{P}^{\pi_u}$ in Eq. (6),

$$\sum_{s_{1:H}} \mathbb{P}^{\pi_u}(s_{1:H})(1 - u(\mathcal{B}^{\mathrm{E}}(s_{1:H}))) \leq \mathbb{E}^{\pi_u}\left[D_{\mathrm{H}}^2(\pi_u(s_{1:H}), \pi^{\mathrm{E}}(s_{1:H}))\right]. \tag{7}$$

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:

$$\begin{aligned}(*) =& \sum_{s_{1:H}} \sum_{\nu \in \mathcal{B}^{\mathrm{Det}}} u(\nu) \mathbb{P}^{\nu}(s_{1:H}) \sum_{\nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu') \mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right] \\ =& \sum_{\nu, \nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu) u(\nu') \sum_{s_{1:H}} \mathbb{P}^{\nu}(s_{1:H}) \mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right] \\ =& \sum_{\nu, \nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu) u(\nu') F(\nu; \nu'; \pi^E) \\ =& \frac{1}{2} \sum_{\nu, \nu' \in \mathcal{B}^{\mathrm{Det}}} u(\nu) u(\nu') \left(F(\nu; \nu'; \pi^E) + F(\nu; \nu'; \pi^E)\right),\end{aligned} \tag{8}$$

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 30 (stated below),

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

Combining the above two inequalities with Eq (7) we conclude the proof by

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

Lemma 30 (Symmetric Evaluation Lemma)**.** *Given deterministic Markovian policies ν, ν', and π^{E}, the following holds*

$$\frac{1}{2} \cdot \left(\rho(\nu \parallel \pi^{\mathrm{E}}) + \rho(\nu' \parallel \pi^{\mathrm{E}}) \right) \leq F(\nu; \nu'; \pi^{\mathrm{E}}) + F(\nu'; \nu; \pi^{\mathrm{E}}). \tag{9}$$

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 $\nu_h(s_h) \neq \pi_h^{\mathrm{E}}(s_h)$. 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:

$$\begin{aligned}
\mathcal{X}_2^h :=& \{s_{1:H} \mid \nu \in \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1})\}; \\
\mathcal{X}_3^h :=& \{s_{1:H} \mid \nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1})\}; \\
\mathcal{X}_4^h :=& \{s_{1:H} \mid \nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:h}), \nu \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1}), \nu' \in \mathcal{B}^{\mathrm{E}}(s_{1:h-1})\}.
\end{aligned} \tag{10}$$

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 $\{s_{1:h-1}|s_{1:h} \in \tilde{\mathcal{X}}_i^h\}$,

$$\begin{aligned}\sum_{s_{1:h}\in\tilde{\mathcal{X}}_i^h} \mathbb{P}^\nu(s_{1:h}) &= \sum_{s_{1:h}\in\tilde{\mathcal{X}}_i^h} P_0(\mathbf{E}') \prod_{h'=1}^{h-1} P_h(s_{h+1}|s_h, \nu_h(s_h)) \\ &= \sum_{s_{1:h}\in\tilde{\mathcal{X}}_i^h} P_0(\mathbf{E}') \prod_{h'=1}^{h-1} P_h(s_{h+1}|s_h, \nu'_h(s_h)) = \sum_{s_{1:h}\in\tilde{\mathcal{X}}_i^h} \mathbb{P}^{\nu'}(s_{1:h}).\end{aligned} \tag{11}$$

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

$$\sum_{s_{1:H}\in\mathcal{X}_2} \left(\mathbb{P}^\nu(s_{1:H})\mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right] + \mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left[\nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right]\right) \geq \sum_{s_{1:H}\in\mathcal{X}_2} \mathbb{P}^\nu(s_{1:H}), \tag{12}$$

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

$$\sum_{s_{1:H}\in\mathcal{X}_3} \left(\mathbb{P}^\nu(s_{1:H})\mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right] + \mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left[\nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right]\right) \geq \sum_{s_{1:H}\in\mathcal{X}_3} \mathbb{P}^{\nu'}(s_{1:H}) = \sum_{s_{1:H}\in\mathcal{X}_3} \mathbb{P}^\nu(s_{1:H}). \tag{13}$$

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

$$\begin{aligned}&\sum_{s_{1:H}\in\mathcal{X}_4} \left(\mathbb{P}^\nu(s_{1:H})\mathbb{I}\left[\nu' \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right] + \mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left[\nu \notin \mathcal{B}^{\mathrm{E}}(s_{1:H})\right]\right) \\ &= \sum_{s_{1:H}\in\mathcal{X}_4} \left(\mathbb{P}^\nu(s_{1:H}) + \mathbb{P}^{\nu'}(s_{1:H})\right) \geq \sum_{s_{1:H}\in\mathcal{X}_4} \mathbb{P}^\nu(s_{1:H}).\end{aligned} \tag{14}$$

Now, we combine Eqs. (12), (13), (14) and observe that

$$\sum_{s_{1:H}\in\mathcal{X}_2} \mathbb{P}^\nu(s_{1:H}) + \sum_{s_{1:H}\in\mathcal{X}_3} \mathbb{P}^\nu(s_{1:H}) + \sum_{s_{1:H}\in\mathcal{X}_4} \mathbb{P}^\nu(s_{1:H}) \geq \frac{1}{2} \sum_{s_{1:H}\in\mathcal{X}_2\cup\mathcal{X}_3\cup\mathcal{X}_4} \left(\mathbb{P}^\nu(s_{1:H}) + \mathbb{P}^{\nu'}(s_{1:H})\right), \tag{15}$$

which implies

$$F(\nu; \nu'; \pi^{\mathrm{E}}) + F(\nu'; \nu; \pi^{\mathrm{E}}) \geq \frac{1}{2} \sum_{s_{1:H}\in\mathcal{X}_2\cup\mathcal{X}_3\cup\mathcal{X}_4} \left(\mathbb{P}^\nu(s_{1:H}) + \mathbb{P}^{\nu'}(s_{1:H})\right). \tag{16}$$

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

$$
\begin{aligned}
\sum_{s_{1:H}\in\mathcal{X}_2\cup\mathcal{X}_3\cup\mathcal{X}_4}\left(\mathbb{P}^{\nu}(s_{1:H})+\mathbb{P}^{\nu'}(s_{1:H})\right) =& \sum_{s_{1:H}}\mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left\{\exists h: \nu_h(s_h)\neq\pi_h^{\mathrm{E}}(s_h) \text{ or } \nu'_h(s_h)\neq\pi_h^{\mathrm{E}}(s_h)\right\} \\
&+\sum_{s_{1:H}}\mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left\{\exists h: \nu_h(s_h)\neq\pi_h^{\mathrm{E}}(s_h) \text{ or } \nu'_h(s_h)\neq\pi_h^{\mathrm{E}}(s_h)\right\} \\
\geq& \sum_{s_{1:H}}\mathbb{P}^{\nu}(s_{1:H})\mathbb{I}\left\{\exists h: \nu_h(s_h)\neq\pi_h^{\mathrm{E}}(s_h)\right\} \\
&+\sum_{s_{1:H}}\mathbb{P}^{\nu'}(s_{1:H})\mathbb{I}\left\{\exists h: \nu'_h(s_h)\neq\pi_h^{\mathrm{E}}(s_h)\right\} \\
=&\rho(\nu\parallel\pi^{\mathrm{E}})+\rho(\nu'\parallel\pi^{\mathrm{E}}),
\end{aligned} \tag{17}
$$

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 (17) into (16).

E.2.1 Proof of Lemma 27

Proof. The key to this proof is showing that any Markovian policy π is equivalent, in terms of action distribution on any state sequence, to a first-step mixing of a set of deterministic Markovian policies. This leads to equivalence on trajectory distribution and decoupled Hellinger distance. To clarify further, we present the following claim, which allows us to apply guarantees for mixtures of deterministic policies in Lemma 29.

Claim 31. *For a Markovian policy π, there exists a first-step mixing of deterministic policy π_u such that for any $s_{1:H}\in\mathcal{S}^H$, 1. $\pi(s_{1:H})=\pi_u(s_{1:H})$, and 2. $\mathbb{P}^{\pi}(s_{1:H})=\mathbb{P}^{\pi_u}(s_{1:H})$.*

Given an MDP with finite state space size S and action space size A, the set of all deterministic, Markovian policies, denoted by $\mathcal{B}^{\mathrm{Det}}$, contains A^{SH} deterministic policies, which can be indexed by a tuple of actions $(a_{h,s})_{h\in[H],\, s\in\mathcal{S}}$.

To construct policy π_u, we will set the weight vector u such that its weight on policy ν indexed by $(a_{h,s})_{h\in[H],s\in\mathcal{S}}$ as:

$$
u(\nu)=\prod_{h=1}^{H}\prod_{s\in\mathcal{S}}\pi_h(a_{h,s}|s) \tag{18}
$$

It can be easily verified by 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

$$
\begin{aligned}
\pi_u(a_{1:H}\parallel s_{1:H}) &= \sum_{\nu\in\mathcal{B}^{\mathrm{Det}}}u(\nu)\prod_{h=1}^{H}\nu(a_h|s_h) \\
&= \sum_{(a'_{h,s})_{h\in[H],s\in\mathcal{S}}}\prod_{h=1}^{H}\prod_{s\in\mathcal{S}}\pi_h(a'_{h,s}|s)\prod_{h=1}^{H}\mathbb{I}\left[a'_{h,s_h}=a_h\right] \\
&= \sum_{(a'_{h,s})_{h\in[H],s\neq s_h}}\prod_{h=1}^{H}\prod_{s\neq s_h}\pi_h(a'_{h,s}|s)\sum_{(a'_{h,s})_{h\in[H],s=s_h}}\prod_{h=1}^{H}\pi_h(a'_{h,s_h}|s_h)\prod_{h=1}^{H}\mathbb{I}\left[a'_{h,s_h}=a_h\right] \\
&= \sum_{(a'_{h,s})_{h\in[H],s\neq s_h}}\prod_{h=1}^{H}\prod_{s\neq s_h}\pi_h(a'_{h,s}|s)\prod_{h=1}^{H}\pi_h(a_h|s_h) \\
&= \prod_{h=1}^{H}\pi_h(a_h|s_h)=\pi(a_{1:H}\parallel s_{1:H}).
\end{aligned} \tag{19}
$$

Since this holds for any action sequence $a_{1:H} \in \mathcal{A}^H$, we derive the first part of Claim 31 that $\pi(s_{1:H}) = \pi_u(s_{1:H})$. The second item follows from the first item in combination with Lemma 40.

We conclude that for the π_u in the statement of the claim,

$$\mathbb{E}^{\pi}\left[D^2_{\mathrm{H}}(\pi(s_{1:H}), \pi^{\mathrm{E}}(s_{1:H}))\right] = \mathbb{E}^{\pi_u}\left[D^2_{\mathrm{H}}(\pi_u(s_{1:H}), \pi^{\mathrm{E}}(s_{1:H}))\right].$$

Finally, the proof follows by applying Lemma 29 to π_u.

E.2.2 Proof of Lemma 28

Proof. By Claim 31, any Markovian policy can be viewed as a first-step mixing of A^{SH} deterministic policies from $\mathcal{B}^{\mathrm{Det}}$, then 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 follows by applying Lemma 29.

E.3 New Guarantees for DAgger Variant with Trajectory-wise Annotation

Recall that we have defined decoupled Hellinger estimation error:

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

In the following, 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 26.

Lemma 32. *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.*

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

Proof. By Lemma 27 and Lemma 28, for each π^n, whether it is Markovian or a first-step mixing of Markovian policies, the following holds:

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

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

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

where we apply the fact that $\hat{\pi}$ is a first-step mixing of $\{\pi^n\}_{n=1}^N$ with uniform weights. Finally, we conclude the proof by applying Lemma 36.

Theorem 33 (Theorem 26 Restated)**.** *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 to exponential weight algorithm (see Proposition 37). Then, with probability at least $1 - \delta$,*

$$\mathrm{OnEst}_N^{\mathrm{Traj}} \le \log(B) + 2\log(1/\delta),$$

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

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

Proof. The proof closely follows Proposition C.2 in [15], tailored for another DAgger variant. However, in this case, 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

$$\ell^n(\pi) = \log(1/\pi_u(a^{n,*}_{1:H} \parallel s^n_{1:H})) = \log\left(\frac{1}{\sum_{\pi\in\mathcal{B}} u(\pi)\pi(a^{n,*}_{1:H} \parallel s^n_{1:H})}\right).$$

It can be observed that ℓ^n's are 1-exp-concave. Therefore, implementing $\mathbb{A}$ using the exponential weights algorithm (Proposition 37) ensures that the following bound holds almost surely:

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

Then, Lemma 38 with $x^n = s^n_{1:H}$, $y^n = a^{*,n}_{1:H}$, $g_* = \pi^{\mathrm{E}}$, and $\mathcal{H}^n = \{o^{n'}\}^n_{n'=1}$, where $o^n = \left(\mathbf{E}'^n, a^n_1, a^{*,n}_1, \dots, s^n_H, a^n_H, a^{*,n}_H\right)$, implies that with probability at least $1-\delta$,

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

Finally, the second part of the theorem follows by applying Lemma 32.

F Auxiliary Results

Lemma 34. *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*

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

Lemma 35 (Performance Difference Lemma [21][47])**.** *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)*

$$J(\pi) - J(\pi^E) \leq H \cdot \lambda(\pi^{\mathrm{E}} \parallel \pi).$$

- *suppose $(\mathcal{M}, \pi^{\mathrm{E}})$ is μ-recoverable, then*

$$J(\pi) - J(\pi^E) \leq \mu \cdot \lambda(\pi \parallel \pi^{\mathrm{E}}).$$

Lemma 36 (Lemma D.2. of [15])**.** *For all (potentially stochastic) policies π and π', it holds that*

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

Proposition 37 (Proposition 3.1 of [7])**.** *Suppose $\{\ell^n(u)\}^N_{n=1}$ 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:

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

Lemma 38 (Restatement of Lemma A.14 in [16]). *Under the realizbility assumption, where there exists $g_\star := g_{i_\star} \in \mathcal{G}$ such that for all $n \in [N]$,*

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

where $\mathcal{H}^{n-1}$ denotes all histories at the beginning of round n.

Then, for any estimation algorithm and any $\delta \in (0,1)$, with probability at least $1-\delta$,

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

where $\ell^n_{\log}(g) := \log\left(1/g\left(y^n \mid x^n\right)\right)$, and $\mathbb{E}_n[\cdot] := \mathbb{E}[\cdot \mid \mathcal{H}^n]$.

We have the following well-known lemma for causally-conditioned probability (e.g. [78]).

Definition 39. *The causally-conditioned probability of state sequence $s_{1:H}$ given action sequence $a_{1:H-1}$, is defined as*

$$\mathbb{P}^{\mathcal{M}}(s_{1:H} \parallel a_{1:H-1}) = P_0(\mathbf{E}') \prod_{h=1}^{H-1} P_h(s_{h+1} \mid s_h, a_h)$$

Lemma 40. *For any Markovian policy π,*

$$\mathbb{P}^{\pi}(s_{1:H}, a_{1:H}) = \mathbb{P}^{\mathcal{M}}(s_{1:H} \parallel a_{1:H-1}) \cdot \pi(a_{1:H} \parallel s_{1:H}), \tag{20}$$

and for any first-step mixing of Markovian policy π_u,

$$\mathbb{P}^{\pi_u}(s_{1:H}, a_{1:H}) = \mathbb{P}^{\mathcal{M}}(s_{1:H} \parallel a_{1:H-1}) \cdot \pi_u(a_{1:H} \parallel s_{1:H}). \tag{21}$$

Proof. Eq. (20) follows by noticing that both sides are equal to

$$P_0(\mathbf{E}') \prod_{h=1}^{H-1} P_h(s_{h+1} \mid s_h, a_h) \prod_{h=1}^{H} \pi_h(a_h \mid s_h).$$

Eq. (21) follows by noticing that both sides are equal to

$$\sum_{\nu} u(\nu) P_0(\mathbf{E}') \prod_{h=1}^{H-1} P_h(s_{h+1} \mid s_h, a_h) \prod_{h=1}^{H} \nu_h(a_h \mid s_h).$$

G Experiment Details

We compare WARM-STAGGER against Behavior Cloning (BC) and STAGGER on continuous-control tasks from OpenAI Gym MuJoCo [70, 6] 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 framework [5] (https://github.com/xkianteb/dril), with modifications to support online learning. The continuous control environments used in our experiments are: "HalfCheetahBulletEnv-v0", "AntBulletEnv-v0", "Walker2DBulletEnv-v0", and "HopperBulletEnv-v0". We include an anonymous link to our implementation here: `https://github.com/anonymous-submi/neurips2025`.

Environments and Expert Policies. We use four MuJoCo environments: Ant, Hopper, HalfCheetah, and Walker2D. The expert policy is a deterministic MLP pretrained via TRPO [50, 51], with two hidden layers of size 64.

Learner Architecture. The learner uses the same MLP architecture as the expert. The output is a diagonal Gaussian policy:

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

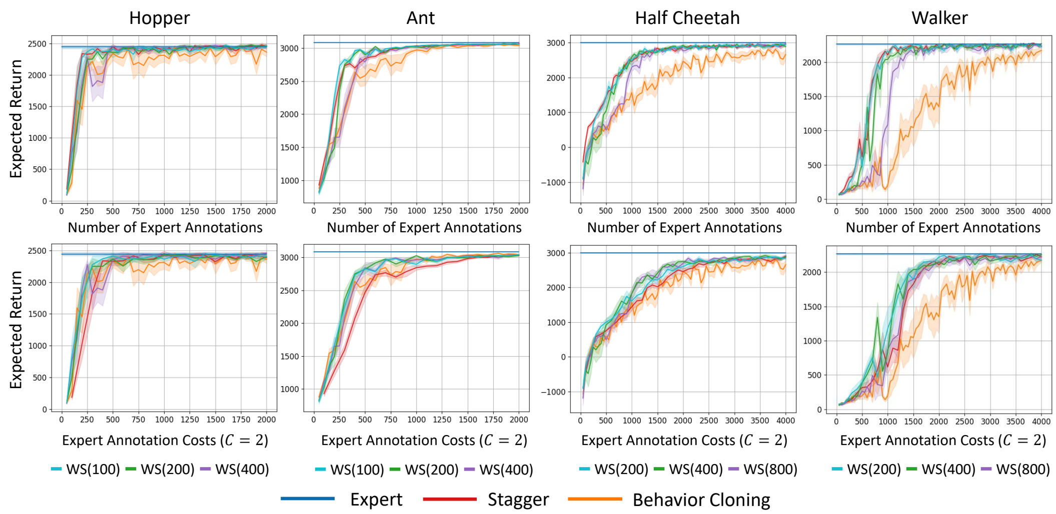

Figure 4: Sample and cost efficiency on MuJoCo tasks. The top row shows expected return vs. number of annotations ($C = 1$); the bottom row shows performance under a cost-aware setting ($C = 2$). WARM-STAGGER (WS) is initialized with 1/20, 1/10, or 1/5 of the samples as offline demonstrations. It matches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

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

Learning Protocols. For BC, we reveal expert state-action pairs sequentially along expert trajectories. For STAGGER, each round rolls out the latest policy, samples a state uniformly from the trajectory, queries the expert action, and updates immediately.

For WARM-STAGGER, we begin with BC and switch to STAGGER after a predefined number of offline examples. We use switch points of 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 cost 1 to each offline state-action pair and either cost 1 or cost 2 to each interactive query. We run each method for 10 random seeds. Every 50 rounds, we evaluate the current policy by running 25 full-episode rollouts and reporting the average cumulative reward.

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 the policy class is expressive enough (so that the problem is not exactly realizable but still meaningful). For example, [31] observed that with nonrealizable stochastic experts, DAgger variants outperform BC, and exhibit learning curves similar to ours.

G.1 Additional Experiment Plots

In this section, we present extended experiment results with longer training horizons. As shown in Figure 4, we allocate a total annotation budget of 2000 samples for Hopper and Ant, and 4000 samples for HalfCheetah and Walker. This complements the main paper by showing the full training curves without zooming in on early rounds. 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 $C = 2$.

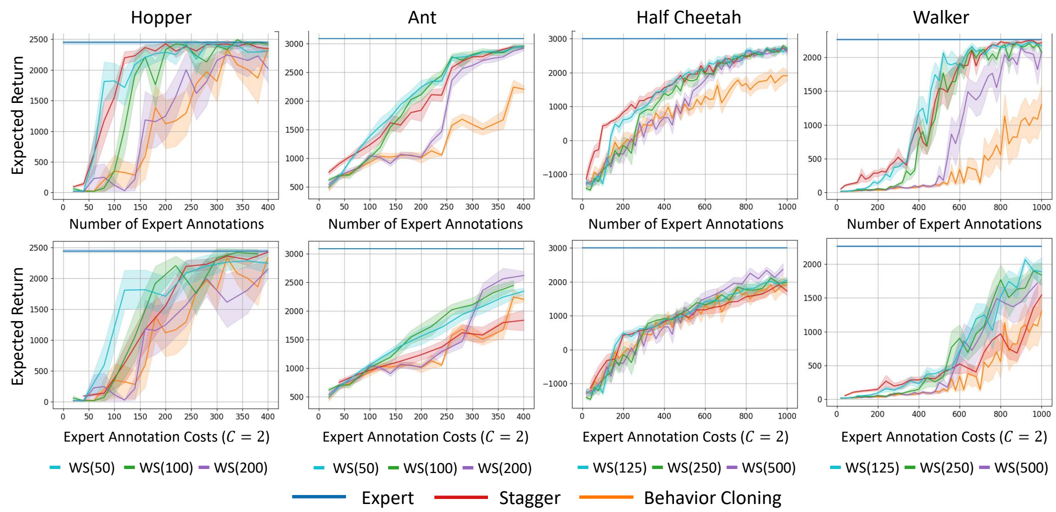

Figure 5: Performance comparison under MSE loss across MuJoCo tasks. Results show that WARM-STAGGER (WS) achieves comparable sample efficiency and performance to the log loss setting, with improved training stability. Each curve represents the average over 10 seeds.

G.2 Experiment with MSE Loss

Additionally, we evaluate our algorithms using mean squared error (MSE) as the loss function. All training settings remain the same as in the log loss experiments, except that we use a learning rate of 2.5×10^{-4}. As shown in 5, we observe qualitatively similar results to those under log loss, consistent with prior observations in [15], with the added benefit of more stable training dynamics.

G.3 Additional Experiments with Algorithm 3

For completeness, we evaluate TRAGGER and its warm-start version as shown in Algorithm 4 as follows on continuous control tasks and the MDP example the same as in Figure 2. The main difference between WARM-TRAGGER and WARM-STAGGER is by changing interactive annotation oracle from state wise annotation to trajectory wise annotation, which exhibits significant difference as shown in Figure 6. Especially for Ant and Half-cheetah, the sample efficiency ($C = 1$) of TRAGGER and WARM-TRAGGER are sigificantly worse than STAGGER, which is due to the cold start problem, where early dagger rollouts does not have good visitation coverage but still have to go on until the end of trajectory. However, this is not the case for STAGGER, which only samples one state per trajectory, unleashing the benefit of interactive learning. Also observe that the performance of TRAGGER and WARM-TRAGGER are closer to STAGGER in Hopper and Walker, which is due to the hard reset induced by the environment if the learner performs badly and triggers the terminate state (unlikely in Ant and Half-cheetah), enabling a shorter trajectory early stop and more efficient learning. Combining these observarion, it is natural to see the middle groud between full trajectory annotation and single state annotation, which is batch query, for example sampling 50 states per trajectory, which has been implemented by [31] with similar performance.

For completeness, we evaluate TRAGGER and its warm-start variant (WARM-TRAGGER) as shown in Algorithm 4, on continuous control tasks and the same MDP setup 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 Half-Cheetah, the sample efficiency ($C = 1$) of TRAGGER and WARM-TRAGGER is significantly worse than that of STAGGER, due to the cold-start problem: early DAgger rollouts have poor state coverage but must still proceed until the end of each trajectory. In contrast, STAGGER only samples one state per trajectory, thus better leveraging interactive feedback.

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

Input: MDP $\mathcal{M}$, trajectory-wise expert annotation oracle $\mathcal{O}^{\mathrm{Traj}}$, Markovian policy class $\mathcal{B}$, online learning oracle $\mathbb{A}$, offline expert dataset D_{off} of size N_{off}, online budget N_{int}

Initialize $\mathbb{A}$ with policy class $\mathcal{B}_{\mathrm{bc}} := \{\pi \in \mathcal{B} : \pi(s) = \pi^{\mathrm{E}}(s)\ , \forall s \in D_{\mathrm{off}}\}$.

for $n = 1, \ldots, N_{\mathrm{int}}/H$ **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

$$\ell^n(\pi) := \log\left(\frac{1}{\pi^n(a^{*,n}_{1:H} \parallel s^n_{1:H})}\right). \tag{22}$$

end for

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

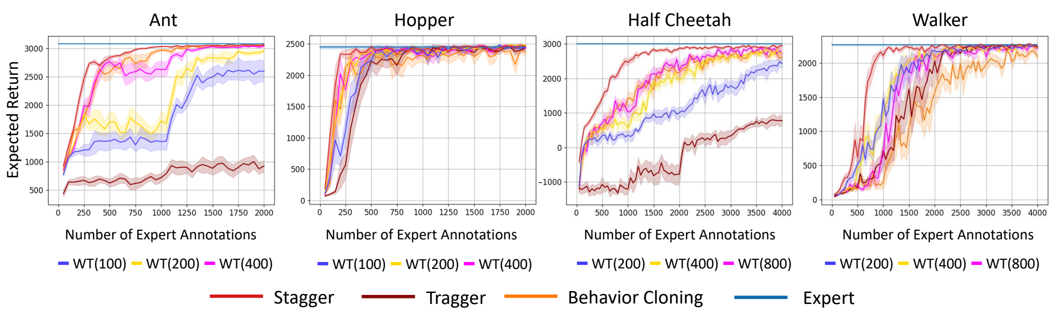

Figure 6: Sample efficiency of algorithms on MuJoCo tasks, showing expected return vs. number of annotations ($C = 1$). WARM-TRAGGER (WT) is initialized with 1/8, 1/4, or 1/2 of the total annotation budget as offline demonstrations. Although the performance of WARM-TRAGGER improves with more offline 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 annotation.

For Hopper and Walker, however, TRAGGER and WARM-TRAGGER achieve performance closer to STAGGER. This is because these environments inherit hard resets when the agent fails, effectively truncating poor trajectories and improving 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 [31] with comparable results.

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

However, the advantage of state-wise annotation does not hold in general; under certain MDP designs, trajectory-wise annotation can achieve nearly identical performance, as shown in Figure 8.

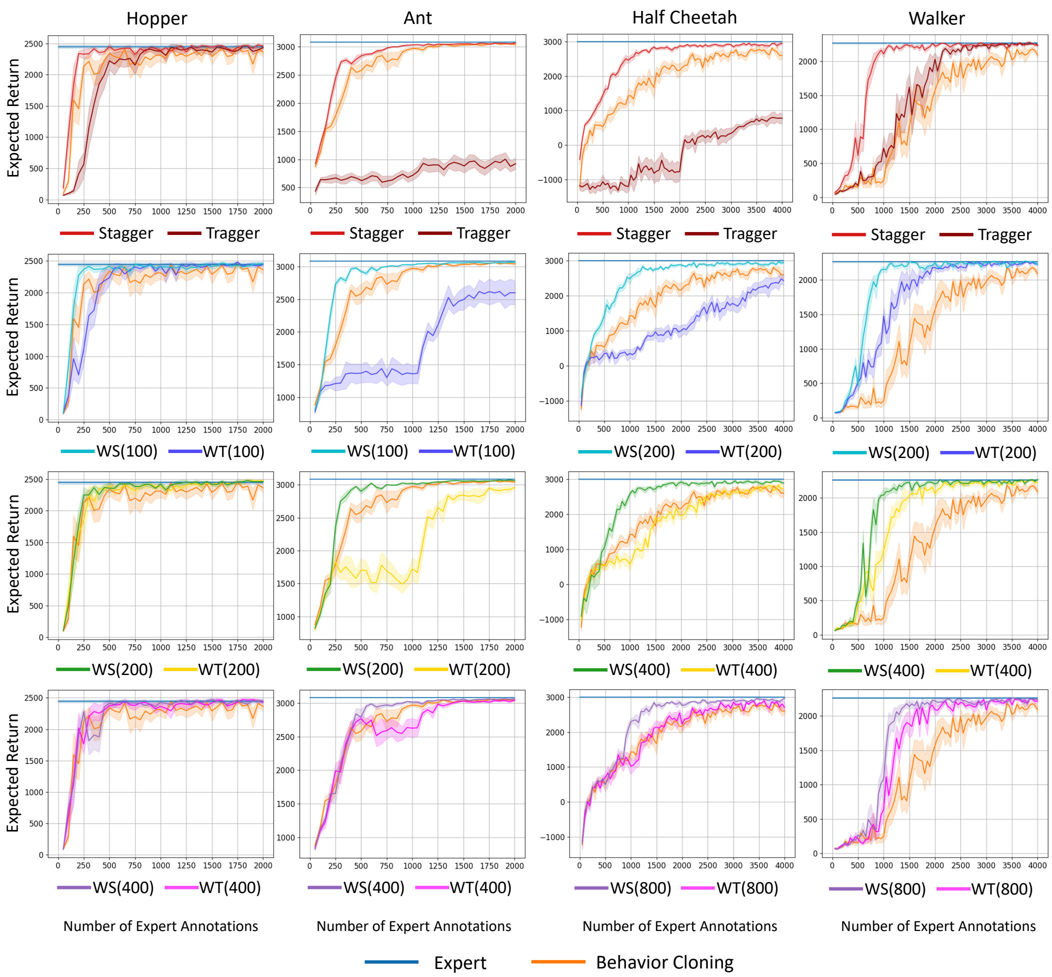

Figure 7: Head-to-head sample efficiency comparison between TRAGGER and STAGGER, and between WARM-TRAGGER and WARM-STAGGER under different offline demonstration budgets. STAGGER and WARM-STAGGER consistently outperform TRAGGER and WARM-TRAGGER. The performance gap narrows as the offline budget increases, effectively alleviating the cold-start problem suffered by TRAGGER.

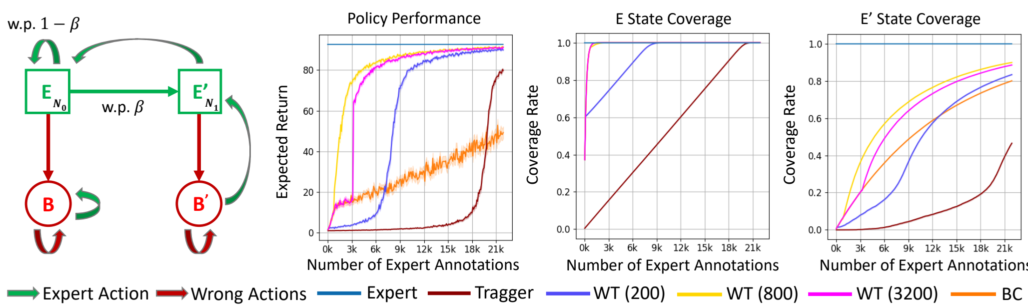

Figure 8: MDP construction and simulation results of algorithms with rewards assigned only in $\mathbf{E}$. Similar to Figure 2, we evaluate WARM-TRAGGER (WT) with 200, 800, 3200 offline (state, expert action) pairs. All methods are evaluated under equal total annotation cost with $C = 1$. With 800 offline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