How to Leverage Diverse Demonstrations in Offline Imitation Learning

Sheng Yue [1] **Jiani Liu** [1] **Xingyuan Hua** [1] **Ju Ren** [1 2] **Sen Lin** [3] **Junshan Zhang** [4] **Yaoxue Zhang** [1 2]

Abstract

Offline Imitation Learning (IL) with imperfect demonstrations has garnered increasing attention owing to the scarcity of expert data in many real-world domains. A fundamental problem in this scenario is *how to extract positive behaviors from noisy data*. In general, current approaches to the problem select data building on state-action similarity to given expert demonstrations, neglecting precious information in (potentially abundant) *diverse* state-actions that deviate from expert ones. In this paper, we introduce a simple yet effective data selection method that identifies positive behaviors based on their *resultant states* – a more informative criterion enabling explicit utilization of dynamics information and effective extraction of both expert and beneficial diverse behaviors. Further, we devise a lightweight behavior cloning algorithm capable of leveraging the expert and selected data correctly. In the experiments, we evaluate our method on a suite of complex and high-dimensional offline IL benchmarks, including continuous-control and vision-based tasks. The results demonstrate that our method achieves state-of-the-art performance, outperforming existing methods on **20/21** benchmarks, typically by **2-5x**, while maintaining a comparable runtime to Behavior Cloning (`BC`).

1. Introduction

Offline Imitation Learning (IL) is the study of learning from demonstrations with no reinforcement signals or interaction with the environment. It has been deemed as a promising solution for safety-sensitive domains like healthcare and autonomous driving, where manually formulating a reward function is challenging but historical human demonstrations are readily available (Bojarski et al., 2016). Conventional offline IL methods, such as Behavior Cloning (`BC`) (Pomerleau, 1988), often necessitate an expert dataset with sufficient coverage over the state-action space to combat error compounding (Rajaraman et al., 2020), which is prohibitively expensive for many real-world applications. Instead, a more realistic scenario might allow for a limited expert dataset, coupled with substantial imperfect demonstrations sampled from unknown policies (Wu et al., 2019; Xu et al., 2022; Li et al., 2023). For example, autonomous vehicle companies may possess modest high-quality data from experienced drivers but can amass a wealth of mixed-quality data from ordinary drivers. Effective utilization of the imperfect demonstrations would significantly enhance the robustness and generalization of offline IL.

A fundamental question raised in this scenario is: *how can we extract good behaviors from noisy data?* To address this question, several prior studies have attempted to explore and imitate the imperfect behaviors that resemble expert ones (Sasaki & Yamashina, 2021; Xu et al., 2022; Li et al., 2023). Nevertheless, due to the scarcity of expert data, such approaches are ill-equipped to harness valuable information in (potentially abundant) *diverse* behaviors that deviate from limited expert demonstrations (see Section 3 for details). Of course, a natural solution to incorporate these behaviors is inferring a reward function and labeling all imperfect data, subsequently engaging in an offline Reinforcement Learning (RL) process (Zolna et al., 2020; Chang et al., 2021; Yue et al., 2023; Zeng et al., 2023; Cideron et al., 2023). Unfortunately, it is highly challenging to define and learn meaningful reward functions without environmental interaction. As a consequence, current offline reward learning methods typically rely on complex adversarial optimization using a learned world model. They easily suffer from hyperparameter sensitivity, learning instability, and limited scalability in practical and high-dimensional environments.

In this paper, we introduce a simpler data selection principle to fully exploit positive diverse behaviors in imperfect demonstrations without indirect reward learning procedures. Specifically, instead of examining a behavior's similarity to expert demonstrations in and of itself, we assess its value based on whether its *resultant states*, to which environment

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, China [2] Zhongguancun Laboratory, Beijing, China [3] Department of Computer Science, University of Houston, Texas, US [4] Department of Electrical and Computer Engineering, University of California, Davis, US. Correspondence to: Ju Ren <renju@tsinghua.edu.cn>.

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

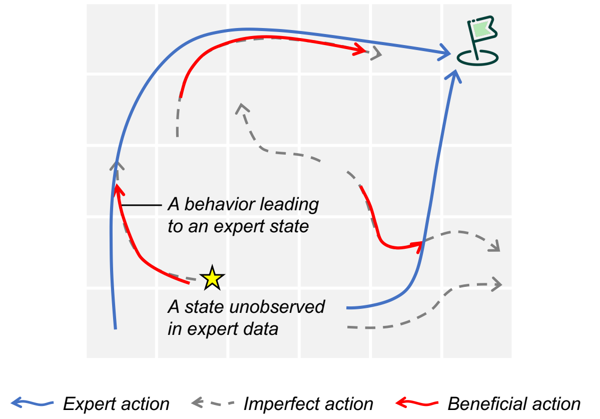


Figure 1. A cartoon illustration of potential beneficial behaviors in a navigation task, with the goal of reaching the target state (marked by the flag) from arbitrary initial states. With no other prior information, in a state out of the expert observations, a reasonable and often safe choice is to get back to given expert states.

transitions after performing that behavior, fall within the expert data manifold. In other words, we properly select the state-actions that can lead to expert states, even if they bear no resemblance to expert demonstrations. As depicted in Fig. 1 and supported by the theoretical results in Section 4.1, the underlying rationale is that when the agent encounters a state unobserved in expert demonstrations, opting to return to the expert states is more sensible than taking a random action. Otherwise, it may persist in making mistakes and remain out-of-expert-distribution for subsequent time steps. Of note, the resultant state is a more informative criterion than the state-action similarity, as it explicitly utilizes the dynamics information, enabling the identification of both expert and beneficial diverse state-actions in noisy data.

Drawing upon this principle, we first train a *state-only* discriminator to distinguish expert and non-expert states in imperfect demonstrations. Leveraging the identified expert states, we appropriately extract their *causal state-actions* and build a complementary training dataset. In light of the suboptimality of the complementary data, we further devise a lightweight weighted behavior cloning algorithm to mitigate the potential interference among behaviors. We term our method *offline Imitation Learning with Imperfect Demonstrations* (ILID) and evaluate it on a suite of offline IL benchmarks, including 14 continuous-control tasks and 7 vision-based tasks. Our method achieves state-of-the-art performance, consistently surpassing existing methods by **2-5x** while maintaining a comparable runtime to BC. Our main contributions are summarized as follows:

- We introduce a simple yet effective method that can explicitly exploit the dynamics information and extract beneficial behaviors from imperfect demonstrations;
- We devise a lightweight weighted behavior cloning algorithm capable of correctly learning from the extracted behaviors, which can be easily implemented on top of BC;
- We conduct extensive experiments that corroborate the superiority of our method over state-of-the-art baselines in terms of performance and computational cost.

2. Related Work

Offline IL deals with training an agent to mimic the actions of a demonstrator in an entirely offline fashion. The simplest approach to offline IL is BC (Pomerleau, 1988) that directly mimics the behavior using supervised learning, but it is prone to covariate shift and inevitably suffers from error compounding, i.e., there is no way for the policy to learn how to recover if it deviates from the expert behavior to a state not seen in the expert demonstrations (Rajaraman et al., 2020). Considerable research has been devoted to developing new offline IL methods to remedy this problem (Klein et al., 2012a;b; Piot et al., 2014; Herman et al., 2016; Kostrikov et al., 2020; Jarrett et al., 2020; Swamy et al., 2021; Chan & van der Schaar, 2021; Garg et al., 2021; Florence et al., 2022). However, since these methods imitate all given demonstrations, they typically require a large amount of clean expert data, which is expensive for many real-world tasks.

Recently, there has been growing interest in exploring how to effectively leverage imperfect data in offline IL (Sasaki & Yamashina, 2021; Kim et al., 2022; Xu et al., 2022; Yu et al., 2022; Li et al., 2023). Sasaki & Yamashina (2021) analyze why the imitation policy trained by BC deteriorates its performance when using noisy demonstrations. They reuse an ensemble of policies learned from the previous iteration as the weight of the original BC objective to extract the expert behaviors. Nevertheless, this requires that expert data occupy the majority proportion of the offline dataset; otherwise, the policy will be misguided to imitate the suboptimal data. Kim et al. (2022) retrofit the BC objective with an additional KL-divergence term to regularize the learned policy to stay close to the behavior policy. Albeit with enhanced offline data support, it may fail to achieve satisfactory performance when the imperfect data is highly suboptimal. Xu et al. (2022) cope with this issue by introducing an additional discriminator, the outputs of which serve as the weights of the original BC loss, to imitate demonstrations selectively. Analogously, Li et al. (2023) weight the BC objective by the density ratio of empirical expert data and union offline data, implicitly extracting the imperfect behaviors resembling expert ones. Unfortunately, the criterion of state-action similarity neglects the dynamics information and does not suffice to leverage the diverse behaviors in imperfect demonstrations. In offline RL, Yu et al. (2022) propose to utilize unlabeled data by applying zero rewards, but this method necessitates massive labeled offline data. In contrast, this paper focuses on the setting with no access to reward signals.

Offline Inverse Reinforcement Learning (IRL) explicitly learns a reward function from offline datasets, aiming to comprehend and generalize the underlying intentions behind expert actions (Lee et al., 2019). Zolna et al. (2020) propose ORIL that constructs a reward function that discriminates expert and exploratory trajectories, followed by an offline RL progress. Chan & van der Schaar (2021) use a variational method to jointly learn an approximate posterior distribution over the reward and policy. Garg et al. (2021) propose to learn a soft Q-function that implicitly represents both the reward function and policy. Watson et al. (2024) develop CSIL that exploits a BC policy to define an estimate of a shaped reward function that can then be used to finetune the policy using online interactions. However, the heteroscedastic parametric reward functions have undefined values beyond the offline data manifold and easily collapse to the reward limits due to the tanh transformation and network extrapolation. The reward extrapolation errors may cause the learned reward functions to incorrectly explain the task and misguide the agent in unseen environments (Yue et al., 2023; 2024). To tackle the issue, Chang et al. (2021) introduce a model-based offline IRL algorithm that uses a model inaccuracy estimate to penalize the learned reward function on out-of-distribution state-actions. Yue et al. (2023) propose to compute a conservative element-wise weight function that implicitly penalizes out-of-distribution behaviors. Zeng et al. (2022) propose MLIRL that can recover the reward function, whose corresponding optimal policy maximizes the likelihood of observed expert demonstrations under a learned conservative world model. However, the model-based approaches struggle to scale in high-dimensional environments, and their min-max optimization usually renders training unstable and inefficient.

3. Background and Challenge

In this section, we first provide the necessary preliminaries and then elaborate on the challenges of our problem.

Episodic Markov decision process. Episodic MDP can be specified by $M \doteq \langle \mathcal{S}, \mathcal{A}, T, R, H, \mu \rangle$, with state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $T : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, reward function $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$, horizon H, and initial state distribution $\mu : \mathcal{S} \to \mathcal{P}(\mathcal{S})$, where $\mathcal{P}(\mathcal{S})$ represents the set of distributions over $\mathcal{S}$. A stationary stochastic policy maps states to distributions over actions, $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$. The value function of π is defined as the expected cumulative reward, $V^\pi \doteq \mathbb{E}_\pi[\sum_{h=1}^{H} R(s_h, a_h)]$, with the expectation taken w.r.t. trajectories generated by rolling out π with M. The average state visitation and state-action visitation of π are denoted as $\rho^\pi(s) \doteq \frac{1}{H}\sum_{h=1}^{H} \Pr(s_h = s \mid \pi)$ and $\rho^\pi(s, a) \doteq \rho^\pi(s)\pi(a|s)$ respectively, where $\Pr(s_h = s \mid \pi)$ represents the probability of visiting state s at step h. The objective of RL can be expressed as $\max_\pi V^\pi$.

Offline imitation learning. Offline IL is the setting where the algorithm is neither allowed to interact with the environment nor provided ground-truth rewards. Rather, it has access to an expert dataset and a mix-quality imperfect dataset, collected from unknown expert policy π_e and (potentially highly suboptimal) behavior policy π_b, respectively. We represent the expert and imperfect datasets as $\mathcal{D}_e \doteq \{\tau_i\}_{i=1}^{n_e}$ and $\mathcal{D}_b \doteq \{\tau_i\}_{i=1}^{n_b}$, where $\tau_i \doteq (s_{i,1}, a_{i,1}, \ldots, s_{i,H}, a_{i,H})$ denotes a trajectory.

Behavior cloning. BC is a classical offline IL algorithm, which seeks to learn an imitation policy using supervised learning (Pomerleau, 1988). The standard objective of BC is to maximize the log-likelihood over expert demonstrations:

$$\max_\pi \mathbb{E}_{(s,a)\sim\mathcal{D}_e}\big[\log(\pi(a|s))\big]. \tag{1}$$

Recent studies consider a more generalized objective (Xu et al., 2022; Li et al., 2023), incorporating additional yet imperfect demonstrations:

$$\min_\pi \mathbb{E}_{(s,a)\sim\mathcal{D}_u}\left[f(s,a)\log\pi(a|s)\right] \tag{2}$$

where $\mathcal{D}_u \doteq \mathcal{D}_e \cup \mathcal{D}_b$ represents the union offline dataset comprised of both expert and imperfect demonstrations, and $f : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is a weighting function aiming to discard low-quality behaviors and only imitate the beneficial ones. For example, DWBC (Xu et al., 2022) pick f as

$$f(s,a) = \begin{cases} \alpha - \frac{\eta}{d_\pi(s,a)(1-d_\pi(s,a))}, & (s,a) \in \mathcal{D}_e \\ \frac{1}{1-d_\pi(s,a)}, & (s,a) \in \mathcal{D}_b \end{cases} \tag{3}$$

where $\alpha, \eta > 0$ are hyperparameters. $d_\pi(s, a)$ is the output of a discriminator that is jointly trained with π to distinguish the expert and diverse state-actions:

$$\max_{d_\pi} \mathbb{E}_{\mathcal{D}_e}\left[\log d_\pi(s,a)\right] + \frac{1}{\eta}\mathbb{E}_{\mathcal{D}_b}\left[\log(1-d_\pi(s,a))\right] - \mathbb{E}_{\mathcal{D}_e}\left[\log(1-d_\pi(s,a))\right]. \tag{4}$$

Eqs. (3) and (4) indicate that DWBC assigns high values to $(s, a) \in \mathcal{D}_e$ and low values to $(s, a) \in \mathcal{D}_b \backslash \mathcal{D}_e$. In addition, ISWBC (Li et al., 2023) let f denote the importance weight $f(s, a) = \tilde{\rho}^e(s, a)/\tilde{\rho}^u(s, a)$ where $\tilde{\rho}^e$ and $\tilde{\rho}^u$ are the empirical distributions of $\mathcal{D}_e$ and $\mathcal{D}_u$, respectively. In the same spirit as Xu et al. (2022), the weight assigns positive values to $(s, a) \in \mathcal{D}_e$ and close-to-zero values to $(s, a) \in \mathcal{D}_b \backslash \mathcal{D}_e$.

Challenge. The above-mentioned weighting functions can extract $(s, a) \in \mathcal{D}_e$ from $\mathcal{D}_b$, (implicitly) filtering out the state-actions in $\mathcal{D}_b \backslash \mathcal{D}_e$. However, the limited *state* coverage of expert data would render these learned policies still brittle to covariate shift due to their inability to get back on track if encountering a state not observed in the expert demonstrations (see Fig. 2 for an illustrative example). Moreover, considering that offline (forward) RL can learn effective policies from highly diverse behavioral data (Fu et al., 2020; Rashidinejad et al., 2021), these methods neglect potentially substantial *beneficial* behaviors in $\mathcal{D}_b \backslash \mathcal{D}_e$ that deviate the expert demonstrations. Thus, there is a clear need for new offline IL methods capable of capitalizing on the diverse behaviors of imperfect demonstrations.

4. Offline Imitation Learning with Imperfect Demonstrations

This section elaborates on our proposed method. We begin by presenting a hypothesis on behavior selection and providing it with theoretical justification. Building on the hypothesis and theoretical insights, we then delineate our data selection and policy learning methods.

4.1. Selection of Imperfect Behaviors

In contrast to the existing works that select data building on state-action resemblance to given expert demonstrations, we propose to access an imperfect behavior by its *resultant states*, to which the environment transitions after performing the behavior. Formally, we present the following hypothesis.

Hypothesis 4.1. With no other prior knowledge, if a state s lies *beyond* given expert data ($s \notin \mathcal{D}_e$), then, in s, taking the action that can transition to a known expert state is more beneficial than selecting actions at random.

To support this hypothesis, we provide the following theoretical results under deterministic dynamics.[1] Represent $\mathcal{D}$ as a demonstration dataset, $\mathcal{S}(\mathcal{D})$ as the set of states in $\mathcal{D}$, and $\mathcal{S}_h(\mathcal{D})$ as the set of h-step visited states in $\mathcal{D}$. Suppose that π_e is optimal and deterministic (Sutton & Barto, 2018), and there exists a supplementary dataset consisting of transitions *from initial states to given expert states*, $\mathcal{D}_s \doteq \{(s_i, a_i, s'_i) \mid s_i \sim \mu, T(s_i, a_i) = s'_i, s'_i \in \mathcal{S}_1(\mathcal{D}_e), i = 1, \dots, n_s\}$. According to Hypothesis 4.1, we consider the policy $\tilde{\pi}$ that takes the logging actions in $\mathcal{D}_s$ at states $\mathcal{S}_1(\mathcal{D}_s) \backslash \mathcal{S}_1(\mathcal{D}_e)$ and takes the expert actions in $\mathcal{D}_e$ at expert states $\mathcal{S}(\mathcal{D}_e)$:

$$\tilde{\pi}(a|s) \doteq \begin{cases} \frac{n((s,a)\in\mathcal{D}_s)}{n(s\in\mathcal{S}_1(\mathcal{D}_s))}, & \text{if } s \in \mathcal{S}_1(\mathcal{D}_s)\backslash\mathcal{S}_1(\mathcal{D}_e) \\ \frac{n((s,a)\in\mathcal{D}_e)}{n(s\in\mathcal{S}(\mathcal{D}_e))}, & \text{if } s \in \mathcal{S}(\mathcal{D}_e) \\ \frac{1}{|\mathcal{A}|}, & \text{otherwise} \end{cases} \tag{5}$$

where $n(s \in \mathcal{D}) = \sum_{s' \in \mathcal{D}} \mathbb{1}(s' = s)$ denotes the number of element s in set $\mathcal{D}$, and $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$.[2] Denote $\delta \doteq \max\{V^{\pi_e}(s_1) - V^{\pi_e}(s_2) \mid \mu(s_1), \mu(s_2) > 0\}$ as the maximum return difference among expert trajectories, with $V^{\pi}(s) \doteq \mathbb{E}_{\pi}[\sum_{h=1}^{H} R(s_h, a_h) \mid s_1 = s]$. Next, we characterize the suboptimality of $\tilde{\pi}$ in Theorem 4.2.

Theorem 4.2. *For any finite and episodic MDP with deterministic transition dynamics, the following fact holds:*

$$V^{\pi_e} - \mathbb{E}[V^{\tilde{\pi}}] \le H\epsilon_o + (\delta + 1)\sqrt{\epsilon_e(1 - \epsilon_s)} \tag{6}$$

where ϵ_o, ϵ_e, and ϵ_s are the missing mass, defined as

$$\epsilon_o \doteq \mathbb{E}_{\mathcal{D}_e, \mathcal{D}_s}\big[\mathbb{E}_{s\sim\mu}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e) \cup \mathcal{S}_1(\mathcal{D}_s))\big]\big] \tag{7}$$

$$\epsilon_e \doteq \mathbb{E}_{\mathcal{D}_e}\big[\mathbb{E}_{s\sim\mu}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\big]\big] \tag{8}$$

$$\epsilon_s \doteq \mathbb{E}_{\mathcal{D}_s}\big[\mathbb{E}_{s\sim\mu}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_s))\big]\big]. \tag{9}$$

Sketch of proof. The error stems from the initial states that are not covered by $\mathcal{S}_1(\mathcal{D}_e)$. We bound the errors generated from the states not in $\mathcal{S}_1(\mathcal{D}_e) \cup \mathcal{S}_1(\mathcal{D}_s)$ and from the states in $\mathcal{S}_1(\mathcal{D}_s) \backslash \mathcal{S}_1(\mathcal{D}_e)$ by $H\epsilon_o$ and $(\delta+1)\sqrt{\epsilon_e(1-\epsilon_s)}$, respectively. Combining these two errors yields the result. For a detailed proof, please refer to Appendix C.1. □

The missing mass means the probability mass contributed by the states never observed in the corresponding set. Recall that n_e and n_s denote the numbers of trajectories and transitions in $\mathcal{D}_e$ and $\mathcal{D}_s$, respectively. Building on Theorem 4.2, we have the following result on sample complexity.

Corollary 4.3. *For any finite and episodic MDP with deterministic transition dynamics, the following fact holds:*

$$V^{\pi_e} - \mathbb{E}[V^{\tilde{\pi}}] \le \frac{|\mathcal{S}|H}{e(n_e + n_s)} + (\delta + 1) \cdot \sqrt{\frac{|\mathcal{S}|}{e n_e}}$$

where e denotes the Euler's number. Moreover, with a sufficiently large n_s, to obtain an ε-optimal policy, $\tilde{\pi}$ requires at most $\mathcal{O}(\min\{|\mathcal{S}|/\varepsilon^2, |\mathcal{S}|H/\varepsilon\})$ expert trajectories.

Sketch of proof. The result is concluded via quantifying the missing mass in terms of n_e and n_s (see Appendix C.2). □

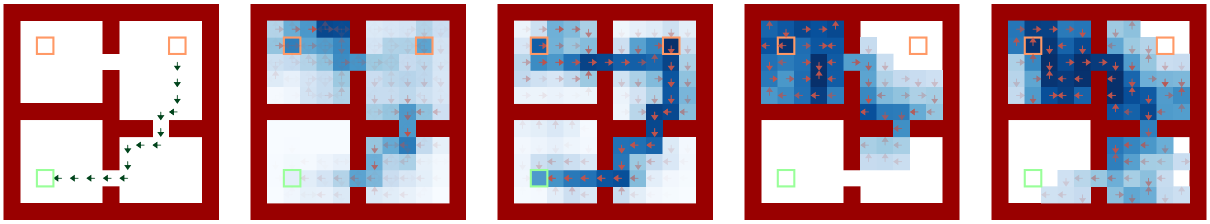
Remark 4.4. It is known that the minimax expected suboptimality of BC is limited to $\mathcal{O}(|S|H/n_e)$ in this setting (Rajaraman et al., 2020; Xu et al., 2021), a linear dependency on the episode horizon. This is because μ may largely differ from $\mathcal{S}(\mathcal{D}_e)$; when the BC policy encounters an initial state far outside $\mathcal{S}(\mathcal{D}_e)$, it will be essentially forced to take an arbitrary action in this state, potentially leading to compounding mistakes over H time steps.

Remark 4.5. As stated in Theorem 4.2 and Corollary 4.3, with sufficient $\mathcal{D}_s$, $\tilde{\pi}$ achieves an expected suboptimality of $\mathcal{O}(\min\{\sqrt{|\mathcal{S}|/n_e}, |\mathcal{S}|H/n_e\})$, superior to BC especially with large state spaces, long horizons, and limited expert data.[3] Thanks to the independency of H in the first term, $\tilde{\pi}$

[1] The setting covers many practical environments like MuJoCo.

[2] Throughout this paper, we use $(s, a, \dots, (s'), (a')) \in \mathcal{D}$ to denote that dataset $\mathcal{D}$ contains sub-trajectory $(s, a, \dots, (s'), (a'))$.

[3] Due to following expert behaviors in $\mathcal{S}(\mathcal{D}_e)$, the suboptimality of $\tilde{\pi}$ is also bounded by $|\mathcal{S}|H/n_e$ (see Appendix C.2 for details).



(a) Expert data (b) Imperfect data (c) `ILID` trajectories (d) `DWBC` trajectories (e) `ISWBC` trajectories

Figure 2. An illustration on the impact of limited expert state coverage in the Four Rooms domains (Sutton et al., 1999; Lee et al., 2021). The initial and goal states are represented as orange and green squares, respectively. The maximum trajectory length is 50. (a) depicts the given expert demonstration, which only covers one initial state. (b) shows an imperfect dataset, where the opacity of each square is determined by the empirical state marginal of imperfect data, and the opacity of each arrow represents the empirical action density in a state. (c)-(e) show the empirical trajectory distributions induced by rolling out the policies in the environment from the *left* initial state (beyond expert data). The policies are learned by `ILID`, `DWBC`, and `ISWBC` using both the expert and imperfect data, respectively. In (c)-(e), an arrow denotes the action with the maximum frequency in each state.

provably alleviates the error compounding and is robust to initial state perturbations. The underlying rationale is that $\mathcal{D}_s$ empowers $\tilde{\pi}$ to recover from 'mistakes': in the states beyond $\mathcal{S}(\mathcal{D}_e)$, albeit without expert guidance, the policy could take actions capable of returning to $\mathcal{S}(\mathcal{D}_e)$ where it exactly knows expert behaviors. In fact, this is very similar to human decision-making: when lost, we always want to get back to familiar roads; when a machine malfunctions, we aim to restore it to normalcy as soon as possible.

Practical behavior selection. Hypothesis 4.1 implies that resultant states can serve as a criterion for selecting imperfect behaviors – positive behaviors can be identified according to whether their resultant states fall within the expert state manifold. As an example, if there is an imperfect subtrajectory $(s_1, a_1, s_2, a_2, s_3) \in \mathcal{D}_b$ such that $s_3 \in \mathcal{D}_e$, we can treat (s_1, a_1) and (s_2, a_2) as positive behaviors, even without resemblance to any $(s, a) \in \mathcal{D}_e$. Guided by this, we first train a *state-only* discriminator $d : \mathcal{S} \times \mathcal{A} \rightarrow (0, 1)$ to contrast expert and non-expert states in $\mathcal{D}_b$:

$$\max_d \mathbb{E}_{s\sim\mathcal{D}_e}\big[\log d(s)\big] + \mathbb{E}_{s\sim\mathcal{D}_u}\big[\log(1 - d(s))\big] \qquad (10)$$

with $\mathcal{D}_u = \mathcal{D}_e \cup \mathcal{D}_b$. From Goodfellow et al. (2014), the optimal discriminator d^* satisfies

$$d^*(s) = \mathcal{D}_e(s)/(\mathcal{D}_e(s) + \mathcal{D}_u(s)) \qquad (11)$$

where we overload notation, denoting $\mathcal{D}_e(s)$ and $\mathcal{D}_u(s)$ as the empirical state marginals in $\mathcal{D}_e$ and $\mathcal{D}_u$, respectively. Building on Eq. (11), given a small positive threshold $\sigma > 0$, if $s \in \mathcal{D}_b$ and $d^*(s) > \sigma$, we identify s as an expert state; otherwise, we treat it as a non-expert one.

Based on the extracted expert states, we in turn select their *causal state-actions* to construct complementary dataset $\mathcal{D}_s$. Recall $\mathcal{D}_b = \{\tau_i\}_{i=1}^{n_b}$ with $\tau_i = (s_{i,1}, a_{i,1}, \dots, s_{i,H}, a_{i,H})$. If there exist $s_{i,h} \in \mathcal{D}_b$ such that $d^*(s_{i,h}) \geq \sigma$ for $h > 1$ and $i \in \{1, \dots, n_b\}$, we include K causal state-action pairs of $s_{i,h}$ into $\mathcal{D}_s$ as follows:

$$\mathcal{D}_s \leftarrow \mathcal{D}_s \cup \big\{(k, s_{i,h-k}, a_{i,h-k})\big\}_{k=1:\min\{h-1,K\}} \qquad (12)$$

where $K \in \{1, 2, \dots\}$ is termed as the *rollback step*. We iterate the above process for all identified expert states. For clarity, the process is depicted in Fig. 3.

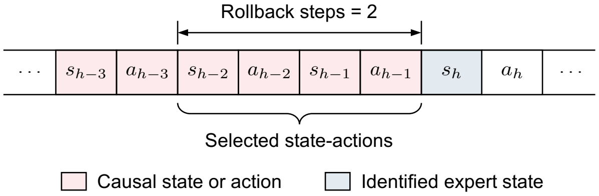


Figure 3. An illustration of our behavior selection.

Our behavior selection scheme possesses the following advantages. 1) The resultant state is informative, capable of effectively identifying both positive diverse behaviors and expert behaviors in $\mathcal{D}_b$. This can be easily seen from the fact that for an expert transition $(s_e, a_e, s'_e) \in \mathcal{D}_e \cap \mathcal{D}_b$, the identification of $s'_e \in \mathcal{S}(\mathcal{D}_e)$ ensures the selection of its causal expert behavior (s_e, a_e). 2) It explicitly utilizes the dynamics information in $\mathcal{D}_b$, enabling $\mathcal{D}_s$ to cover a relatively large portion of $\mathcal{D}_b$ (with m identified expert states, $\mathcal{D}_s$ can include approximately mK selected state-actions), thus significantly enhancing the utilization of imperfect demonstrations. 3) The method is easy to implement. Given that the computation in data selection primarily resides in training the discriminator, which is straightforward, it is highly applicable in practical, high-dimensional environments.

4.2. Learning from Expert and Selected Behaviors

After obtaining $\mathcal{D}_s$, a natural solution to learn an imitation policy is carrying out `BC` from the union of $\mathcal{D}_e$ and $\mathcal{D}_s$.

Algorithm 1 ILID

```
Require: Expert data $\mathcal{D}_e$, imperfect data $\mathcal{D}_b$, rollback $K$
Initialize policy parameter $\theta$
Train discriminators $d^*$ and $D^*$ by Eqs. (10) and (15)
// Data selection
Build complementary dataset $\mathcal{D}_s$ by Eq. (12)
// Policy extraction
for $i = 1$ to $n$ do
    $\theta \leftarrow \theta + \eta \tilde{\nabla} J(\pi_\theta)$
end for
```

However, due to the suboptimality of $\mathcal{D}_s$, this solution may suffer from potential *interference* among actions. That is, for a selected (s, a, s'), if $s, s' \in \mathcal{D}_e$ but $a \neq \pi_e(s)$, action a will affect mimicking the expert behavior in expert state s when learning from the union data (see Fig. 8(e)). Thus, it necessitates exactly following the expert in given expert states (it has been implied by the definition of $\tilde{\pi}$ in Eq. (5)).

To this end, we cast the policy learning as the following weighted behavior cloning problem:

$$\max_{\pi} \mathbb{E}_{\mathcal{D}_e}[\log(\pi(a|s))] + \mathbb{E}_{\mathcal{D}_s}[\mathbb{1}(\mathcal{D}_e(s) = 0) \log(\pi(a|s))]$$

where the expectation is taken w.r.t. state-action (s, a), and $\mathcal{D}_e(s)$ denotes the empirical state marginals in $\mathcal{D}_e$. In the problem, the first term matches BC, and the second term aims to clone the selected behaviors *outside* the expert state manifold, which essentially discards the suboptimal actions in expert states. Of note, albeit with a Dirichlet function in the second term, based on Eq. (11), it can be well approximated via the output of d^*. In practice, we instantiate the above objective as follows:

$$\begin{aligned} \max_{\pi} J(\pi) \doteq\ & \mathbb{E}_{\mathcal{D}_u}[\alpha(s, a) \log(\pi(a|s))] \\ & + \mathbb{E}_{\mathcal{D}_s}[\beta(s, a) \log(\pi(a|s))] \end{aligned} \tag{13}$$

with $\mathcal{D}_u = \mathcal{D}_e \cup \mathcal{D}_b$. In Eq. (13), we exploit the trick of importance sampling (which is unbiased) to enhance the expert data support, as in Li et al. (2023):

$$\alpha(s, a) \doteq \frac{\mathcal{D}_e(s, a)}{\mathcal{D}_u(s, a)} = \frac{D^*(s, a)}{1 - D^*(s, a)} \tag{14}$$

where another discriminator D^* is obtained by solving

$$\max_{D} \mathbb{E}_{\mathcal{D}_e}[\log D(s, a)] + \mathbb{E}_{\mathcal{D}_u}[\log(1 - D(s, a))]. \tag{15}$$

In addition, $\beta(s, a)$ approximates the Dirichlet function by

$$\beta(s, a) \doteq \mathbb{1}(d^*(s) \leq \sigma). \tag{16}$$

In summary, we term our algorithm *offline Imitation Learning with Imperfect Demonstrations* (ILID) with the pseudocode outlined in Algorithm 1, which can be easily implemented on top of BC and enjoys fast convergence speed and training stability (see Section 5).

5. Experiments

In this section, we carry out extensive experiments to evaluate our proposed method and answer the following key questions: **1)** Can ILID effectively utilize imperfect demonstrations, especially in complex, high-dimensional environments? **2)** How does ILID perform given different numbers of expert demonstrations or varying qualities of imperfect demonstrations? **3)** What are the effects of components and hyperparameters such as $\alpha(s, a)$, $\beta(s, a)$, and K? Experimental details are elaborated in Appendix A.[4]

Baselines. We evaluate our method against six strong baseline methods in offline IL: **1)** BCE, the standard BC trained only on expert demonstrations; **2)** BCU, BC trained on union data; **3)** DWBC (Xu et al., 2022), an offline IL method that leverages suboptimal demonstrations by jointly training a discriminator to re-weight the BC objective; **4)** ISWBC (Li et al., 2023), an offline IL method that adopts importance sampling to enhance BC; **5)** CSIL (Watson et al., 2024), a model-free IRL method that learns a shaped reward function using the BC policy; **6)** MLIRL (Zeng et al., 2023), a model-based offline IRL algorithm based on bi-level optimization.

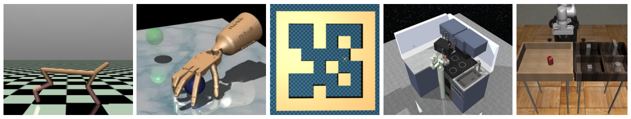


Figure 4. Benchmark environments. From left to right: MuJoCo, Adroit, AntMaze, FrankaKitchen, and vision-based Robomimic. We also consider vision-based MuJoCo with image observations.

Environments and datasets. We run experiments with 6 domains including 21 tasks: 1) AntMaze (umaze, medium, large), 2) Adroit (pen, hammer, door, relocate), 3) MuJoCo (ant, hopper, halfcheetah, walker2d), 4) FrankaKitchen (complete, partial, undirect), 5) vision-based Robomimic (lift, can, square), and 6) vision-based MuJoCo. We employ the D4RL datasets (Fu et al., 2020) for AntMaze, MuJoCo, Adroit, and FrankaKitchen and use the robomimic (Mandlekar et al., 2021) datasets for vision-based Robomimic. In addition, we construct vision-based MuJoCo datasets using the method introduced in Fu et al. (2020). Details on environments and datasets can be found in Appendices A.1 and A.2.

Performance measure. We train a policy using 3 random seeds and evaluate it by running it in the environment for 10 episodes and computing the average undiscounted return of the environment reward. Akin to Fu et al. (2020), we use the normalized scores in figures and tables, which are measured by $\texttt{score} = 100 \times \frac{\texttt{score}-\texttt{random\_score}}{\texttt{expert\_score}-\texttt{random\_score}}$.

Reproducibility. All details of our experiments are pro-

[4] The code is available at https://github.com/HansenHua/ILID-offline-imitation-learning.

Table 1. Normalized performance under limited expert demonstrations and low-quality imperfect data. The number of expert trajectories is 1 for MuJoCo and AntMaze, 10 for Adroit and FrankaKitchen, and 25 for vision-based MuJoCo and Robomimic; and the number of imperfect trajectories is 1000 across tasks. Uncertainty intervals depict standard deviation. The sampling datasets can be found in Table 5.

Task	BCE	BCU	DWBC	CSIL	MLIRL	ISWBC	ILID (ours)
ant	-15.6 ± 7.0	31.4 ± 0.1	23.4 ± 7.1	0.2 ± 0.0	35.9 ± 9.3	27.1 ± 6.7	$\mathbf{62.7 \pm 4.1}$
halfcheetah	0.4 ± 1.0	2.3 ± 0.0	0.9 ± 1.3	15.1 ± 4.3	21.5 ± 0.8	12.6 ± 2.4	$\mathbf{32.4 \pm 2.4}$
hopper	16.7 ± 4.3	7.7 ± 6.0	$\mathbf{78.3 \pm 10.9}$	16.1 ± 3.7	55.2 ± 14.6	73.1 ± 8.9	68.9 ± 4.8
walker2d	7.1 ± 5.4	0.3 ± 0.1	46.1 ± 9.8	8.9 ± 4.2	23.5 ± 1.2	39.8 ± 1.9	$\mathbf{58.4 \pm 4.8}$
hammer	4.5 ± 5.3	0.2 ± 0.0	14.6 ± 12.6	15.3 ± 7.1	0.2 ± 0.0	3.8 ± 3.0	$\mathbf{51.0 \pm 2.4}$
pen	40.0 ± 9.6	2.8 ± 7.8	36.0 ± 18.9	22.1 ± 0.2	17.2 ± 3.6	31.8 ± 0.0	$\mathbf{75.1 \pm 5.2}$
relocate	-0.1 ± 0.1	-0.1 ± 0.0	-0.1 ± 0.0	4.0 ± 3.2	0.2 ± 0.0	0.2 ± 0.0	$\mathbf{28.2 \pm 1.6}$
door	2.9 ± 2.1	-0.1 ± 0.0	-0.1 ± 0.1	16.7 ± 7.1	0.2 ± 0.0	0.2 ± 0.0	$\mathbf{25.9 \pm 1.1}$
antmaze-umaze	3.6 ± 0.0	3.6 ± 0.0	22.0 ± 2.7	12.0 ± 3.2	6.4 ± 0.3	9.9 ± 1.1	$\mathbf{72.3 \pm 3.8}$
antmaze-medium	0.2 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	6.4 ± 0.3	$\mathbf{64.6 \pm 5.2}$
antmaze-large	0.2 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	4.8 ± 0.0	$\mathbf{39.8 \pm 2.5}$
undirect	0.2 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	35.0 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	$\mathbf{52.8 \pm 3.1}$
partial	0.2 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	21.7 ± 1.4	0.2 ± 0.0	0.2 ± 0.0	$\mathbf{32.5 \pm 2.6}$
complete	0.2 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	11.7 ± 0.0	0.2 ± 0.0	0.2 ± 0.0	$\mathbf{29.9 \pm 1.7}$
ant-img	16.0 ± 4.1	15.6 ± 2.4	17.6 ± 3.2	10.7 ± 2.4	0.0 ± 0.0	19.2 ± 2.1	$\mathbf{31.5 \pm 4.0}$
halfcheetah-img	26.6 ± 3.2	27.9 ± 4.7	18.5 ± 6.4	25.3 ± 4.8	0.0 ± 0.0	23.5 ± 1.5	$\mathbf{41.6 \pm 3.2}$
hopper-img	12.8 ± 4.0	10.9 ± 5.2	16.7 ± 5.6	11.8 ± 4.0	0.0 ± 0.0	15.4 ± 6.3	$\mathbf{61.5 \pm 5.0}$
walker2d-img	8.3 ± 2.0	7.7 ± 6.3	22.8 ± 5.0	7.5 ± 5.5	0.0 ± 0.0	27.9 ± 3.3	$\mathbf{58.9 \pm 4.4}$
can-img	13.7 ± 9.6	21.4 ± 2.4	21.9 ± 1.4	23.3 ± 3.2	0.0 ± 0.0	9.8 ± 11.9	$\mathbf{38.8 \pm 2.7}$
lift-img	48.5 ± 4.9	28.9 ± 3.3	46.6 ± 5.7	35.9 ± 1.7	0.0 ± 0.0	56.9 ± 2.4	$\mathbf{90.4 \pm 2.4}$
square-img	2.0 ± 1.6	5.0 ± 4.1	11.5 ± 2.2	5.0 ± 3.3	0.0 ± 0.0	13.2 ± 1.4	$\mathbf{37.8 \pm 3.0}$

vided in the appendices in terms of the tasks, network architectures, hyperparameters, etc. We implement all baselines and environments based on open-source repositories. Of note, our method is robust in hyperparameters – they are *identical* for all tasks except for the change of neural nets to CNNs in vision-based domains.

Comparative results. To answer the first question, we evaluate ILID's performance in each task using limited expert demonstrations and a set of low-quality imperfect data. For example, in the MuJoCo domain, we sample 1 expert trajectory and 1000 random trajectories from D4RL as the expert and imperfect data, respectively (refer to Table 5 for the complete data setup). Comparative results are presented in Table 1, and learning curves are depicted in Figs. 14 and 15. We find ILID consistently outperforms baselines in **20/21** tasks often by a significant margin while enjoying fast and stabilized convergence. Due to limited state coverage of expert data and low quality of imperfect data, BCE and BCU fail to fulfill most of the tasks. This reveals ILID's effectiveness in extracting and leveraging positive behaviors from imperfect demonstrations. DWBC and ISWBC exhibit similar performances, demonstrating relative success in MuJoCo but facing challenges in robotic manipulation and maze domains, which require precise long-horizon manipulation. This is because the similarity-based behavior selection confines their training data to the expert states with narrow coverage, rendering them prone to error compounding. In contrast, ILID, utilizing dynamics information, can *stitch* parts of trajectories and empower the policy to recover from mistakes. In addition, the IRL methods struggle in high-dimensional environments owing to reward extrapolation and world model estimates.

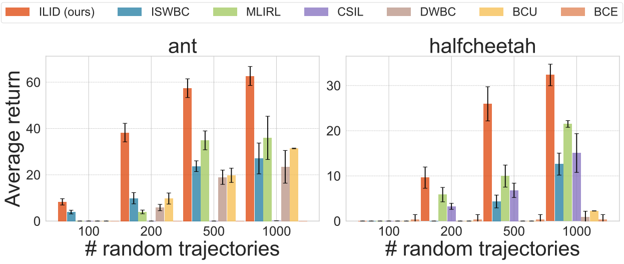


Figure 5. Performance with 1 expert trajectory and varying numbers of random trajectories.

Expert demonstrations. To answer the second question, we run experiments with varying numbers of expert trajectories (ranging from 1 to 30 in MuJoCo and AntMaze, from 10 to 300 in Adroit and FrankaKitchen, and from 25 to 200 in vision-based domains). The data setup adheres to that of Table 5. We present selected results in Fig. 6 and the full results in Fig. 16 of Appendix B.2. Our method, consistently requiring much fewer expert trajectories to attain expert

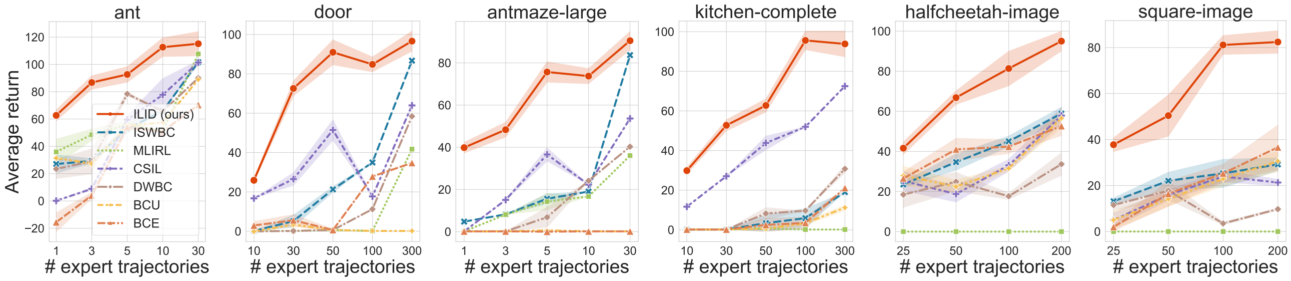
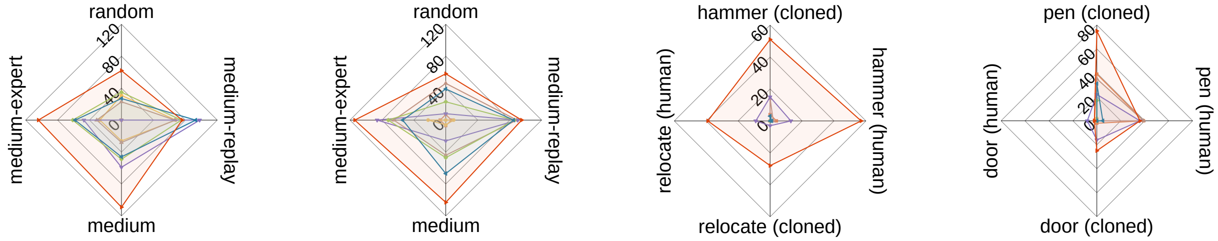


Figure 6. Normalized scores under varying numbers of expert demonstrations.



(a) ant (b) walker2d (c) hammer & relocate (d) pen & door

Figure 7. Comparative performance under varying qualities of imperfect demonstrations. Each axis represents a specific data quality, where the values denote normalized scores of methods. The correspondence between methods and colored lines can be found in Fig. 6.

performance, demonstrates great demonstration efficiency in comparison with prior methods.

Quality and quantity of imperfect data. For the second question, we also conduct experiments using imperfect demonstrations with varying qualities and quantities to test the robustness of ILID's performance in behavior selection (the data setup is showcased in Table 6). Selected results are shown in Figs. 5 and 7, with complete results provided in Figs. 17 and 18 of Appendix B.3. We find that ILID surpasses the baselines in **20/24** settings, corroborating its efficacy and superiority in the utilization of noisy data. Moreover, Fig. 5 underscores the importance of leveraging suboptimal data.

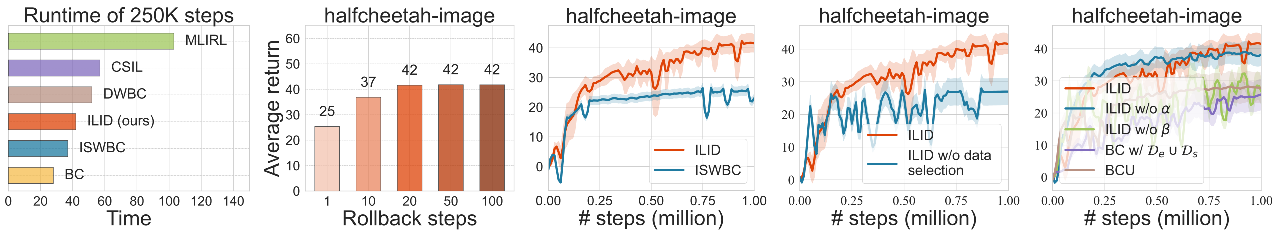
Rollback steps. Regarding the fourth question, we vary K from 1 to 100 and run experiments across all benchmarks. A selected result is shown in Fig. 8(b) and full results are depicted in Fig. 19 of Appendix B.4. The results clearly indicate that as K increases, there is an initial improvement in performance; once it reaches a sufficiently large value, performance tends to stabilize. Considering that a larger rollback step leads to more selected behaviors capable of reaching expert states, this observation offers support for Hypothesis 4.1. Importantly, the performance proves to be robust to a relatively large K, rendering ILID forgiving to the hyperparameter.

Ablation studies. We assess the effect of key components by ablating them on *all* benchmarks, under the same setting as that of Table 5 (see Appendix B.6 for complete results). *1) Only importance-sampling weighting.* Without the second term in Problem (13), ILID reduces to ISWBC. Yet, as shown in Fig. 8(c) and aforementioned comparative experiments, ISWBC does not suffice satisfactory performance. *2) Effect of importance-sampling weighting.* We ablate importance weighting, and accordingly the first term of Problem (13) becomes the BC loss. The observed performance degradation in Fig. 8(e) suggests its benefits, which can enhance expert data support, particularly in continuous domains. *3) Importance of data selection.* We ablate the data selection and replace $\mathcal{D}_s$ in Problem (13) by entire imperfect data of $\mathcal{D}_b$. Fig. 8(d) corroborates Hypothesis 4.1 and underscores the importance of our data selection scheme. *4) Importance of* $\beta(s, a)$. As demonstrated in Fig. 8(e), $\beta(s, a)$ assumes a crucial role in imitating selected data. The absence of $\beta(s, a)$ (setting $\beta(s, a) \equiv 1$) renders training ineffective and unstable, due to behavior interference.

Runtime. We evaluate the runtime of ILID in comparison with baselines. Fig. 8(a) demonstrates ILID remains comparable wall-clock time to BC (see Appendix B.5).

6. Conclusion and Future Work

In this paper, we introduce a simple yet effective data selection method along with a lightweight behavior cloning algorithm, which can explicitly harness the dynamics information in imperfect data, significantly enhancing the utilization of imperfect demonstrations. A limitation of this work is the requirement of *state* overlap/similarity between the expert and imperfect data. While this assumption is weaker than most existing works (which necessitates *state-*



(a) Runtime (b) Rollback steps (c) Only ISW (d) On data selection (e) On weighting

Figure 8. Ablation studies and comparative results of wall-clock runtime in policy learning.

action overlap), there might be scenarios where only the expert can reach expert states. In general, it is hard to assess suboptimal behaviors persuasively if none of them bear a state resemblance to the expert's. A potential compromise is to involve prior information like the quality of imperfect data in the problem. This opens up an interesting future direction on offline IL with multi-quality demonstrations.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China under Grant No. 62341201, 62122095, 62072472, 62172445, and 62302260, by the National Key R&D Program of China under Grant No. 2022YFF0604502, by China Postdoctoral Science Foundation under Grant No. 2023M731956, and by a grant from the Guoqiang Institute, Tsinghua University.

Impact Statement

This paper presents work whose goal is to advance the field of Reinforcement Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Chan, A. J. and van der Schaar, M. Scalable bayesian inverse reinforcement learning. In *International Conference on Learning Representations*, 2021.

Chang, J., Uehara, M., Sreenivas, D., Kidambi, R., and Sun, W. Mitigating covariate shift in imitation learning via offline data with partial coverage. In *Advances in Neural Information Processing Systems*, volume 34, pp. 965–979. Curran Associates, 2021.

Cideron, G., Tabanpour, B., Curi, S., Girgin, S., Hussenot, L., Dulac-Arnold, G., Geist, M., Pietquin, O., and Dadashi, R. Get back here: Robust imitation by return-to-distribution planning. *arXiv preprint arXiv:2305.01400*, 2023.

Florence, P., Lynch, C., Zeng, A., Ramirez, O. A., Wahid, A., Downs, L., Wong, A., Lee, J., Mordatch, I., and Tompson, J. Implicit behavioral cloning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164, pp. 158–168. PMLR, 2022.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Garg, D., Chakraborty, S., Cundy, C., Song, J., and Ermon, S. IQ-Learn: Inverse soft-q learning for imitation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4028–4039. Curran Associates, 2021.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pp. 2672–2680. Curran Associates, 2014.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1861–1870. PMLR, 2018.

Herman, M., Gindele, T., Wagner, J., Schmitt, F., and Burgard, W. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pp. 102–110. PMLR, 2016.

Jarrett, D., Bica, I., and van der Schaar, M. Strictly batch imitation learning by energy-based distribution matching.

In *Advances in Neural Information Processing Systems*, volume 33, pp. 7354–7365. Curran Associates, 2020.

Kim, G.-H., Seo, S., Lee, J., Jeon, W., Hwang, H., Yang, H., and Kim, K.-E. DemoDICE: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022.

Klein, E., Geist, M., and Pietquin, O. Batch, off-policy and model-free apprenticeship learning. In *Recent Advances in Reinforcement Learning*, pp. 285–296. Springer Berlin Heidelberg, 2012a.

Klein, E., Geist, M., Piot, B., and Pietquin, O. Inverse reinforcement learning through structured classification. In *Advances in Neural Information Processing Systems*, volume 25, pp. 1007–1015. Curran Associates, 2012b.

Kostrikov, I., Nachum, O., and Tompson, J. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations*, 2020.

Lee, D., Srinivasan, S., and Doshi-Velez, F. Truly batch apprenticeship learning with deep successor features. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 5909–5915, 2019.

Lee, J., Jeon, W., Lee, B., Pineau, J., and Kim, K.-E. OptiDICE: Offline policy optimization via stationary distribution correction estimation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 6120–6130. PMLR, 2021.

Li, Z., Xu, T., Qin, Z., Yu, Y., and Luo, Z.-Q. Imitation learning from imperfection: Theoretical justifications and algorithms. In *The 37th Conference on Neural Information Processing Systems*, 2023.

Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., and Martín-Martín, R. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

Nakamoto, M., Zhai, Y., Singh, A., Ma, Y., Finn, C., Kumar, A., and Levine, S. Cal-QL: Calibrated offline rl pre-training for efficient online fine-tuning. In *The 37th Conference on Neural Information Processing Systems*, 2023.

Piot, B., Geist, M., and Pietquin, O. Boosted and reward-regularized classification for apprenticeship learning. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 1249–1256, 2014.

Pomerleau, D. A. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, volume 1, pp. 305–313. Morgan Kaufmann, 1988.

Rajaraman, N., Yang, L., Jiao, J., and Ramchandran, K. Toward the fundamental limits of imitation learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2914–2924. Curran Associates, 2020.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., and Russell, S. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. In *Advances in Neural Information Processing Systems*, volume 34, pp. 11702–11716. Curran Associates, 2021.

Sasaki, F. and Yamashina, R. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2021.

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

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

Swamy, G., Choudhury, S., Bagnell, J. A., and Wu, S. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 10022–10032. PMLR, 2021.

Watson, J., Huang, S., and Heess, N. Coherent soft imitation learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 14540–14583. Curran Associates, 2024.

Wu, Y.-H., Charoenphakdee, N., Bao, H., Tangkaratt, V., and Sugiyama, M. Imitation learning from imperfect demonstration. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 6818–6827. PMLR, 2019.

Xu, H., Zhan, X., Yin, H., and Qin, H. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 24725–24742. PMLR, 2022.

Xu, T., Li, Z., Yu, Y., and Luo, Z.-Q. On generalization of adversarial imitation learning and beyond. *arXiv preprint arXiv:2106.10424*, 2021.

Yu, T., Kumar, A., Chebotar, Y., Hausman, K., Finn, C., and Levine, S. How to leverage unlabeled data in offline reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 25611–25635. PMLR, 2022.

Yue, S., Wang, G., Shao, W., Zhang, Z., Lin, S., Ren, J., and Zhang, J. CLARE: Conservative model-based reward learning for offline inverse reinforcement learning. In *International Conference on Learning Representations*, 2023.

Yue, S., Deng, Y., Wang, G., Ren, J., and Zhang, Y. Federated offline reinforcement learning with proximal policy evaluation. *Chinese Journal of Electronics*, 33(6):1–13, 2024.

Zeng, S., Li, C., Garcia, A., and Hong, M. Maximum-likelihood inverse reinforcement learning with finite-time guarantees. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10122–10135. Curran Associates, 2022.

Zeng, S., Li, C., Garcia, A., and Hong, M. When demonstrations meet generative world models: A maximum likelihood framework for offline inverse reinforcement learning. In *The 37th Conference on Neural Information Processing Systems*, 2023.

Zolna, K., Novikov, A., Konyushkova, K., Gulcehre, C., Wang, Z., Aytar, Y., Denil, M., de Freitas, N., and Reed, S. Offline learning from demonstrations and unlabeled experience. In *NeurIPS Workshop on Offline Reinforcement Learning*, 2020.

A. Experimental Setup

In this section, we present full experimental details for reproducibility.

A.1. Benchmarks

We evaluate our method on a number of environments (Robomimic, MuJoCo, Adroit, FrankaKitchen, and AntMaze) which are widely used in prior studies (Nakamoto et al., 2023; Watson et al., 2024). We elaborate in what follows.

- **Vision-based Robomimic.** The Robomimic tasks (`lift`, `can`, `square`) involve controlling a 7-DoF simulated hand robot (Mandlekar et al., 2021), with pixelized observations as shown in Fig. 9. The robot is tasked with lifting objects, picking and placing cans, and picking up a square nut to place it on a rod from random initializations.

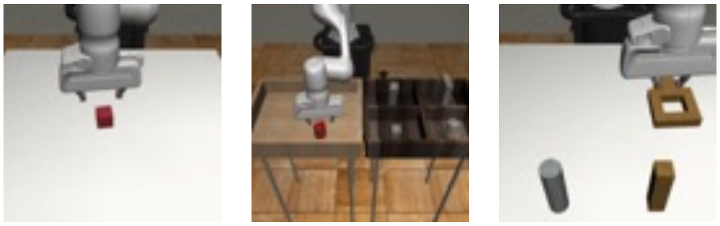


Figure 9. Observations of vision-based Robomimic tasks. From left to right: `lift`, `can`, `square`.

- **Vision-based MuJoCo.** The MuJoCo locomotion tasks (`ant`, `hopper`, `halfcheetah`, `walker2d`) are popular benchmarks used in existing work. In addition to the standard setting, we also consider vision-based MuJoCo tasks which uses the image observation as input (see Fig. 10).

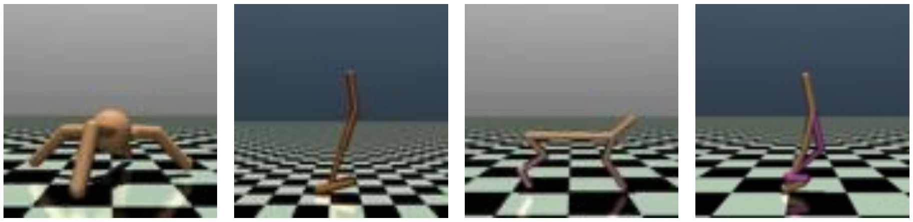


Figure 10. Observations of vision-based MuJoCo tasks. From left to right: `ant`, `hopper`, `halfcheetah`, `walker2d`.

- **Adroit.** The Adroit tasks (`hammer`, `door`, `pen`, and `relocate`) (Rajeswaran et al., 2017) involve controlling a 28-DoF hand with five fingers tasked with hammering a nail, opening a door, twirling a pen, or picking up and moving a ball.

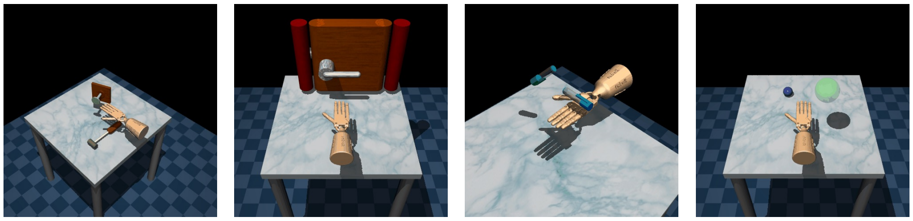


Figure 11. Adroit tasks: `hammer`, `door`, `pen`, and `relocate` (from left to right).

- **FrankaKitchen.** The FrankaKitchen tasks (`complete`, `partial`, `undirect`), proposed by Gupta et al. (2019), involve controlling a 9-DoF Franka robot in a kitchen environment containing several common household items: a microwave, a kettle, an overhead light, cabinets, and an oven. The goal of each task is to interact with the items to reach a desired goal configuration. In the `undirect` task, the robot requires opening the microwave. In the `partial` task, the robot must first open the microwave and subsequently move the kettle. In the `complete` task, the robot needs to open the microwave, move the kettle, flip the light switch, and slide open the cabinet door sequentially (see Fig. 12). These tasks are especially challenging since they require composing parts of trajectories, precise long-horizon manipulation, and handling human-provided teleoperation data.

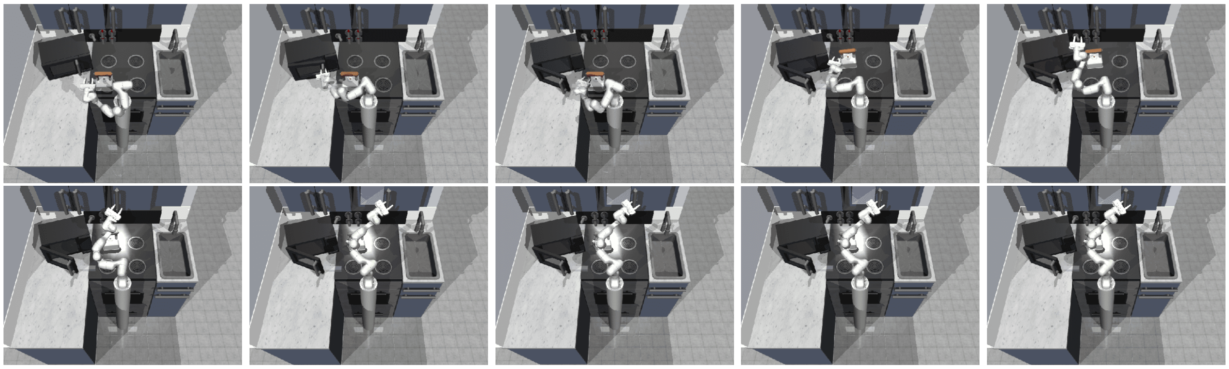


Figure 12. Visualized success for opening the microwave, moving the kettle, turning on the light switch, and sliding the slider.

- **AntMaze.** The AntMaze tasks require controlling an 8-Degree of Freedom (DoF) quadruped robot to move from a starting point to a fixed goal location (Fu et al., 2020). Three maze layouts (`umaze`, `medium`, and `large`) are provided from small to large.

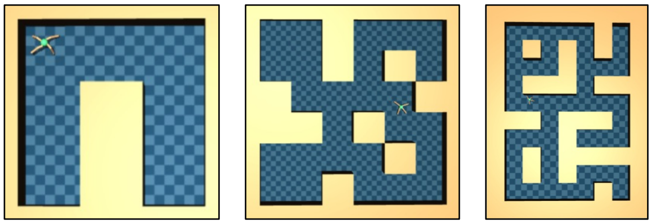


Figure 13. AntMaze with three maze layouts, `umaze`, `medium`, and `large` (from left to right).

Detailed information about the environments including observation space, action space, and expert performance is provided in Tables 2 and 3, where expert and random scores are averaged over 1000 episodes.

Table 2. Details of continuous-control tasks.

Task	State dim.	Action dim.	`random`*	`expert`*
`ant`	27	8	−325.60	3879.70
`halfcheetah`	17	6	−280.18	12135.00
`hopper`	11	3	−20.27	3234.30
`walker2d`	17	6	1.63	4592.30
`antmaze`	27	8	0.00	1.00
`door`	39	28	−56.51	2880.57
`hammer`	46	26	−274.86	12794.13
`pen`	45	24	96.26	3076.83
`relocate`	39	30	−6.43	4233.88
`FrankaKitchen`	59	9	0.00	1.00

* Average scores over 1000 trajectories of `expert` and `random`.

A.2. Datasets

We employ `D4RL` (Fu et al., 2020) for AntMaze, MuJoCo, Adroit, and FrankaKitchen, and use `robomimic` (Mandlekar et al., 2021) for Robomimic. Tables 5 and 6 specify the data setup used for each task across experiments. Of note, we construct vision-based MuJoCo datasets using the same method as Fu et al. (2020): the expert and imperfect data use video samples from a policy trained to completion with `SAC` (Haarnoja et al., 2018) and a randomly initialized policy, respectively.

A.3. Baselines

We evaluate our method against six strong baseline methods in offline IL: ***1)*** *Behavior Cloning with Expert Data* (`BCE`), the standard `BC` trained only on expert demonstrations; ***2)*** *Behavior Cloning with Union Data* (`BCU`), `BC` trained on union data; ***3)*** *Discriminator-Weighted Behavioral Cloning* (`DWBC`) (Xu et al., 2022), an offline IL method that leverages suboptimal demonstrations by jointly training a discriminator to re-weight the `BC` objective (`https://github.com/ryanxhr/`

Table 3. Details of vision-based tasks.

Task	State dim.	Action dim.	random	expert
ant	(84×84)	8	-325.60	3879.70
halfcheetah	(84×84)	6	-280.18	12135.00
hopper	(84×84)	3	-20.27	3234.30
walker2d	(84×84)	6	1.63	4592.30
lift	(84×84)	7	0.00	1.00
can	(84×84)	7	0.00	1.00
square	(84×84)	7	0.00	1.00

Table 4. Hyperparameters across tasks.

Hyperparameter	Value
# Neural net layers	2
Optimizer	Adam
Activation	ReLU
Batchsize	256
All learning rates	1e-5
Threshold σ	0.2
Rollback K	20

DWBC); ***4)*** *Importance-Sampling-Weighted Behavioral Cloning* (ISWBC) (Li et al., 2023), an offline IL method that adopts importance sampling to enhance BC (https://github.com/liziniu/ISWBC); ***5)*** *Coherent Soft Imitation Learning* (CSIL) (Watson et al., 2024), a model-free IRL method that learns a shaped reward function by entropy-regularized BC (https://joemwatson.github.io/csil); ***6)*** *Maximum Likelihood-Inverse Reinforcement Learning* (MLIRL) (Zeng et al., 2023), a recent model-based offline IRL algorithm based on bi-level optimization (https://github.com/Cloud0723/Offline-MLIRL).

We implement and tune baseline methods based on their publicly available implementatinons with the same policy network structures. The tuned codes are included in the supplementary material.

A.4. Implementation

Our method is straightforward to implement and robust to hyperparameters (which are consistent across all benchamarks and settings). We represent the policy as a 2-layer feedforward neural network with 256 hidden units, ReLU activation functions, and Tanh Gaussian outputs. Analogously, the discriminators are represented as a 2-layer feedforward neurl net with 256 hidden units, ReLU activations with the output clipped to $[0.1, 0.9]$. For vision-based tasks, we change the network architectures to a simple CNN, consisting of two convolutional layers, each with a 3×3 convolutional kernel and 2×2 max pooling. We adopt Adam as the optimizer. All learning rates and batchsizes are set to 1e-5 and 256, respectively. The thresholds σ for identifying expert states is set to 0.2, and the rollback step K is set to 20. The hyperparameters are summarized in Table 4.

We implement our code using Pytorch 1.8.1, built upon the open-source framework of offline RL algorithms, provided at https://github.com/tinkoff-ai/CORL (under the Apache-2.0 License) and the implementation of DWBC, provided at https://github.com/ryanxhr/DWBC (under the MIT License). All the experiments are run on Ubuntu 20.04.2 LTS with 8 NVIDIA GeForce RTX 4090 GPUs.

Table 5. Data used in the comparative experiment.

Domain	Dataset	Traj. length	Task	Expert data	# Expert traj.	Imperfect data	# Imperfect traj.
MuJoCo	D4RL	≤ 1000	ant	ant-expert-v2	1	ant-random-v2	1000
			hopper	hopper-expert-v2	1	hopper-random-v2	1000
			halfcheetah	halfcheetah-expert-v2	1	halfcheetah-random-v2	1000
			walker2d	walker2d-expert-v2	1	walker2d-random-v2	1000
AntMaze	D4RL	≤ 100	umaze	antmaze-umaze-v0	1	antmaze-umaze-diverse-v0	1000
			medium	antmaze-medium-v0	1	antmaze-medium-diverse-v0	1000
			large	antmaze-large-v0	1	antmaze-large-diverse-v0	1000
Adroit	D4RL	≤ 100	pen	pen-expert-v1	10	pen-cloned-v1	1000
			hammer	hammer-expert-v1	10	hammer-cloned-v1	1000
			door	door-expert-v1	10	door-cloned-v1	1000
			relocate	relocate-expert-v1	10	relocate-cloned-v1	1000
FrankaKitchen	D4RL	≤ 280	complete	kitchen-complete-v0	10	kitchen-mixed-v0	1000
			partial	kitchen-partial-v0	10	kitchen-mixed-v0	1000
			indirect	kitchen-partial-v0	10	kitchen-mixed-v0	1000
Robomimic	robomimic	≤ 500	lift	lift-proficient-human	25	lift-paired-bad	1000
			can-paired-bad	can-proficient-human	25	can-paired-bad	1000
			square-paired-bad	square-proficient-human	25	square-paired-bad	1000
MuJoCo (vision)	-	≤ 1000	ant	ant-expert-vision	25	ant-random-vision	1000
			hopper	hopper-expert-vision	25	hopper-random-vision	1000
			halfcheetah	halfcheetah-expert-vision	25	halfcheetah-random-vision	1000
			walker2d	walker2d-expert-vision	25	walker2d-random-vision	1000

[1] In vision-based MuJoCo, we collect the expert-vision and random-vision data use video samples from a policy trained to completion with SAC and a randomly initialized policy, respectively.

Table 6. Data used in the experiment on varying data qualities.

Task	Trajectory length	# Expert trjectories	Expert data	# Imperfect trjectories	Imperfect data	Score
`ant`	≤ 1000	1	`ant-expert-v2`	1000	`ant-random-v2`	9.2
					`ant-medium-replay-v2`	19.0
					`ant-medium-v2`	80.3
					`ant-medium-expert-v2`	90.1
`halfcheetah`	≤ 1000	1	`halfcheetah-expert-v2`	1000	`ant-random-v2`	-0.1
					`ant-medium-replay-v2`	7.3
					`ant-medium-v2`	40.7
					`ant-medium-expert-v2`	70.3
`hopper`	≤ 1000	1	`hopper-expert-v2`	1000	`ant-random-v2`	1.2
					`ant-medium-replay-v2`	6.8
					`ant-medium-v2`	44.1
					`ant-medium-expert-v2`	72.0
`walker2d`	≤ 1000	1	`walker2d-expert-v2`	1000	`ant-random-v2`	0.0
					`ant-medium-replay-v2`	13.0
					`ant-medium-v2`	62.0
					`ant-medium-expert-v2`	81.0
`hammer`	≤ 100	10	`hammer-expert-v1`	1000	`pen-human-v1`	2.7
					`pen-cloned-v1`	0.5
`pen`	≤ 100	10	`pen-expert-v1`	1000	`hammer-human-v1`	2.1
					`hammer-cloned-v1`	59.9
`door`	≤ 100	10	`door-expert-v1`	1000	`door-human-v1`	2.6
					`door-cloned-v1`	-0.1
`relocate`	≤ 100	10	`relocate-expert-v1`	1000	`relocate-human-v1`	2.3
					`relocate-cloned-v1`	-0.1

B. Complete Experimental Results

This section provides complete experimental results to answer the questions raised in Section 5.

B.1. Comparative Experiments

To answer the first question, we evaluate ILID's performance in each task using limited expert demonstrations and a set of low-quality imperfect data. For example, in the MuJoCo domain, we sample 1 expert trajectory and 1000 random trajectories from D4RL as the expert and imperfect data, respectively (refer to Table 5 for the complete data setup). Comparative results are presented in Table 1, and learning curves are depicted in Figs. 14 and 15. We find ILID consistently outperforms baselines in **20/21** tasks often by a significant margin while enjoying fast and stabilized convergence. Due to limited state coverage of expert data and low quality of imperfect data, BCE and BCU fail to fulfill most of the tasks. This reveals ILID's effectiveness in extracting and leveraging positive behaviors from imperfect demonstrations. DWBC and ISWBC exhibit similar performances, demonstrating relative success in MuJoCo but facing challenges in robotic manipulation and maze domains, which require precise long-horizon manipulation. This is because the similarity-based behavior selection confines their training data to the expert states with narrow coverage, rendering them prone to error compounding. In contrast, ILID, utilizing dynamics information, can *stitch* parts of trajectories and empower the policy to recover from mistakes. In addition, the IRL methods struggle in high-dimensional environments owing to reward extrapolation and world model estimates.

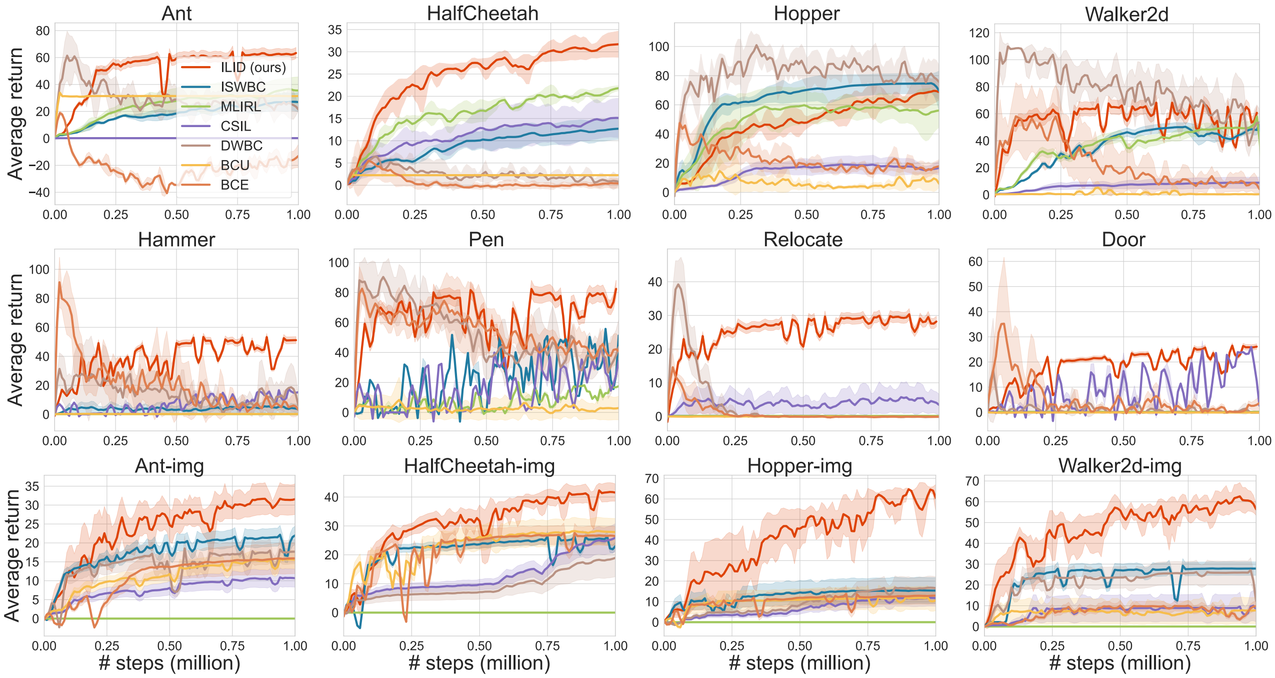


Figure 14. Learning curves for Table 1. '-img' represents vision-based MuJoCo tasks.

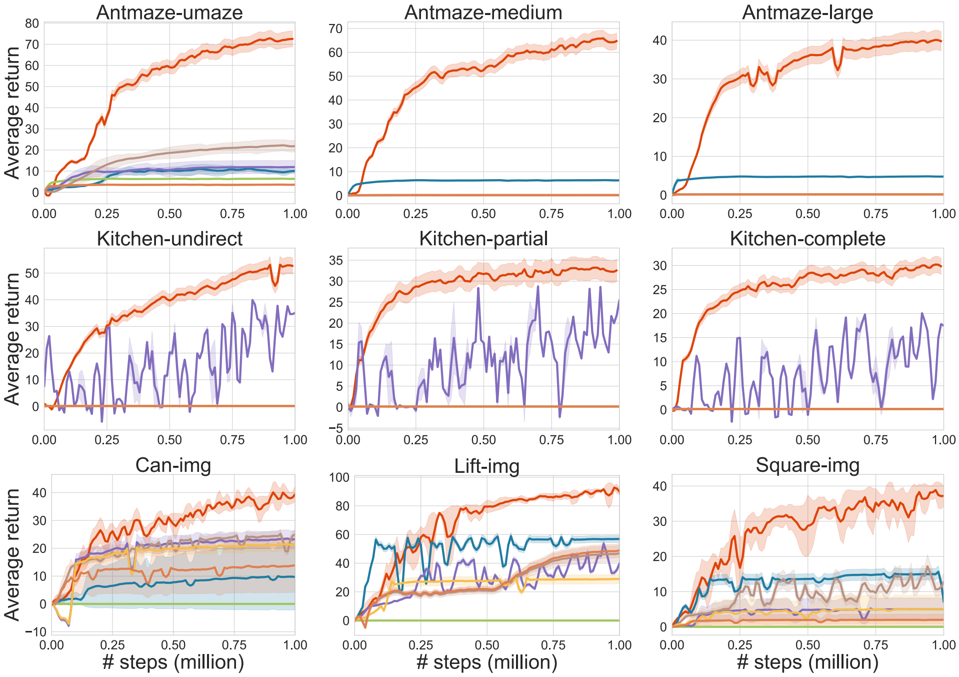


Figure 15. Learning curves for Table 1. Uncertainty intervals depict standard deviation over three seeds.

B.2. Expert Demonstrations

To answer the second question, we run experiments with varying numbers of expert trajectories (ranging from 1 to 30 in MuJoCo and AntMaze, from 10 to 300 in Adroit and FrankaKitchen, and from 25 to 200 in vision-based domains). The data setup adheres to that of Table 5. As illustrated in Fig. 16, our method, consistently requiring much fewer expert trajectories to attain expert performance, demonstrates great demonstration efficiency in comparison with prior methods.

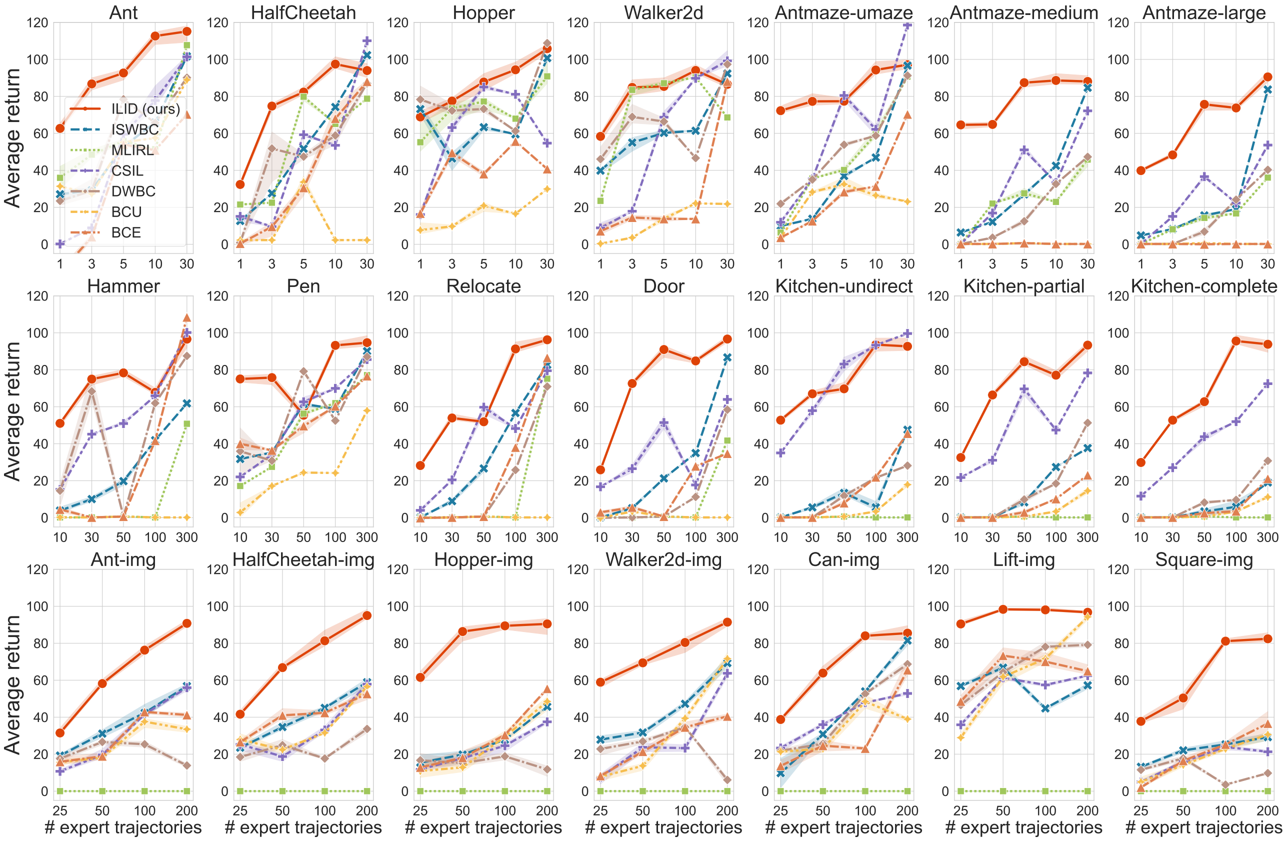


Figure 16. Normalized scores under varying numbers of expert demonstrations.

B.3. Quality and Quantity of Imperfect Data

For the second question, we also conduct experiments using imperfect demonstrations with varying qualities and quantities to test the robustness of `ILID`'s performance in behavior selection (the data setup is showcased in Table 6). As shown in Figs. 5, 7, 17 and 18, we find that `ILID` surpasses the baselines in **20/24** settings, corroborating its efficacy and superiority in the utilization of noisy data. Moreover, Fig. 5 underscores the importance of leveraging suboptimal data.

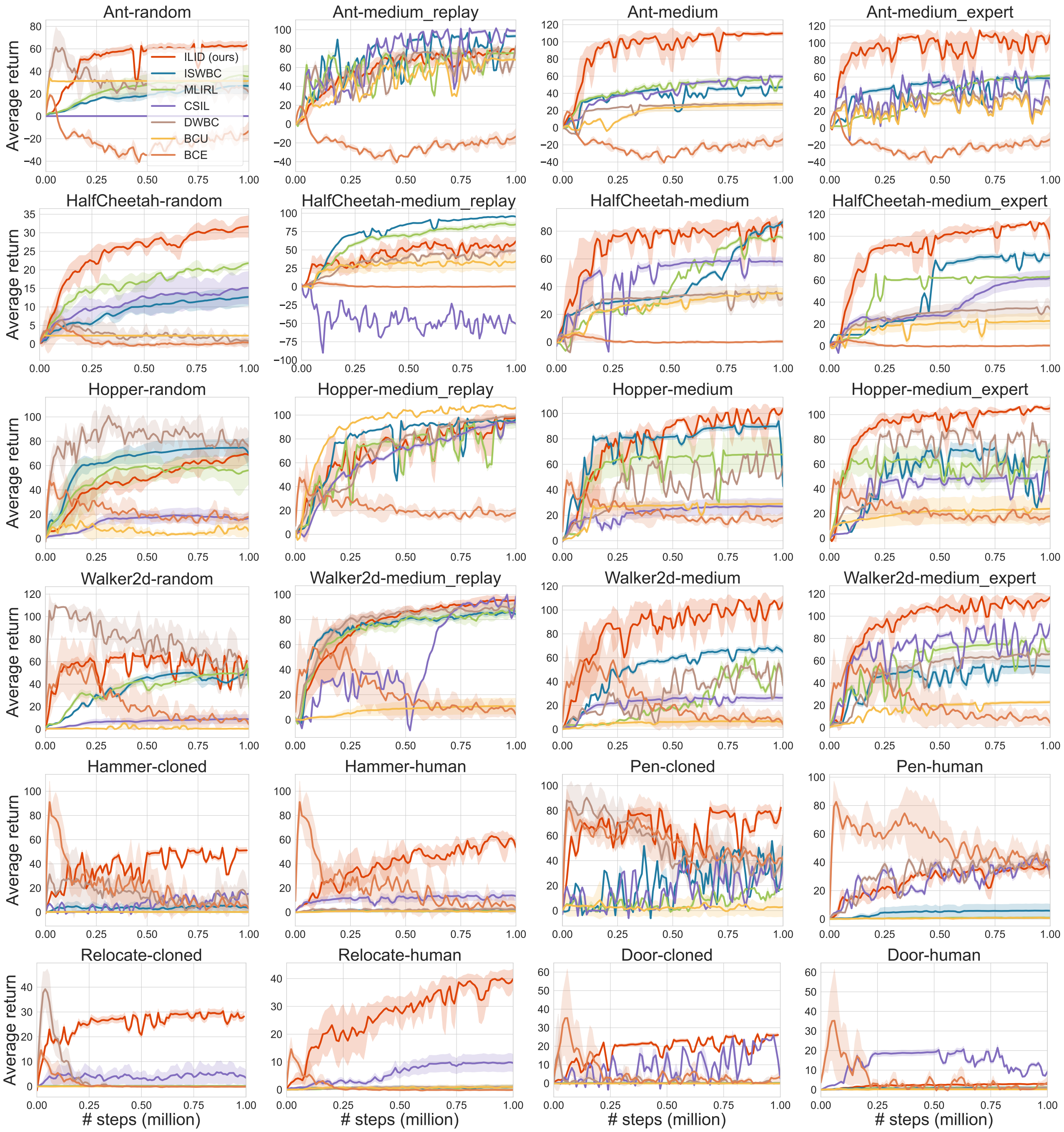


Figure 17. Normalized performance under varying qualities of imperfect data.

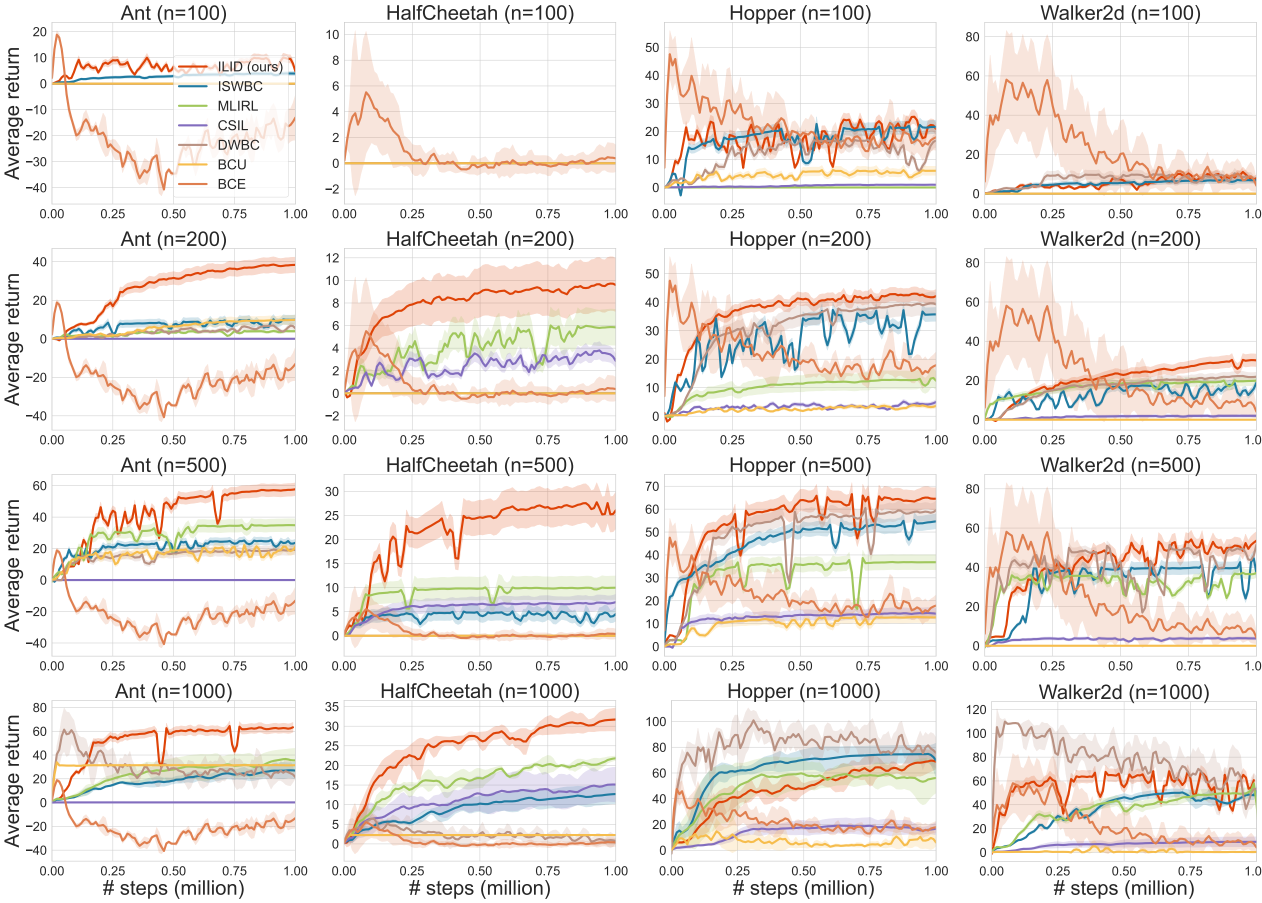


Figure 18. Effect of the quantity of imperfect demonstrations

B.4. Rollback Steps

Regarding the fourth question, we vary K from 1 to 100 and run experiments across all benchmarks. Fig. 19 clearly indicate that as K increases, there is an initial improvement in performance; once it reaches a sufficiently large value, performance tends to stabilize. Considering that a larger rollback step leads to more selected behaviors capable of reaching expert states, this observation offers support for Hypothesis 4.1. Importantly, the performance proves to be robust to a relatively large K, rendering `ILID` forgiving to the hyperparameter.

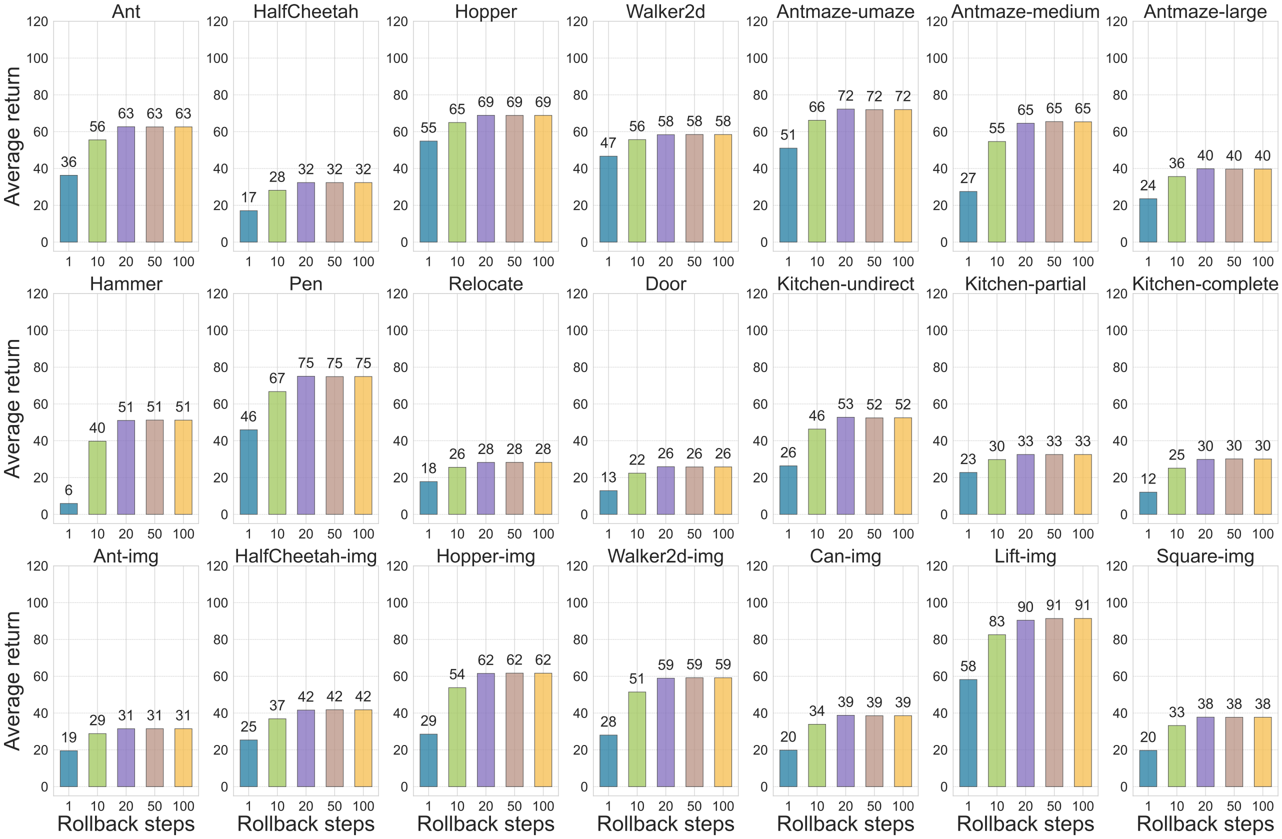


Figure 19. Performance of `ILID` under varying numbers of rollback steps

B.5. Runtime

We evaluate the runtime of `ILID` compared with baseline algorithms for 250,000 training steps, utilizing the same network size and batch size on an NVIDIA 4090 GPU. As illustrated by Fig. 8(a), the runtime of `ILID` (around 40 min) is slightly longer than `BC` (around 30 min), which substantiates the low computational cost of `ILID`.

B.6. Ablation Studies

In this section, we assess the effect of key components by ablating them, under the same setting as that of Table 5.

B.6.1. Only Importance-Sampling Weighting

Without the second term in Problem (13), ILID reduces to ISWBC. Unsurprisingly, as shown in Figs. 20 and 21, ISWBC does not suffice satisfactory performance.

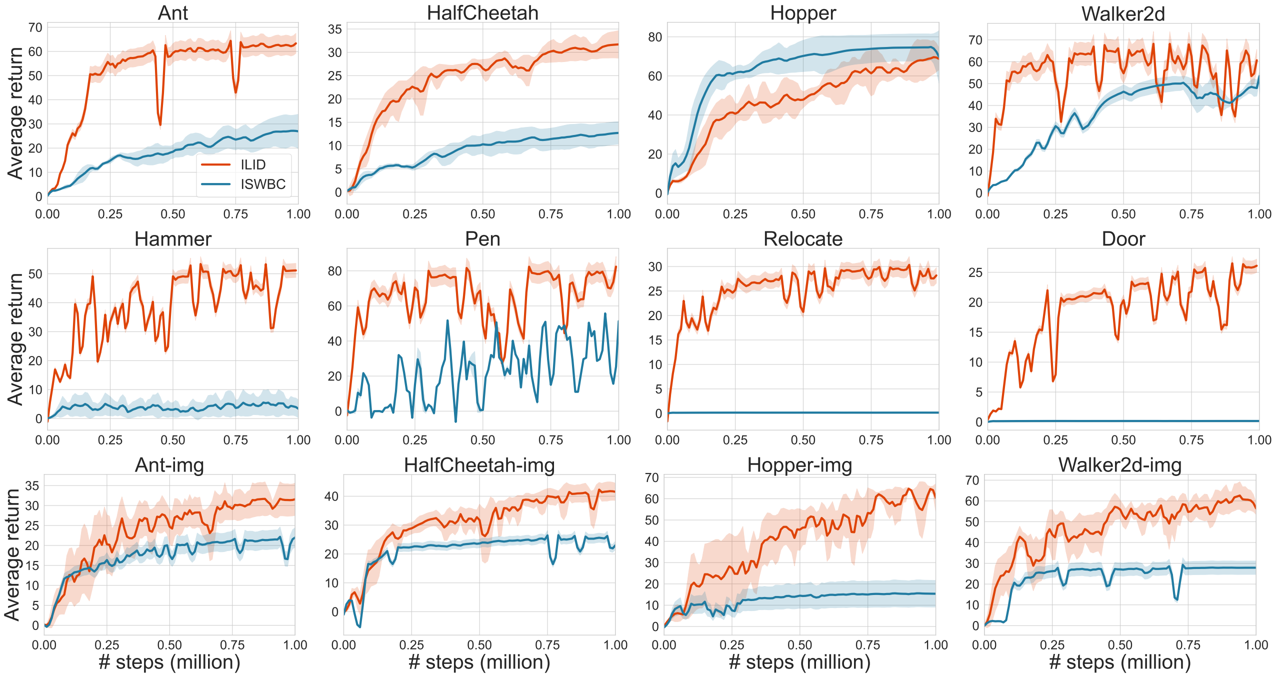


Figure 20. Comparison between ISWBC and ILID

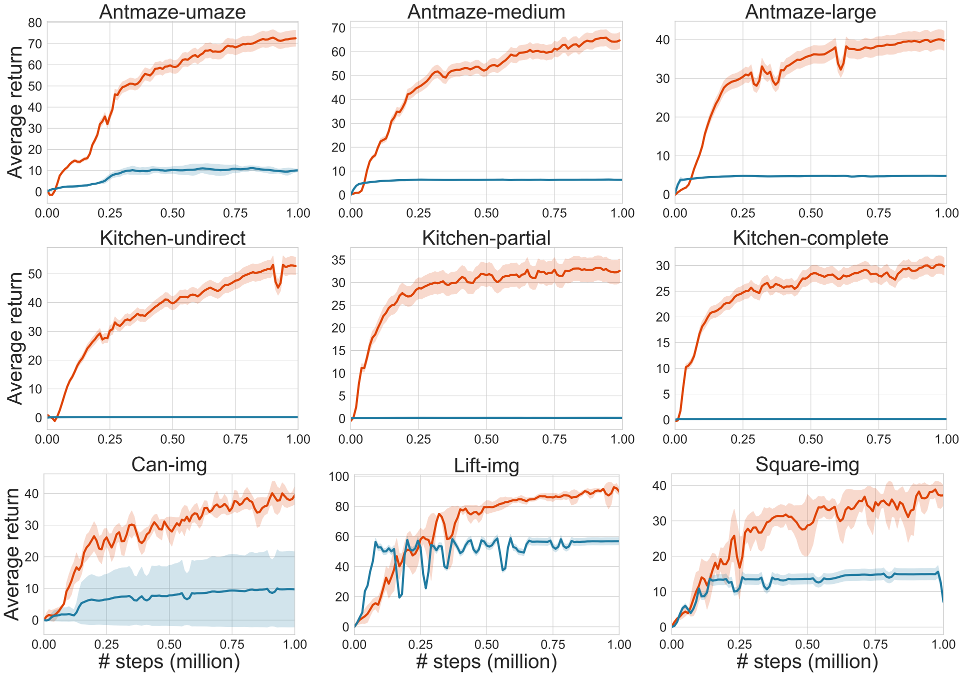


Figure 21. Comparison between `ISWBC` and `ILID`.

B.6.2. Importance of Data Selection

In this section, we ablate the data selection and replace $\mathcal{D}_s$ in Problem (13) by entire imperfect data of $\mathcal{D}_b$. Figs. 22 and 23 corroborate Hypothesis 4.1 and underscores the importance of our data selection scheme.

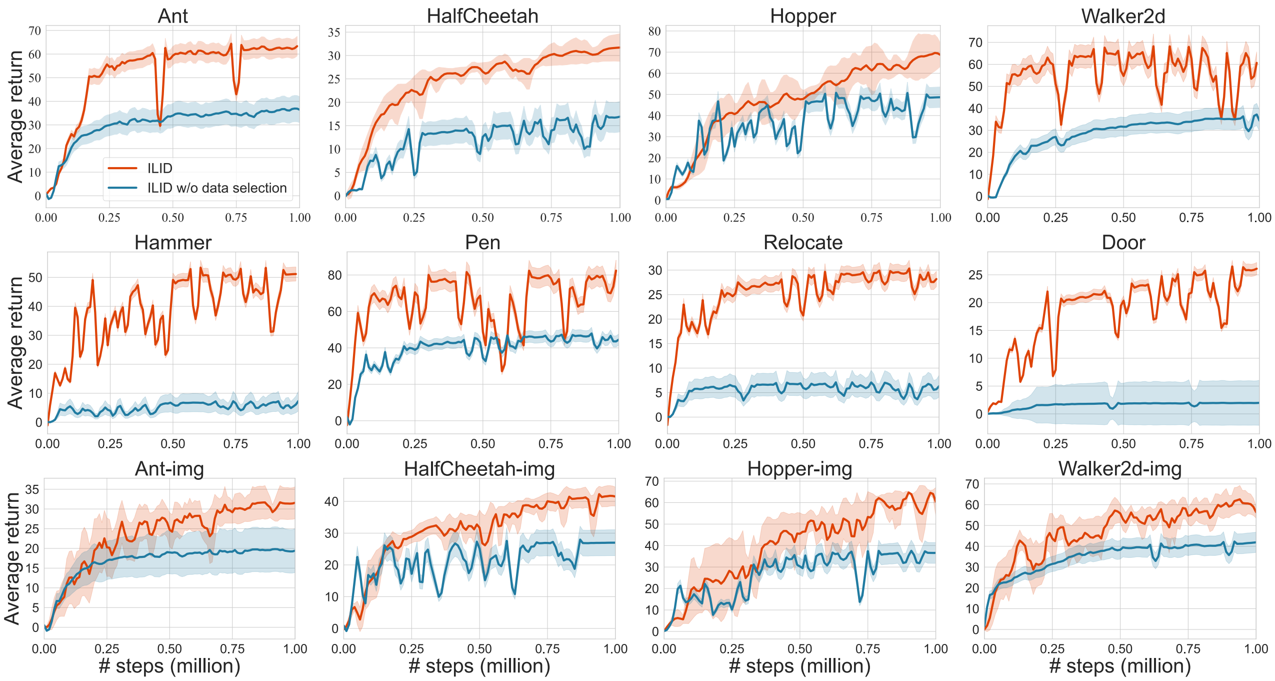


Figure 22. Effect of data selection.

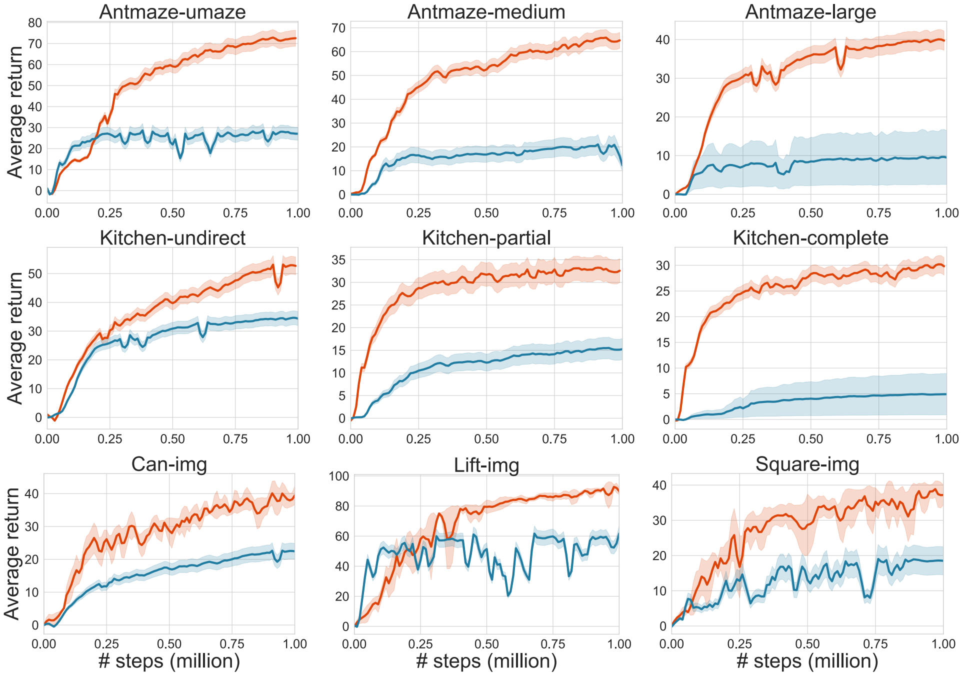


Figure 23. Effect of data selection.

B.6.3. Effect of $\alpha(s, a)$ and $\beta(s, a)$

In this section, we carry out ablation studies on $\alpha(s, a)$ and $\beta(s, a)$. The observed performance degradation in Figs. 24 and 25 clearly demonstrates the benefits of $\alpha(s, a)$ and $\beta(s, a)$. The importance-sampling weights can enhance the expert data support for BC, particularly in continuous domains. The absence of $\beta(s, a)$ renders the training becomes ineffective and unstable, due to behavior interference.

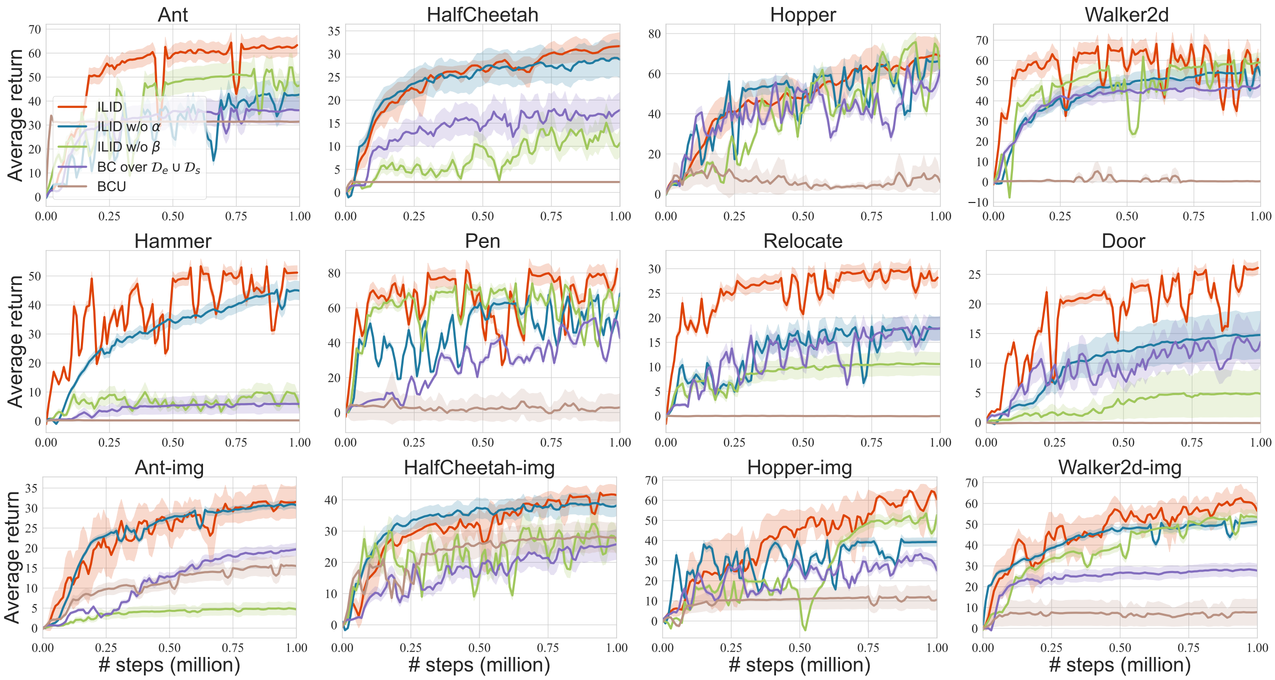


Figure 24. Importance of $\alpha(s, a)$ and $\beta(s, a)$. 'ILID w/o α' refers to change the first term in Problem (13) to BC. 'ILID w/o $\beta(s, a)$' refers to setting $\beta(s, a) \equiv 1$. 'BC over $\mathcal{D}_e \cup \mathcal{D}_s$' refers to running BC on the union of $\mathcal{D}_e$ and $\mathcal{D}_s$.

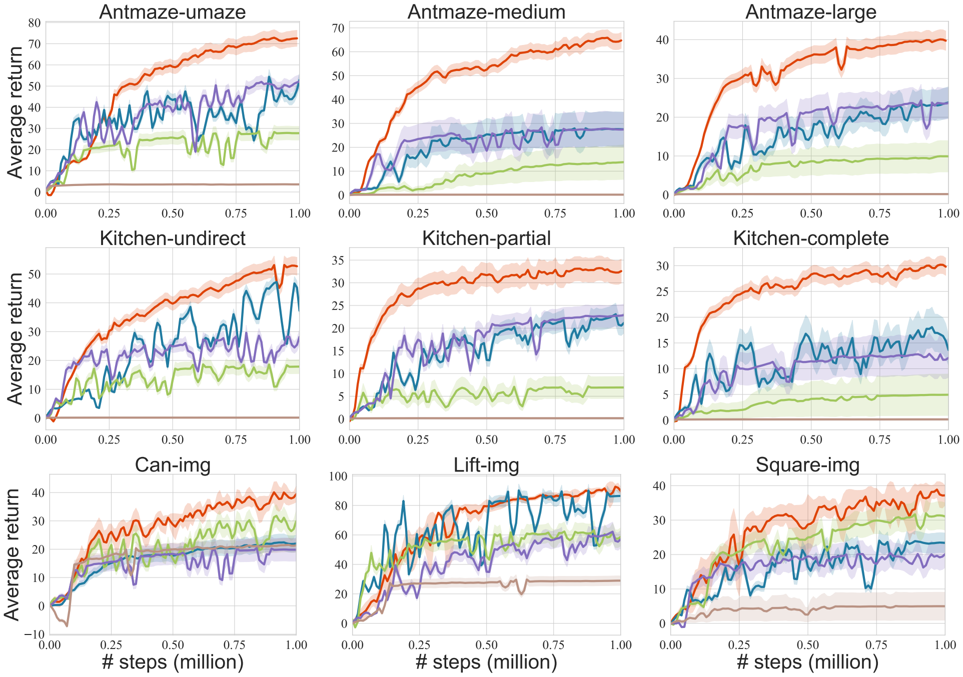


Figure 25. Importance of $\alpha(s,a)$ and $\beta(s,a)$. ‘ILID w/o α’ refers to change the first term in Problem (13) to BC. ‘ILID w/o $\beta(s,a)$’ refers to setting $\beta(s,a) \equiv 1$ in ILID. ‘BC over $\mathcal{D}_e \cup \mathcal{D}_s$’ refers to running BC on the union of $\mathcal{D}_e$ and $\mathcal{D}_s$.

C. Detailed Proofs

C.1. Proof of Theorem 4.2

In this section, we provide the proof details for Theorem 4.2, We use $(s, a, \ldots, (s'), (a')) \in \mathcal{D}$ to denote that dataset $\mathcal{D}$ contains sub-trajectory $(s, a, \ldots, (s'), (a'))$. When clear from the context, we omit the subscript and use $\mathbb{E}[\cdot]$ instead of $\mathbb{E}_{\mathcal{D}_e, \mathcal{D}_s}[\cdot]$ for conciseness.

First, recalling the definition of V^π in Section 3, we can write

$$\begin{aligned}
& V^{\pi_e} - V^{\tilde{\pi}} \\
&= \mathbb{E}_{s\sim\mu}\big[V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big] \qquad \text{(where } V^\pi(s) = \mathbb{E}_\pi[\textstyle\sum_{h=1}^H R(s_h, a_h) \mid s_1 = s]) \\
&= \mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big] + \mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_e)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big] \\
&= \mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big] \\
& \qquad \text{(due to determinism of expert policy and transition dynamics, detailed below)} \\
&= \underbrace{\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_s)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big]}_{(a)} \\
& \quad + \underbrace{\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big]}_{(b)}. \qquad (17)
\end{aligned}$$

More specifically, the third equality holds because: the trajectories, started with the visited initial states, are fully covered in the expert demonstrations; and deterministic dynamics enables $\tilde{\pi}$ to fully recover the expert trajectories.

Note that once the policy enters the states out of training distribution, it may keep making mistakes and remain out-of-distribution for the remainder of the time steps. Hence, we can bound term $\mathbb{E}[(a)]$ as follows:

$$\begin{aligned}
\mathbb{E}[(a)] &= \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_s)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big]\Big] \\
&\leq H\mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_s))\Big]\Big] \qquad \text{(due to } V^\pi(s) \leq H) \\
&= H\mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e) \cup \mathcal{S}_1(\mathcal{D}_s))\Big]\Big] \\
&= H\epsilon_o \qquad (18)
\end{aligned}$$

where $\epsilon_o = \mathbb{E}[\mathbb{E}_{s_1\sim\mu}[\mathbb{1}(s_1 \notin \mathcal{S}_1(\mathcal{D}_e) \cup \mathcal{S}_1(\mathcal{D}_s))]]$ is the missing mass defined in Theorem 4.2.

Regarding term (b), we can write

$$\begin{aligned}
\mathbb{E}[(b)] &= \mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big] \\
&= \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot V^{\pi_e}(s)\Big]\Big] \\
&\quad - \underbrace{\mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot V^{\tilde{\pi}}(s)\Big]\Big]}_{(c)}. \qquad (19)
\end{aligned}$$

For the second term in the last equality of Eq. (19), we have

$$\begin{aligned}
(c) &= \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot V^{\tilde{\pi}}(s)\Big]\Big] \\
&= \mathbb{E}\left[\mathbb{E}_{s\sim\mu}\left[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \mathbb{E}_{\tilde{\pi}}\left[\sum_{h=1}^H R(s_h, a_h) \mid s_1 = s\right]\right]\right] \\
& \qquad \text{(using the definition of } V^{\tilde{\pi}}(s) \text{ where } s_{h+1} = T(s_h, a_h) \text{ and } a_h \sim \tilde{\pi}(\cdot|s_h))
\end{aligned}$$

$$\geq \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \mathbb{E}_{\tilde{\pi}}\Big[\sum_{h=2}^{H} R(s_h, a_h) \mid s_1 = s\Big]\Big]\Big]$$

(omitting $R(s_1, a_1)$ and using $R(s_1, a_1) \geq 0$)

$$= \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \mathbb{E}_{a\sim\tilde{\pi}(\cdot|s), s'\sim T(s,a)}\big[V'(s')\big]\Big]\Big]$$

(denoting $V'(s') \doteq \sum_{h=2}^{H} r(s_h, a_h)$ where $s_2 = s'$, $s_{h+1} = T(s_h, a_h)$ and $a_h \sim \tilde{\pi}(\cdot|s_h)$)

$$= \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \mathbb{E}_{s'\sim\mathcal{D}_s(\cdot|s)}\big[V'(s')\big]\Big]\Big] \tag{20}$$

where $\mathcal{D}_s(s'|s) \doteq \sum_a n((s, a, s') \in \mathcal{D}_s)/n(s \in \mathcal{S}_1(\mathcal{D}_s))$, and the last equality is obtained by

$$\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \mathbb{E}_{a\sim\tilde{\pi}(\cdot|s), s'\sim T(s,a)}\big[V'(s')\big]$$

$$= \mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \sum_a \frac{n((s,a) \in \mathcal{D}_s)}{n(s \in \mathcal{S}_1(\mathcal{D}_s))} V'(T(s,a))$$

(from the definition of $\tilde{\pi}$ in Eq. (5) and the determinism of T)

$$= \mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \sum_a \frac{n((s,a,T(s,a)) \in \mathcal{D}_s)}{n(s \in \mathcal{S}_1(\mathcal{D}_s))} V'(T(s,a))$$

(due to $n((s,a) \in \mathcal{D}_s) = n((s,a,T(s,a)) \in \mathcal{D}_s)$)

$$= \mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \sum_{s',a} \frac{n((s,a,s') \in \mathcal{D}_s)}{n(s \in \mathcal{S}_1(\mathcal{D}_s))} V'(s')$$

(due to the fact that $n((s,a,s') \in \mathcal{D}_s) = 0$ if $s' \neq T(s,a)$)

$$= \mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \mathbb{E}_{s'\sim\mathcal{D}_s(\cdot|s)}\big[V'(s')\big]. \tag{21}$$

Substituting Eq. (20) to Eq. (19) yields

$$\mathbb{E}[(b)] \leq \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s)) \cdot \mathbb{E}_{s'\sim\mathcal{D}_s(\cdot|s)}\big[V^{\pi_e}(s) - V'(s')\big]\Big]\Big]$$

$$\leq (\delta+1)\mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s))\Big]\Big] \tag{22}$$

$$= (\delta+1)\mathbb{E}_{s\sim\mu}\Big[\mathbb{E}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s))\Big]\Big]$$

$$= (\delta+1)\mathbb{E}_{s\sim\mu}\Big[\mathbb{E}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\big] \cdot \mathbb{E}\big[\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s))\big]\Big]$$

(from the independence of $\mathcal{D}_e$ and $\mathcal{D}_s$)

$$\leq (\delta+1)\sqrt{\mathbb{E}_{s\sim\mu}\big[\mathbb{E}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\big]^2\big] \cdot \mathbb{E}_{s\sim\mu}\big[\mathbb{E}\big[\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s))\big]^2\big]}$$

(from the Cauchy-Schwarz inequality $\mathbb{E}[XY] \leq \sqrt{\mathbb{E}[X^2]\mathbb{E}[Y^2]}$)

$$\leq (\delta+1)\sqrt{\mathbb{E}_{s\sim\mu}\big[\mathbb{E}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))^2\big]\big] \cdot \mathbb{E}_{s\sim\mu}\big[\mathbb{E}\big[\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s))^2\big]\big]}$$

(from the fact $\mathbb{E}[X]^2 \leq \mathbb{E}[X^2]$)

$$= (\delta+1)\sqrt{\mathbb{E}_{s\sim\mu}\big[\mathbb{E}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\big]\big] \cdot \mathbb{E}_{s\sim\mu}\big[\mathbb{E}\big[\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_s))\big]\big]}$$

(from the fact that $\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}))^2 = \mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}))$)

$$= (\delta+1)\sqrt{\mathbb{E}_{s\sim\mu}\big[\mathbb{E}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\big]\big] \cdot \big(1 - \mathbb{E}_{s\sim\mu}\big[\mathbb{E}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_s))\big]\big]\big)}$$

$$= (\delta+1)\sqrt{\epsilon_e(1-\epsilon_s)}. \tag{23}$$

Regarding Eq. (22), due to $s' \in \mathcal{S}_1(\mathcal{D}_e)$ (see the definition of $\mathcal{D}_s$) and the definition of $\tilde{\pi}$ (which takes expert actions at given expert states), the sub-trajectory started from s' induced by $\tilde{\pi}$ follows the corresponding expert trajectory in $\mathcal{D}_e$. Based on the definition of δ, Eq. (22) can be derived by

$$V^{\pi_e}(s) - V'(s') \leq V^{\pi_e}(s) - (V^{\pi_e}(s') - 1) \leq \delta + 1, \tag{24}$$

where we use $R(s,a) \leq 1$ and the definition of $V'(s')$ which sums up over just $H-1$ steps. Combining Eqs. (18) and (23), we have

$$V^{\pi_e} - \mathbb{E}[V^{\tilde{\pi}}] \leq H\epsilon_o + (\delta+1)\sqrt{\epsilon_e(1-\epsilon_s)}, \tag{25}$$

thereby yielding the result.

C.2. Proof of Corollary 4.3

Analogouly to Eqs. (17) and (18), $V^{\pi_e} - \mathbb{E}[V^{\tilde{\pi}}]$ is also bounded by

$$\begin{aligned}
&V^{\pi_e} - \mathbb{E}[V^{\tilde{\pi}}] \\
&= \mathbb{E}\big[\mathbb{E}_{s\sim\mu}\big[V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big]\big] \\
&= \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big]\Big] + \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \in \mathcal{S}_1(\mathcal{D}_e)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big]\Big] \\
&= \mathbb{E}\Big[\mathbb{E}_{s\sim\mu}\Big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e)) \cdot \big(V^{\pi_e}(s) - V^{\tilde{\pi}}(s)\big)\Big]\Big] \\
&\leq H\mathbb{E}\big[\mathbb{E}_{s\sim\mu}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\big]\big].
\end{aligned} \tag{26}$$

Invoking Xu et al. (2021, Theorem 2), we can write

$$\begin{aligned}
\mathbb{E}\big[\mathbb{E}_{s\sim\mu}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\big]\big] &= \mathbb{E}_{s\sim\mu}\big[\mathbb{E}\big[\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))\big]\big] \\
&= \sum\nolimits_s \mu(s)\,\Pr(\mathbb{1}(s \notin \mathcal{S}_1(\mathcal{D}_e))) \\
&= \mu(s)(1-\mu(s))^{n_e} \\
&\leq |\mathcal{S}| \max_{x\in[0,1]} x(1-x)^{n_e} \\
&\leq \frac{|\mathcal{S}|}{en_e},
\end{aligned} \tag{27}$$

where e is Euler's number, and the last inequality is obtained via solving the maximization. Specifically, denote $f(x) = x(1-x)^{n_e}$ and take its derivative to zero, yielding

$$f'(x) = (1-x)^{n_e-1}(1-(n_e+1)x) = 0 \quad \Rightarrow \quad x^* = \frac{1}{n_e+1}. \tag{28}$$

Therefore, the following holds:

$$\max_{x\in[0,1]} x(1-x)^{n_e} = \frac{1}{n_e+1}\left(1 - \frac{1}{n_e+1}\right)^{n_e} = \frac{1}{n_e}\left(1 - \frac{1}{n_e+1}\right)^{n_e+1} \leq \frac{1}{en_e}. \tag{29}$$

Substituting Eq. (27) in Eq. (26), we obtain

$$V^{\pi_e} - \mathbb{E}[V^{\tilde{\pi}}] \leq \frac{|\mathcal{S}|H}{en_e}. \tag{30}$$

Similarly, from Theorem 4.2, we have

$$V^{\pi_e} - \mathbb{E}[V^{\tilde{\pi}}] \leq H\epsilon_o + (\delta+1)\sqrt{\epsilon_e(1-\epsilon_s)} \leq \frac{|\mathcal{S}|H}{e(n_e+n_s)} + (\delta+1)\sqrt{\frac{|\mathcal{S}|}{en_e}}. \tag{31}$$

Combining Eqs. (30) and (31), we can write

$$V^{\pi_e} - \mathbb{E}[V^{\tilde{\pi}}] \leq \min\left\{\frac{|\mathcal{S}|H}{en_e}, \frac{|\mathcal{S}|H}{e(n_e+n_s)} + (\delta+1)\sqrt{\frac{|\mathcal{S}|}{en_e}}\right\} \tag{32}$$

Taking n_s to infinity, $V^{\pi_e} - \mathbb{E}[V^{\tilde{\pi}}] \leq \min\{(\delta+1)\sqrt{|\mathcal{S}|/(en_e)}, |\mathcal{S}|H/(en_e)\}$. Thus, with a sufficiently large n_s, to obtain an ε-optimal policy, $\tilde{\pi}$ requires at most $\mathcal{O}(\min\{|\mathcal{S}|/\varepsilon^2, |\mathcal{S}|H/\varepsilon\})$ expert trajectories.