How To Guide Your Learner: Imitation Learning with Active Adaptive Expert Involvement

Xu-Hui Liu*
Nanjing University
Nanjing, China
liuxh@lamda.nju.edu.cn

Feng Xu*
Nanjing University
Nanjing, China
xufeng@lamda.nju.edu.cn

Xinyu Zhang
Nanjing University
Nanjing, China
zhangxinyu@lamda.nju.edu.cn

Tianyuan Liu
Nanjing University
Nanjing, China
191300034@smail.nju.edu.cn

Shengyi Jiang
The University of Hong Kong
Hong Kong, China
syjiang@cs.hku.hk

Ruifeng Chen
Nanjing University
Nanjing, China
chenrf@lamda.nju.edu.cn

Zongzhang Zhang
Nanjing University
Nanjing, China
zzzhang@nju.edu.cn

Yang Yu†
Nanjing University
Nanjing, China
Peng Cheng Laboratory
ShenZhen, China
yuy@nju.edu.cn

ABSTRACT

Imitation learning aims to mimic the behavior of experts without explicit reward signals. Passive imitation learning methods which use static expert datasets typically suffer from compounding error, low sample efficiency, and high hyper-parameter sensitivity. In contrast, active imitation learning methods solicit expert interventions to address the limitations. However, recent active imitation learning methods are designed based on human intuitions or empirical experience without theoretical guarantee. In this paper, we propose a novel active imitation learning framework based on a teacher-student interaction model, in which the teacher's goal is to identify the best teaching behavior and actively affect the student's learning process. By solving the optimization objective of this framework, we propose a practical implementation, naming it AdapMen. Theoretical analysis shows that AdapMen can improve the error bound and avoid compounding error under mild conditions. Experiments on the MetaDrive benchmark and Atari 2600 games validate our theoretical analysis and show that our method achieves near-expert performance with much less expert involvement and total sampling steps than previous methods. The code is available at https://github.com/liuxhym/AdapMen.

KEYWORDS

Reinforcement Learning; Imitation Learning; Human in the Loop

ACM Reference Format:

Xu-Hui Liu, Feng Xu, Xinyu Zhang, Tianyuan Liu, Shengyi Jiang, Ruifeng Chen, Zongzhang Zhang, Yang Yu. 2023. How To Guide Your Learner: Imitation Learning with Active Adaptive Expert Involvement. In *Proc.of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

*Equal Contribution †Corresponding Author.

Proc.of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

1 INTRODUCTION

Imitation Learning (IL) [9, 33, 35] aims to learn a policy from expert demonstrations with no explicit task-relevant knowledge like reward and transition. IL has achieved huge success in a variety of domains, including games [35] and recommendation systems [4, 36].

The traditional IL method Behavior Cloning (BC) [33] imitates expert behaviors via supervised learning. Although BC works fine in simple environments, it requires a lot of data and small errors compound quickly when the learned policy deviates from the states in the expert dataset. This issue can be formalized by the sub-optimality bound of the learned policy, which is $\tilde{O}(\epsilon_b H^2)$ for BC [33], where ϵ_b is the optimization error, H is the horizon of the Markov Decision Processes (MDPs) and $\tilde{O}$ means the constant and log terms are omitted. The quadratic dependency on H is known as the *compounding error* issue.

To tackle the compounding error issue, Apprenticeship Learning (AL) [1, 10] and Adversarial Imitation Learning (AIL) [6, 9, 16, 17] algorithms introduce interactions with environment. They first infer a reward function from expert demonstrations, then learn a corresponding policy by Reinforcement Learning (RL). The sub-optimality bound is then reduced to $\tilde{O}(\epsilon_g H)$ [39], where ϵ_g is the optimization error of AL and AIL. From another perspective, DAgger [35] attributes the compounding error issue to the difference between the train distribution and test distribution. Thus, DAgger queries the expert for action labels corresponding to each state visited by the learner [35].

Despite the reduction of the order of H, the complicated optimization process of AL and AIL leads to even worse sample complexity than BC [40]. Additionally, these algorithms are highly sensitive to hyper-parameters and are hard to converge in practice [43]. DAgger also relies on an additional assumption that the learner can recover

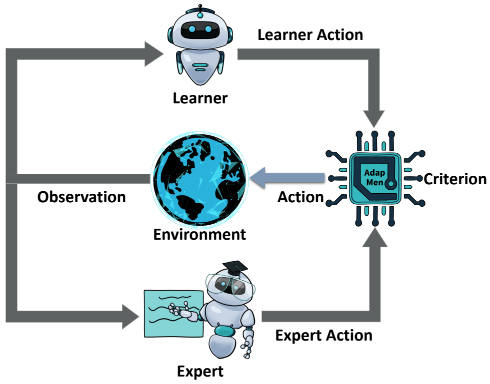


Figure 1: Framework of AdapMen

	Sub-optimality	Sample Complexity
BC	$\tilde{O}(\epsilon_b H^2)$	$\tilde{O}(\frac{\lvert S\rvert H^2}{\epsilon})$
AIL	$\tilde{O}(\epsilon_g H)$	$\tilde{O}(\frac{\lvert S\rvert H^2}{\epsilon^2})$
DAgger	$\tilde{O}(\mu\epsilon_b H)$	$\tilde{O}(\frac{\mu\lvert S\rvert H}{\epsilon})$
AdapMen	$\tilde{O}(\epsilon_b H)$	$\tilde{O}(\frac{\lvert S\rvert H}{\epsilon})$

Table 1: Theoretical Guarantee of IL Methods. $\tilde{O}$ means the log term of H is omitted.

from mistakes made by itself to a certain extent, which is known as the μ-recoverability condition on the MDPs. [31] proves that DAgger has a better theoretical guarantee than BC under such an assumption while [32] shows a negative result for general cases. Moreover, the assumption is satisfied only when any sub-optimal action leads to little performance degradation, which can be impractical, e.g., in risk-sensitive environments [31, 32]. Some recent methods [11, 15, 21, 24, 42] modify DAgger so that they only solicit expert interventions based on certain criteria. Though these methods achieve certain empirical success, there were no theoretical understanding of these methods and their design of intervention criteria is totally intuitive, hindering further algorithmic design.

To address the issues in previous methods, we study the IL problems from a new perspective. From experience, sometimes experts are not the best teachers. For example, many legendary players end up with controversial coaching careers. Experts advance disciplines, while teachers advance learners. Inspired by the idea of machine teaching [30, 44] , we formulate the IL process as a teacher-student framework. In this framework, the teacher decides what to teach and how to impart knowledge rather than simply correcting the student. With more attentive help from the teacher, the student agent can learn faster.

We formalize this intuition by introducing an optimization problem on minimizing the value loss of the learned policy. By solving the optimization problem in the framework, we obtain a novel imitation learning method **A**ctive a**da**ptive ex**p**ert involve**Men**t (AdapMen), where a teacher actively involves in the learner's interaction with the environment and adjusts its teaching behavior accordingly. The overall interaction structure is illustrated in Fig. 1, where the criterion and the expert is together viewed as the teacher. At each time step, a criterion calculated from expert actions judges whether to take the learner's action or ask the expert to take over control.

The sub-optimality and sample complexity bounds of AdapMen and other typical IL methods are listed in Tab. 1. Under mild conditions, AdapMen achieves no compounding error with much lower sample complexity than previous methods. To validate our theories, we also experimentally verify the validity of the assumption and demonstrate the power of AdapMen in several tasks.

2 RELATED WORK

Imitation Learning. The most traditional approach to imitation learning is Behavioral Cloning (BC) [2, 5, 33], where a classifier or regressor is trained to fit the behaviors of the expert. This simple form of IL suffers from high compounding error because of covariate shifts. By allowing the learner agent to further interact with the environment, Apprenticeship Learning (AL) [1, 10] infers a reward function from expert demonstrations by Inverse Reinforcement Learning (IRL) [27] and learns a policy with Reinforcement Learning (RL) using the recovered reward function. In this way, the learner can correct its behavior on unseen states to mitigate the compounding error issue. Recently, based on Generative Adversarial Network (GAN) [7], Adversarial Imitation Learning (AIL) [9, 16, 17] performs state-action distribution matching in an adversarial manner and has a stronger empirical performance than AL. Since AL and AIL have access to environment transitions, they are classified as known-transition methods. Notwithstanding the compounding error issue, this type of method is highly sensitive to hyper-parameters and hard to converge in practice [43]. Different from the known-transition methods, DAgger-style algorithms [34, 35, 38] address the covariate shift by querying the expert online. Without the min-max optimization in known-transition methods, DAgger-style algorithms tend to be more stable. However, these algorithms can only avoid compounding error under the μ-recoverability assumption, which is often not satisfied in risk-sensitive environments [31, 32]. Our method AdapMen is free from the μ-recoverability assumption and the hyper-parameters are automatically tuned.

Human-in-the-loop. Many works focus on incorporating human interventions in the training loop of RL or IL paradigms. DAgger [35] can be seen as one of the human-in-the-loop methods if the expert is a human. DAgger requires experts to provide action labels without being fully in control of the system, which can introduce safety concerns and is very likely to degrade the quality of the collected labels due to the loss of direct feedback. To address this challenge, a list of learning from intervention approaches have been proposed to empower humans to intervene and guide the learner agent to safe states. "Human-Gated" approaches [15, 21, 37] require humans to determine when the agent needs help and when to cede control, which is unreliable because of the high randomness of human behavior. In contrast, "Agent-Gated" approaches [11, 12, 24, 42] allow the learner agent to actively seek human interventions based on certain criteria including the novelty or the risk of the visited states. However, all of the criteria are heuristic without theoretical guarantees and the hyper-parameters are hard to tune. Our method AdapMen can actively involve in the interaction process and adaptively adjust its intervention probability.

3 BACKGROUND

Consider an MDP task denoted by $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, H, r, \rho)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function, H is the planning horizon, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and ρ is the distribution of initial states. Without loss of generality, we assume $r(s, a) \in [0, 1]$. A policy is defined as $\pi(\cdot \mid s)$, which outputs an action distribution. To facilitate later analysis, we introduce the state-action distribution at time step t as follows:

$$d_h^\pi(s,a) = \Pr\left(s_h = s, a_h = a | s_1 \sim \rho, a_t \sim \pi(s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t), t \in [h]\right),$$

where $[h] = \{1, 2, \dots, h\}$. We define $d^\pi = \frac{1}{H}\sum_{h=1}^H d_h^\pi$, which is the average distribution of states if we follow policy π for H steps.

In imitation learning, the reward function of a task is not accessible. Instead, the learner agent has access to an expert with policy π^*, and the goal is to recover the policy π^* by learning from labeled training data, e.g., state-action pairs generated by an expert agent. Following [39] and [32], we assume the expert policy is deterministic in the theoretical analysis, while it can be stochastic in practice.

To measure the quality of a learner policy, we define the *policy value* as

$$J(\pi) = \mathbb{E}\left[\sum_{h=1}^H r(s_h, a_h) | s_1 \sim \rho; \; a_h \sim \pi(\cdot|s_h), s_{h+1} \sim \mathcal{P}(\cdot|s_h, a_h), \forall h \in [H]\right].$$

This is the cumulative return for the learner agent in the task demonstrated by the expert. Accordingly, the quality of imitation learning is measured by the *sub-optimality gap*: $J(\pi^*) - J(\pi)$. We also introduce the Q-function at time step h:

$$Q_h^\pi(s,a) = \mathbb{E}\left[\sum_{t=h}^H r(s_t, a_t) | s_h = s, a_h = a; \; a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t), \forall t \in \{h+1, \dots, H\}\right].$$

For brevity, we use $Q_h^{\pi_1}(s, \pi_2)$ as a shorthand of $\mathbb{E}_{a \sim \pi_2} Q_h^{\pi_1}(s, a)$. Then, we have $J(\pi) = \mathbb{E}_{s \sim \rho} Q_H^\pi(s, \pi)$ and $J(\pi^*) = \mathbb{E}_{s \sim \rho} Q_H^{\pi^*}(s, \pi^*)$.

4 TEACHER-STUDENT INTERACTION MODEL

Given the inspiration that experts may not be the best teachers, we construct a teaching policy for the agent. In the learning process, the agent aims to mimic the teacher policy instead of the expert policy. This intuition can be formulated as the following optimization problem:

$$\min_{\pi'} \quad J(\pi^*) - J(\pi_{\pi'}) \quad \text{s.t. } \mathbb{E}_{s \sim \beta} \ell(s, \pi_{\pi'}, \pi') \le \epsilon_b, \tag{1}$$

where π' is the teaching policy, $\pi_{\pi'}$ is the corresponding learned student policy, β is the data distribution of the buffer that stores intervened samples, $\ell(s, \pi_{\pi'}, \pi')$ is the 0-1 loss, i.e., $\ell(s, \pi_{\pi'}, \pi') = 0$ if $\pi_{\pi'}(\cdot|s) = \pi'(\cdot|s)$ and $\ell(s, \pi_{\pi'}, \pi') = 1$ otherwise, and ϵ_b is the upper bound of the optimization loss. Intuitively, we aim to find a policy π' that generates data to not only correct the learner when it deviates from the desired behavior, but also helps it learn as quickly as possible.

Denote π as the policy before the policy optimization process, i.e., $\pi_{\pi'}$ is optimized from π. Because it is useless to store the data coinciding with the agent policy, a natural choice for the distribution of buffer is

$$\beta(s) = \frac{1}{H\delta} \sum_{h=1}^H \mathbb{I}(\pi(\cdot|s) \ne \pi'(\cdot|s)) d_h^{\pi'}(s). \tag{2}$$

That is, we only save the samples when π and π' behave differently. δ is the normalization factor for the distribution of the buffer, i.e.,

$$\delta = \sum_s \frac{1}{H} \sum_{h=1}^H \mathbb{I}(\pi'(\cdot|s) \ne \pi(\cdot|s)) d_h^{\pi'}(s) = \mathbb{E}_{s \sim d^{\pi'}} \mathbb{I}(\pi'(\cdot|s) \ne \pi(\cdot|s)). \tag{3}$$

Before solving this optimization problem, we introduce Lemma 4.1 for better understanding of the derivation.

Lemma 4.1 (Policy Difference Lemma [14]). *For any policies π_1 and π_2,*

$$J(\pi_1) - J(\pi_2) = \sum_{h=1}^H \mathbb{E}_{s \sim d_h^{\pi_1}} [Q_h^{\pi_2}(s, \pi_1) - Q_h^{\pi_2}(s, \pi_2)].$$

With this lemma, we rewrite the optimization objective as

$$\begin{aligned} J(\pi^*) - J(\pi_{\pi'}) &= J(\pi^*) - J(\pi') + J(\pi') - J(\pi_{\pi'}) \\ &\overset{(a)}{=} \sum_{h=1}^H \mathbb{E}_{s \sim d_h^{\pi'}} [Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi')] \\ &\quad + \sum_{h=1}^H \mathbb{E}_{s \sim d_h^{\pi'}} [Q_h^{\pi_{\pi'}}(s, \pi') - Q_h^{\pi_{\pi'}}(s, \pi_{\pi'})]. \end{aligned} \tag{4}$$

(a) is derived from Lemma 4.1. The minimization of the first term implies the teaching policy π' should be similar to the expert policy π^*, while the minimization of the second term implies π' should be close to $\pi_{\pi'}$. Note that we cannot determine π' simply from $\pi_{\pi'}$ since $\pi_{\pi'}$ is learned from π'. However, π' can be close to $\pi_{\pi'}$ if we assume $\pi_{\pi'}(\cdot|s) = \pi(\cdot|s)$ if $\pi(\cdot|s) = \pi'(\cdot|s)$. The assumption is straightforward because $\pi(\cdot|s) = \pi'(\cdot|s)$ implies we do not need to do optimization on state s, thus $\pi_{\pi'}$ stays unchanged on this state. Therefore, the overall optimization leads to a trade-off of π' between π^* and π.

To decompose the objective into a more tractable one, we assume Q_h^π can be upper-bounded by Δ, then

$$\begin{aligned} &\sum_{h=1}^H \mathbb{E}_{s \sim d_h^{\pi'}} [Q_h^{\pi_{\pi'}}(s, \pi') - Q_h^{\pi_{\pi'}}(s, \pi_{\pi'})] \\ &\le \Delta \sum_{h=1}^H \mathbb{E}_{s \sim d_h^{\pi'}} \mathbb{I}(\pi'(\cdot|s) \ne \pi_{\pi'}(\cdot|s)). \end{aligned} \tag{5}$$

In this way, the problem is transformed to increasing the probability that π' equals $\pi_{\pi'}$ and reducing the value degradation between π' and π^* simultaneously.

Applying the constraint in (1) to the right-hand side of Eq. (5) with the mentioned Δ, we have

$$\Delta \sum_{h=1}^{H} \mathbb{E}_{s\sim d_h^{\pi'}} \mathbb{I}(\pi'(\cdot|s) \neq \pi_{\pi'}(\cdot|s)) \tag{6}$$

$$\overset{(b)}{\leq} \Delta \sum_{h=1}^{H} \mathbb{E}_{s\sim d_h^{\pi'}} \mathbb{I}(\pi'(\cdot|s) \neq \pi(\cdot|s))\mathbb{I}(\pi'(\cdot|s) \neq \pi_{\pi'}(\cdot|s)) \tag{7}$$

$$\overset{(c)}{=} \Delta H\delta\, \mathbb{E}_{s\sim\beta}\mathbb{I}(\pi'(\cdot|s) \neq \pi_{\pi'}(\cdot|s)) \overset{(d)}{\leq} \Delta H\delta\epsilon_b \tag{8}$$

$$\overset{(e)}{=} \Delta\epsilon_b \sum_{h=1}^{H} \mathbb{E}_{s\sim d_h^{\pi'}} \mathbb{I}(\pi'(\cdot|s) \neq \pi(\cdot|s)), \tag{9}$$

where (b) uses the fact that $\pi'(\cdot|s) \neq \pi_{\pi'}(\cdot|s)$ implies $\pi'(\cdot|s) = \pi_{\pi'}(\cdot|s)$, (c) is derived from the definition of β in Eq. (2), (d) uses the condition $\mathbb{E}_{s\sim\beta}\ell(s, \pi_{\pi'}, \pi') \leq \epsilon_b$, and (e) is derived from the definition of δ in Eq. (3).

The first term of Eq. (4) can be rewritten as

$$\sum_{h=1}^{H} \mathbb{E}_{s\sim d_h^{\pi'}} [Q_h^{\pi^*}(s, \pi') - Q_h^{\pi^*}(s, \pi^*)] \tag{10}$$

$$= \sum_{h=1}^{H} \mathbb{E}_{s\sim d_h^{\pi'}} [Q_h^{\pi^*}(s, \pi') - Q_h^{\pi^*}(s, \pi^*)]\mathbb{I}(\pi'(\cdot|s) \neq \pi^*(\cdot|s)). \tag{11}$$

The added $\mathbb{I}(\pi'(\cdot|s) \neq \pi^*(\cdot|s))$ does not contribute to this term, because $Q_h^{\pi^*}(s, \pi') - Q_h^{\pi^*}(s, \pi^*) = 0$ when $\pi'(\cdot|s) = \pi^*(\cdot|s)$. The total value loss is composed of Eq. (9) and Eq. (11). Fixing the distribution $d^{\pi'}$, Eq. (9) equals 0 if $\pi'(\cdot|s) = \pi(\cdot|s)$ and Eq. (11) equals 0 if $\pi'(\cdot|s) = \pi^*(\cdot|s)$. Thus, the agent will suffer from a $Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi')$ value loss if $\pi'(\cdot|s) = \pi(\cdot|s)$, and suffer from a $\Delta\epsilon_b$ value loss if $\pi'(\cdot|s) = \pi^*(\cdot|s)$. In this way, proper choice of π' is

$$\pi'(\cdot|s) = \begin{cases} \pi^*(\cdot|s) & \text{if } Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi) \geq \Delta\epsilon_b, \\ \pi(\cdot|s) & \text{otherwise.} \end{cases} \tag{12}$$

The resultant π' switches between the expert policy and the learner policy according to whether $Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi)$ exceeds the threshold. In other words, the expert intervenes the interaction when deemed necessary according to the Q-value difference.

In the teacher-student interaction model, the intervention mode of the teacher is somewhat similar to DAgger-based active learning methods [11, 15, 24, 42]. The good performance achieved by them can be explained in the way that their intervention strategies make the expert a better teacher.

Note that Eq. (12) does not tell us how to design π' as Δ is not available. However, it exposes the mode of a good teacher: let the expert intervenes in the interaction according to the value of $Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi)$ and a threshold. Denote the threshold as p, the remaining work is to analyze the influence of p and figure out a proper p.

5 ANALYSIS

In this section, we analyze the theoretical properties of the intervention mode in both infinite and finite sample cases, and compare it with previous IL approaches.

First, we derive the sub-optimality bound for the teacher-student interaction model in the infinite sample case. The result is shown in the following theorem, whose proof can be found in Appendix ??.

Theorem 5.1. *Let π be a policy such that $\mathbb{E}_{s\sim\beta}[\ell(s, \pi, \pi')] \leq \epsilon_b$, then $J(\pi^*) - J(\pi) \leq pH + \delta\epsilon_b H^2$, where $\delta = \mathbb{E}_{s\sim d^{\pi'}}\mathbb{I}(Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi) > p)$.*

Remark 1. It seems $O(H^2)$, the term in the BC method, also appears in this sub-optimality bound. However, δ can be small if p is properly chosen and may even nullify the effect of $O(H^2)$. The definition of δ implies it decreases as p increases, while the first term pH increases as p increases. Therefore, p provides a trade-off between the two terms. Intuitively, the first term is the error induced by neglecting some erroneous actions, while the second term is caused by optimization error.

Remark 2. BC is a special case of our method. When p equals 0, δ equals 1. In this case, the expert takes over the entire training process, which is exactly the paradigm of BC. Replacing p with 0 and δ with 1, the bound becomes $\epsilon_b H^2$, which is the sub-optimality bound of BC, as shown in Appendix ??. Therefore, BC is the upper bound of sub-optimality in our framework.

Remark 3. Suppose $Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi)$ follows a distribution P, then p equals the δ quantile of P. If P is concentrated, in other words, P has strong tail decay, then a little increase in p leads to a large drop of δ, and the error bound can be improved to a great extent.

When P belongs to the Sub-Exponential distribution class, which includes many common distributions, e.g., Gaussian distribution, exponential distribution and Bernoulli distribution, we have

Corollary 5.2. *If distribution P belongs to $O(\epsilon_b)$-Sub-Exponential distribution class with expectation $O(\epsilon_b)$, let $p = \Omega(\epsilon_b \log H)$, then $J(\pi^*) - J(\pi) = \tilde{O}(\epsilon_b H)$, where $\tilde{O}$ omits the constant and* log *term.*

The proof is given in Appendix ??. For brevity, we use D_Q to denote $Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi)$ for the remaining of this paper. This corollary implies our method can avoid compounding error under a mild assumption on the distribution of D_Q. In Sec. 7, we show that the distribution P in actual tasks satisfies this assumption.

We then derive the sub-optimality bound in the finite sample case. Let $\{\hat{\pi}_i\}_{i=1}^N$ be the sequence of policies generated by our method in N iterations with a fixed p, and $\delta_i = \mathbb{E}_{s\sim d^{\hat{\pi}'_i}}\mathbb{I}(Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \hat{\pi}_i) > p)$, then we obtain the following theorem.

Theorem 5.3. *Let $\hat{\pi} = \frac{1}{N}\sum_i \hat{\pi}_i$, then $J(\pi^*) - \mathbb{E}[J(\hat{\pi})] \lesssim pH + \delta\frac{|\mathcal{S}|H^2}{N}$, where $\delta = \frac{1}{N}\sum_i \delta_i$ and $\lesssim$ omits the constant and the log term.*

If the condition of Corollary 5.2 is satisfied for all N iterations, the bound can be improved as $J(\pi^*) - \mathbb{E}[J(\hat{\pi})] \lesssim \frac{|\mathcal{S}|H}{N}$.

The bound of sample complexity can be derived from this theorem. Let value loss be ϵ, then $N = \tilde{O}(\frac{\delta|\mathcal{S}|H^2}{\epsilon - pH})$. Under the condition of Corollary 5.2, the sample complexity is $\tilde{O}(\frac{|\mathcal{S}|H}{\epsilon})$. This shows our method can also avoid the quadratic term of H in the sample complexity. In contrast, AL and AIL methods suffer from such a term in complexity even if the compounding error in the sub-optimality bound is avoided.

6 PRACTICAL IMPLEMENTATION

In this section, we design a practical algorithm based on the analysis in Sec. 4 and 5. The key idea is to find a proper value of the threshold p and a surrogate of $Q_h^{\pi^*}$ when $Q_h^{\pi^*}$ is not available.

To facilitate our derivation, we first introduce the definition of TV divergence and KL divergence.

Definition 6.1. Let P and Q be two distributions over a sample space $\mathcal{S}$, then the TV divergence between P and Q, $D_{\mathrm{TV}}(P, Q)$, is defined as

$$D_{\mathrm{TV}}(P, Q) = \frac{1}{2}\int |P(s) - Q(s)| ds.$$

The KL divergence between P and Q, $D_{\mathrm{KL}}(P, Q)$, is defined as

$$D_{\mathrm{KL}}(P, Q) = \int P(s) \log \frac{P(s)}{Q(s)} ds.$$

6.1 The choice of p

According to Corollary 5.2, the sub-optimality bound is small when the assumption on P, i.e., $p = \Omega(\epsilon_b H)$, is satisfied. However, letting $p = \Omega(\epsilon_b H)$ is inappropriate because it cannot generalize to other distribution classes and the constant in Ω is difficult to determine.

To avoid the drawbacks of Corollary 5.2, we choose p according to Theorem 5.1. Remember that p provides a trade-off between the first term and the second term of the sub-optimality bound, i.e., pH and $\delta\epsilon_b H^2$, and the order of the error depends on the larger term. Therefore, the best order of the bound can be achieved when the two terms are equal. Based on this intuition, the relationship between p and δ should be $p = \delta\epsilon_b H$. In fact, the choice of p preserves the $\tilde{O}(\epsilon_b H)$ bound in Corollary 5.2 when the assumption on P is satisfied. Please refer to Appendix **??** for a detailed discussion.

It is natural to assume the optimization process is smooth, i.e., the intervention probability δ and policy 0-1 loss ϵ_b changes slowly throughout the optimization process. Therefore, we can calculate p using δ and ϵ_b of the last iteration as an approximation. δ and ϵ_b of the last iteration are easy to obtain because ϵ_b can be calculated directly and δ can be estimated with the intervention frequency.

For tasks with continuous action spaces, the policy 0-1 loss is exactly 1, which makes the bound in Theorem 5.1 trivial. In fact, Theorem 5.1 holds for ℓ is the TV divergence between π and π^*, and we discuss this in Appendix **??**. According to Pinsker's inequality [29], $D_{\mathrm{TV}}(P, Q) \leq \sqrt{D_{\mathrm{KL}}(P, Q)}$, i.e., KL divergence can be the upper bound of TV divergence. Thus we use KL divergence instead to avoid the complex computation of TV divergence because the condition $\mathbb{E}_{s\sim\beta}[\ell(s, \pi, \pi')] \leq \epsilon_b$ still holds when ℓ is selected as the TV divergence of policy and ϵ_b is selected as the KL divergence of policy. In this way, we only need to determine p in the first iteration. The key idea to tune the initial p is to let p approximately equals $\delta\epsilon_b H$, which can be easily calculated after a few interactions with environments.

6.2 Surrogate of Q-value difference

In many real-world applications, though the exact expert Q-values are hard to get upfront, many existing methods can acquire a Q that is close to Q^{π^*}, including learning from offline datasets [13, 18, 41], using human advice [19, 26], and computing from rules [22, 23]. However, in some cases the Q-function cannot be obtained, we hope to find a surrogate of Q^{π^*}. Note that the expert policy π^* is accessible, and we derive the relationship between Q-value difference and policy divergence as follows.

Theorem 6.2. *The Q-value difference can be bounded by the policy divergence:*

$$Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi) \leq D_{\mathrm{TV}}(\pi^*(\cdot|s), \pi(\cdot|s))(H - h).$$

This theorem shows $D_{\mathrm{TV}}(\pi^*(\cdot|s), \pi(\cdot|s))(H - h)$ is the upper bound of $Q_h^{\pi^*}(s, \pi^*) - Q_h^{\pi^*}(s, \pi)$. Using the upper bound as a surrogate is reasonable because the sub-optimality bound in Theorem 5.1 is preserved.

Similarly, in the environments with continuous action spaces, we use $\sqrt{D_{\mathrm{KL}}(\pi^*(a|s), \pi(a|s))}$ instead of $D_{\mathrm{TV}}(\pi^*(a|s), \pi(a|s))$. This is because TV divergence is difficult to calculate in continuous action spaces, and Pinsker's inequality [29] guarantees the theoretical results under this modification.

For our practical algorithm, as the threshold is adaptively tuned in the training process, we name it **A**ctive a**dap**tive ex**p**ert involve**Men**t (AdapMen). The pseudo-code of AdapMen is given in Alg. 1.

Algorithm 1 Training procedure of AdapMen

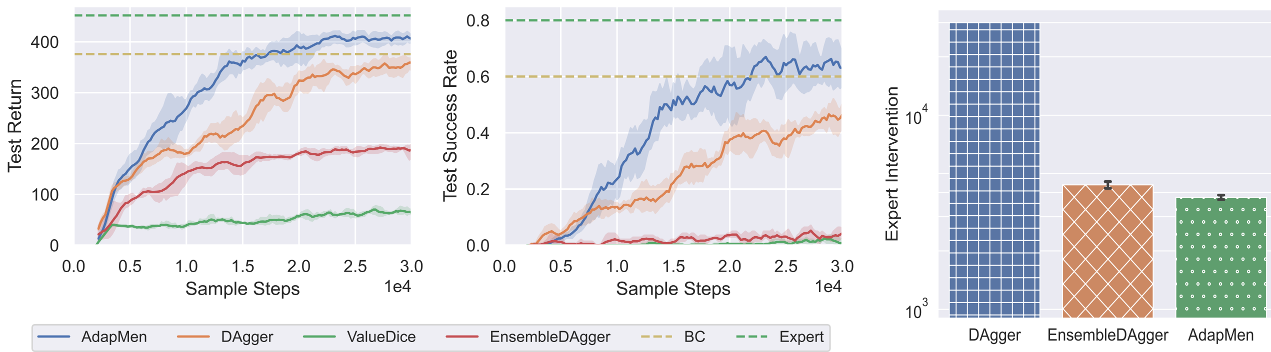
Require: An expert policy π^*; A Q-function Q corresponding to π^*; Number of sampling steps N; Learner update interval K

```
Initialize learner policy π, buffer B, p
for n = 1 to N do
    Get learner agent action a_l and expert action a_e
    Calculate the surrogate Q-value difference D_Q
    if D_Q > p then
        Take expert action a_e, add the transition to B
    else
        Take learner agent action a_l
    end if
    if n%K == 0 then
        Sample batches of transitions from B to train π
        Update p-value
    end if
end for
```

7 EXPERIMENTS

In this section, we conduct experiments to test whether AdapMen reaches the theoretical advantages of our framework.

We choose MetaDrive [20] and Atari 2600 games from ALE [3] as benchmarks. MetaDrive is a highly compositional autonomous driving benchmark that is closely related to real-world applications. The MetaDrive simulator can generate an infinite number of diverse driving scenarios from both procedural generation and real data importing. The agent observes a 259-dimensional vector which



(a) Performance (b) Total expert action usage

Figure 2: Performance in MetaDrive with policy experts

is composed of a 240-dimensional vector denoting the 2D-Lidar-like point clouds, a vector summarizing the target vehicle's state and a vector for the navigation information. The action space is a continuous 2-dimensional vector representing the acceleration and steering of the car, respectively. The goal is to follow the traffic rules and reach the target position as fast as possible. The training configuration of MetaDrive follows that in [28]. For the justice of comparison, the evaluation is performed on 20 randomly selected scenarios. Atari 2600 games are challenging visual-input RL tasks with discrete action spaces. Using conventional environment wrappers and processing techniques, the agent observes a (84×84) grayscale image and has discrete action spaces ranging from 6 valid actions to 18 valid actions depending on the game. We randomly select six common Atari games. To avoid the small stochasticity problem of the Atari simulator, we activate the "sticky action" feature to simulate actual human input and increase stochasticity.

We choose BC [2], DAgger [35], HG-DAgger [15], EnsembleDAgger [24], and ValueDICE [17] as baselines. The details of BC and DAgger have been introduced in Sec. 1. HG-DAgger and EnsembleDAgger are representative methods of active imitation learning methods. HG-DAgger allows interactive imitation learning from human experts in real-world systems by letting a human expert take over control when deemed necessary, and EnsembleDAgger uses both action variance from an ensemble of policies and action discrepancies between learner and expert as the criterion to decide whether the expert should take over control. ValueDICE is the SoTA of AIL methods, which trains the learner agent via robust divergence minimization in an off-policy manner. Hyper-parameters of the implementations of baselines are listed in Appendix **??**.

We first test the performance of AdapMen and baselines in the two benchmarks with expert in the form of trained policies, namely policy experts. Then, we dive into AdapMen and demonstrate some key properties of our algorithm to answer the following questions:

- How is the intervention threshold automatically adjusted during the training process?
- Can the distribution of D_Q satisfy the assumption in Corollary 5.2 in most cases?
- Is policy divergence a good surrogate of D_Q?

Finally, we simulate real-world scenarios by letting a human be the expert and control the vehicle in the MetaDrive benchmark.

7.1 Performance with policy experts

7.1.1 Performance in MetaDrive. The expert policy of MetaDrive is trained by Soft Actor-Critic (SAC) [8]. For AdapMen, we take one of the trained Q-networks as Q^{π^*} and calculate D_Q based on it. To demonstrate the robustness towards inaccurate Q^{π^*} when the ground truth value is not available, we also perform experiments on the estimated value function in Appendix **??**.

The performance in the MetaDrive benchmark is plotted in Fig. 2. The horizontal axis represents the total number of steps sampled in the environment. The vertical axis of Fig. 2(a) and Fig. 2(b) are policy return, success rate, and number of expert interventions, respectively. HG-DAgger is omitted in experiments for the sake of fairness because the expert of the algorithm should be human.

For the MetaDrive benchmark, AdapMen achieves the best performance in terms of both cumulative return and success rate. ValueDICE achieves the worst performance probably because of its highest sample complexity and sensitivity to hyper-parameters thus we fail to find a working configuration. Notwithstanding the low expert intervention counts of EnsembleDAgger, the performance of EnsembleDAgger severely degrades. The μ-recoverability property of DAgger is hard to satisfy in risk-sensitive environments so that DAgger shows no advantage than BC. BC achieves the best performance among all the baseline algorithms. This is because the policy expert has little stochasticity and the dimension of input is small.

The total number of expert data usage is shown in Fig. 2(b). Here the expert data usage is defined as the number of expert state-action pairs added to the buffer for training the learner. This quantity of BC, ValueDICE is the same as DAgger and we omit them in the figure. DAgger always adds the expert state-action pair to the buffer, thus having the biggest expert data usage. Compared with EnsembleDAgger, by generating the best buffer distribution for teaching, AdapMen requires fewer expert interventions while achieving a better test performance.

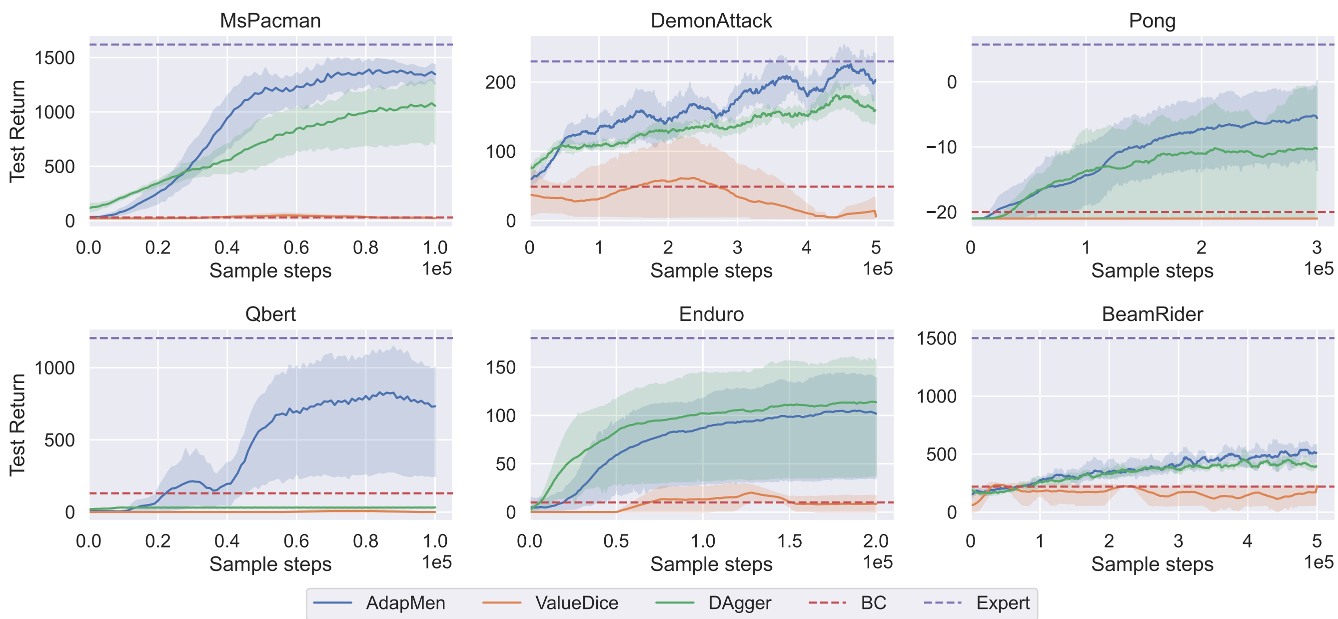


Figure 3: Performance in six Atari games with policy experts

To further verify our theory, we draw the trend of p-value and actual intervention probability throughout the training process in Fig. 4, where the left vertical axis represents the value of p,while the right vertical axis represents the intervention probability. The probability is calculated every 200 sample steps in the environment. Theorem 5.1 implies the sub-optimality is negatively related to p and δ. This is verified by the decreasing trend of p and δ in the training process, coinciding with the increasing policy return in Fig. 2(a). Meanwhile, the sharply changing p also demonstrates the importance of adaptively changing intervention criterion. Intuitively, as the learner agent gets better at driving the car, the teacher should increase the difficulty of the teaching policy. A lower p-value indicates more difficult learning content.

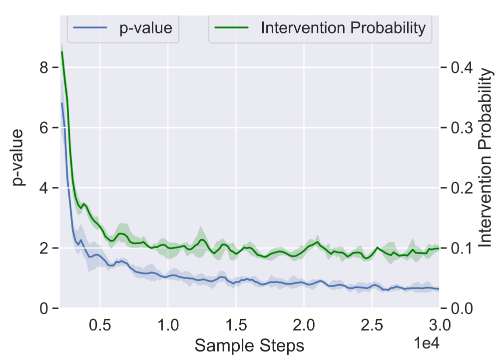


Figure 4: p-value and intervention probability of AdapMen on MetaDrive

7.1.2 Performance in Atari games. The expert policies of Atari games are trained by Deep Q Learning [25]. Note that we activate the "sticky action" features to increase the stochasticity of the tasks. Since Ensemble-DAgger requires a continuous action space, we omit it for comparison in the Atari 2600 games. The performance curves in the Atari 2600 games are plotted in Fig. 3.

AdapMen outperforms baselines in 5 out of 6 Atari games. These tasks are more challenging, which can be inferred from the performance of baselines. In Qbert, all algorithms fail to learn from the expert except for AdapMen. In all the tasks, ValueDICE performs equally poorly as in MetaDrive. BC, which has near-optimal performance in MetaDrive, also collapses in most of the six Atari games. This shows that BC fails in higher-dimensional environments. DAgger performs better than other baselines, this is probably because the μ-recoverability assumption can still be satisfied in most states in Atari games.

7.2 Performance of AdapMen criterion based on policy divergence

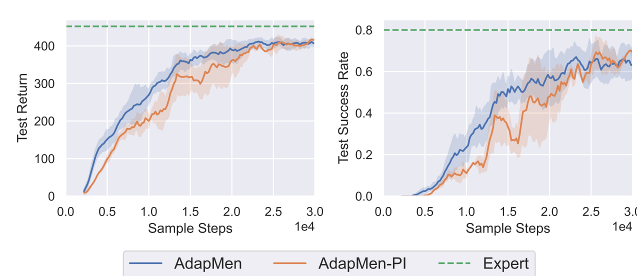
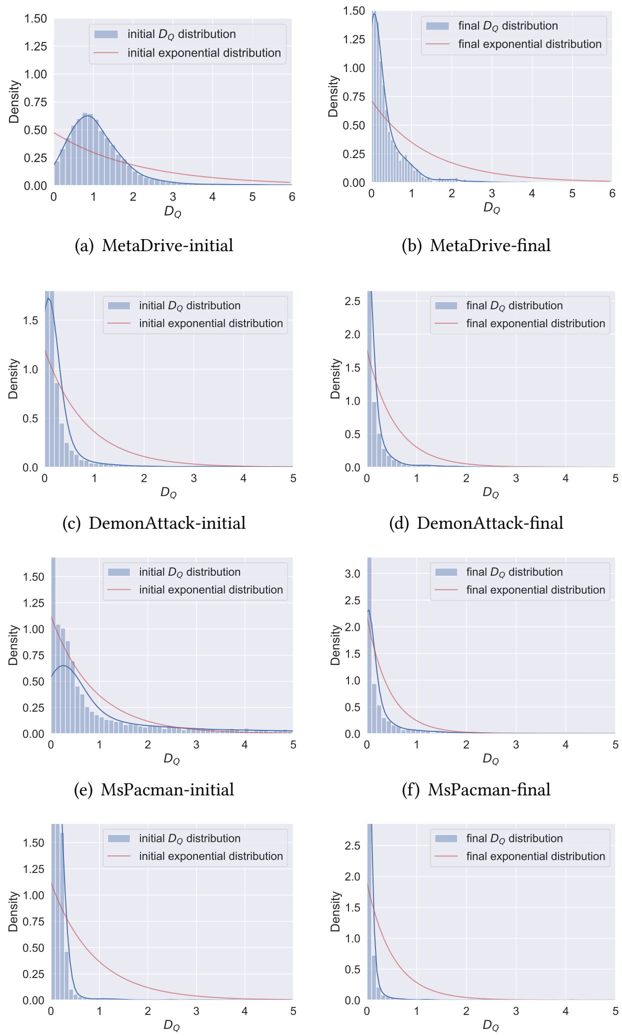


Figure 5: Performance in MetaDrive with different criteria of AdapMen

As mentioned in Sec. 6, when Q^{π^*} is not available, we use policy divergence as a surrogate of D_Q. To validate the correctness of this surrogate, we test it on MetaDrive, and plot its performance in Fig. 5. AdapMen is the original algorithm, while AdapMen-PI uses the policy divergence instead of D_Q. The result shows AdapMen-PI has comparable performance with AdapMen. This experiment validates our theory and demonstrates that policy divergence is also a proper criterion.

7.3 Analysis of D_Q distribution

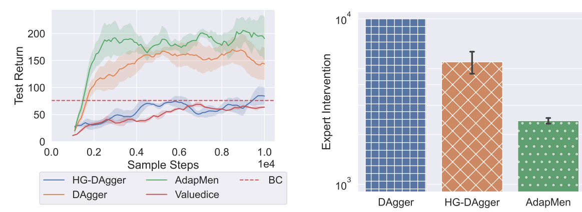


(a) MetaDrive-initial (b) MetaDrive-final (c) DemonAttack-initial (d) DemonAttack-final (e) MsPacman-initial (f) MsPacman-final (g) BeamRider-initial (h) BeamRider-final

Figure 6: D_Q distributions in MetaDrive and Atari games. The blue lines show the distributions of D_Q estimated by kernel density estimation.

In Corollary 5.2, we assume the distributions of D_Q, i.e., P in Sec. 5, belongs to $O(\epsilon_b)$-Sub-Exponential distribution class with expectation $O(\epsilon_b)$. The assumption is satisfied if the tail of P is bounded by an exponential distribution with parameter ϵ_b. To verify this assumption, we plot P of MetaDrive and the six Atari games and compare their tails with exponential distribution. The partial results are shown in Fig. 6 because of space limitation. The rest of the results are in Appendix **??**. The distributions at the beginning and at the end of the training are on the left-hand side and right-hand side, respectively. All the tails of P are bounded by the exponential distribution, which implies the assumption is satisfied in nearly all tested tasks. This bridges the gap between the theoretical analysis and practical applicability of AdapMen.

7.4 Performance in MetaDrive with a human expert



(a) Test Return (b) Expert action usage

Figure 7: Performance in MetaDrive with a human expert

In real-world tasks, humans are important sources of expert information, especially in autonomous driving tasks. To mimic real-world tasks, we substitute the SAC expert in MetaDrive with a human. The experimental results are shown in Fig. 7. The random and sometimes irrational behaviors of human experts raise a huge challenge for imitation learning algorithms, and general degradation of performance happens for all methods. BC has a 75% performance degradation. In contrast, AdapMen has a relatively small performance degradation and achieves the best final performance. The performance of HG-DAgger is surprising. Although our human expert has tries his best to correct the behavior of the learner agent, HG-DAgger is only slightly better than BC. HG-DAgger even uses more expert actions to train the learner policy than AdapMen. This shows the teaching strategy of humans are unreliable and an objective criterion is important.

8 CONCLUSION

In this paper, we formulate the IL process as a teacher-student interaction framework. The proposed framework shows expert should involve in the interaction of the agent with the environment according to a certain criterion. We theoretically verify the effectiveness of this framework, and derive a better error bound and sample complexity under a mild condition, which we experimentally demonstrate common in many benchmarks. Based on the teacher-student interaction framework, we propose a practical method AdapMen, where the intervention criterion is tuned automatically in the training process, which frees the hyper-parameter tuning budget of other active imitation learning methods. Experimental results demonstrate that AdapMen achieves a better performance than other IL methods.

ACKNOWLEDGMENTS

This work is supported by National Key Research and Development Program of China (2020AAA0107200), the National Science Foundation of China (61921006, 62276126) and the Major Key Project of PCL (PCL2021A12).

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of the 21st International Conference of Machine Learning*. 1.
[2] Michael Bain and Claude Sammut. 1995. A Framework for Behavioural Cloning. In *Machine Intelligence 15*. 103–129.
[3] Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research* 47 (2013), 253–279.
[4] Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. 2019. Generative Adversarial User Model for Reinforcement Learning Based Recommendation System. In *Proceedings of the 36th International Conference on Machine Learning*. 1052–1061.
[5] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. 2019. Exploring the Limitations of Behavior Cloning for Autonomous Driving. In *Proceedings of the International Conference on Computer Vision*. 9328–9337.
[6] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. 2021. IQ-Learn: Inverse Soft-Q Learning for Imitation. In *Advances in Neural Information Processing Systems 34*. 4028–4039.
[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. In *Advances in Neural Information Processing Systems 27*.
[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*. 1856–1865.
[9] Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems 29*. 4565–4573.
[10] Jonathan Ho, Jayesh K. Gupta, and Stefano Ermon. 2016. Model-Free Imitation Learning with Policy Optimization. In *Proceedings of the 33rd International Conference on Machine Learning*. 2760–2769.
[11] Ryan Hoque, Ashwin Balakrishna, Ellen R. Novoseller, Albert Wilcox, Daniel S. Brown, and Ken Goldberg. 2021. ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning. In *Proceedings of the 5th Conference on Robot Learning*. 598–608.
[12] Ryan Hoque, Ashwin Balakrishna, Carl Putterman, Michael Luo, Daniel S. Brown, Daniel Seita, Brijen Thananjeyan, Ellen R. Novoseller, and Ken Goldberg. 2021. LazyDAgger: Reducing Context Switching in Interactive Imitation Learning. In *Proceedings of the 17th International Conference on Automation Science and Engineering*. 502–509.
[13] Xue-Kun Jin, Xu-Hui Liu, Shengyi Jiang, and Yang Yu. 2022. Hybrid Value Estimation for Off-policy Evaluation and Offline Reinforcement Learning. *CoRR* abs/2206.02000 (2022).
[14] Sham M. Kakade and John Langford. 2002. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the 19th International Conference on Machine Learning*. 267–274.
[15] Michael Kelly, Chelsea Sidrane, Katherine Rose Driggs-Campbell, and Mykel J. Kochenderfer. 2018. HG-DAgger: Interactive Imitation Learning with Human Experts. In *Proceedings of the 35th International Conference on Robotics and Automation*. 8077–8083.
[16] Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. 2019. Discriminator-Actor-Critic: Addressing Sample Inefficiency and Reward Bias in Adversarial Imitation Learning. In *Proceedings of the 7th International Conference on Learning Representations*.
[17] Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. 2019. Imitation Learning via Off-Policy Distribution Matching. In *Proceedings of the 8th International Conference on Learning Representations*.
[18] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems 33*.
[19] Kimin Lee, Laura M. Smith, and Pieter Abbeel. 2021. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. *CoRR* abs/2106.05091 (2021).
[20] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. 2022. Metadrive: Composing Diverse Driving Scenarios for Generalizable Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2022), 3461–3475.
[21] Quanyi Li, Zhenghao Peng, and Bolei Zhou. 2022. Efficient Learning of Safe Driving Policy via Human-AI Copilot Optimization. In *Proceedings of the 10th International Conference on Learning Representations*.
[22] Richard Maclin, Jude Shavlik, Trevor Walker, and Lisa Torrey. 2005. Knowledge-based support-vector regression for reinforcement learning. *Reasoning, Representation, and Learning in Computer Games* (2005).
[23] Olvi L. Mangasarian, Jude W. Shavlik, and Edward W. Wild. 2004. Knowledge-Based Kernel Approximation. *Journal of Machine Learning Research* 5 (2004), 1127–1141.
[24] Kunal Menda, Katherine Rose Driggs-Campbell, and Mykel J. Kochenderfer. 2019. EnsembleDAgger: A Bayesian Approach to Safe Imitation Learning. In *Proceedings of International Conference on Intelligent Robots and Systems*. 5041–5048.
[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level Control through Deep Reinforcement Learning. *Nature* 518, 7540 (2015), 529–533.
[26] Anis Najar, Olivier Sigaud, and Mohamed Chetouani. 2016. Training a robot with evaluative feedback and unlabeled guidance signals. In *Proceedings of the 25th International Symposium on Robot and Human Interactive Communication*. 261–266.
[27] Andrew Y. Ng and Stuart Russell. 2000. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the 17th International Conference on Machine Learning*. 663–670.
[28] Zhenghao Peng, Quanyi Li, Chunxiao Liu, and Bolei Zhou. 2021. Safe Driving via Expert Guided Policy Optimization. In *Proceedings of 5th Conference on Robot Learning*. 1554–1563.
[29] David Pollard. 2000. *Asymptopia: An Exposition of Statistical Asymptotic Theory*.
[30] Hong Qian, Xu-Hui Liu, Chen-Xi Su, Aimin Zhou, and Yang Yu. 2022. The Teaching Dimension of Regularized Kernel Learners. In *Proceedings of the 39th International Conference on Machine Learning*. 17984–18002.
[31] Nived Rajaraman, Yanjun Han, Lin Yang, Jingbo Liu, Jiantao Jiao, and Kannan Ramchandran. 2021. On the Value of Interaction and Function Approximation in Imitation Learning. In *Advances in Neural Information Processing Systems 34*. 1325–1336.
[32] Nived Rajaraman, Lin F. Yang, Jiantao Jiao, and Kannan Ramchandran. 2020. Toward the Fundamental Limits of Imitation Learning. In *Advances Neural Information Processing Systems 33*.
[33] Stéphane Ross and Drew Bagnell. 2010. Efficient Reductions for Imitation Learning. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. 661–668.
[34] Stéphane Ross and J. Andrew Bagnell. 2014. Reinforcement and Imitation Learning via Interactive No-Regret Learning. *CoRR* abs/1406.5979 (2014).
[35] Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, Vol. 15. 627–635.
[36] Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. 2019. Virtual-Taobao: Virtualizing Real-world Online Retail Environment for Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence 33*. 4902–4909.
[37] Jonathan C. Spencer, Sanjiban Choudhury, Matt Barnes, Matthew Schmittle, Mung Chiang, Peter J. Ramadge, and Siddhartha S. Srinivasa. 2020. Learning from Interventions: Human-robot Interaction as Both Explicit and Implicit Feedback. In *Robotics: Science and Systems XVI*.
[38] Wen Sun, Arun Venkatraman, Geoffrey J. Gordon, Byron Boots, and J. Andrew Bagnell. 2017. Deeply AggreVaTeD: Differentiable Imitation Learning for Sequential Prediction. In *Proceedings of the 34th International Conference on Machine Learning*. 3309–3318.
[39] Tian Xu, Ziniu Li, and Yang Yu. 2020. Error Bounds of Imitating Policies and Environments. In *Advances in Neural Information Processing Systems 33*.
[40] Tian Xu, Ziniu Li, and Yang Yu. 2021. Nearly Minimax Optimal Adversarial Imitation Learning with Known and Unknown Transitions. *CoRR* abs/2106.10424 (2021).
[41] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. 2021. COMBO: Conservative Offline Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems 34*. 28954–28967.
[42] Jiakai Zhang and Kyunghyun Cho. 2017. Query-efficient Imitation Learning for End-to-End Autonomous Driving. In *Proceedings of the AAAI Conference on Artificial Intelligence 31*.
[43] Yi-Feng Zhang, Fan-Ming Luo, and Yang Yu. 2022. Improve Generated Adversarial Imitation Learning with Reward Variance regularization. *Machine Learning* 111, 3 (2022), 977–995.
[44] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N. Rafferty. 2018. An Overview of Machine Teaching. *CoRR* abs/1801.05927 (2018).