Gradients can train reward models: An Empirical Risk Minimization Approach for Offline Inverse RL and Dynamic Discrete Choice Model

ENOCH H. KANG*, HEMA YOGANARASIMHAN, AND LALIT JAIN

We study the problem of estimating Dynamic Discrete Choice (DDC) models, also known as offline Maximum Entropy-Regularized Inverse Reinforcement Learning (offline MaxEnt-IRL) in machine learning. The objective is to recover reward or Q functions that govern agent behavior from offline behavior data. In this paper, we propose a globally convergent gradient-based method for solving these problems without the restrictive assumption of linearly parameterized rewards. The novelty of our approach lies in introducing the Empirical Risk Minimization (ERM) based IRL/DDC framework, which circumvents the need for explicit state transition probability estimation in the Bellman equation. Furthermore, our method is compatible with non-parametric estimation techniques such as neural networks. Therefore, the proposed method has the potential to be scaled to high-dimensional, infinite state spaces. A key theoretical insight underlying our approach is that the Bellman residual satisfies the Polyak-Łojasiewicz (PL) condition – a property that, while weaker than strong convexity, is sufficient to ensure fast global convergence guarantees. Through a series of synthetic experiments, we demonstrate that our approach consistently outperforms benchmark methods and state-of-the-art alternatives.

CONTENTS

[1] Entropy regularized IRL's equivalence with dynamic discrete choice (DDC) model is discussed in Appendix C.5.

Manuscript submitted for review to the 26th ACM Conference on Economics & Computation (EC'25).

1 Introduction

Learning from previously collected datasets has become an essential paradigm in sequential decision-making problems where exploration during interactions with the environment is infeasible (e.g., self-driving cars, medical applications) or leveraging large-scale offline data is preferable (e.g., social science, recommendation systems, and industrial automation) [Levine et al., 2020]. However, in such cases, defining a reward function (a flow utility function) that accurately captures the underlying decision-making process is often challenging due to the unobservable/sparse rewards [Zolna et al., 2020] and complexity of real-world environments [Foster et al., 2021]. To circumvent these limitations, learning from expert demonstrations has gained prominence, motivating approaches such as Imitation Learning (IL) and offline Inverse Reinforcement Learning (offline IRL) or equivalently, Dynamic Discrete Choice (DDC) model estimation[2].

While IL directly learns a policy by mimicking expert actions, it is susceptible to *distribution shift*, i.e., when the testing environment (reward, transition function) is different from the training environment. On the other hand, offline IRL aims to infer the underlying reward function that best explains expert behavior. Given this reward function, a new policy can be trained after a change in the environment's transition dynamics (e.g., modifications in recommendation systems) or in the reward function (e.g., marketing interventions). This capability enables offline IRL to be employed in counterfactual simulations, such as evaluating the effects of different policy decisions without direct experimentation. However, an imprecise reward function can lead to suboptimal policy learning and unreliable counterfactual analyses, ultimately undermining its practical utility. As a result, offline IRL's key metric becomes the *precision* of reward inference.

While the precise reward function estimation objective has been studied in recent offline IRL literature, theoretically guaranteed existing methods have been limited to explicitly learning a transition model (e.g., Zeng et al. [2023]). However, if relearning the transition function is required every time it changes, the premise of IRL for counterfactual simulations may be undermined. The Dynamic Discrete Choice (DDC) literature in econometrics has separately explored the problem towards the goal of precise reward estimation [Aguirregabiria and Mira, 2007, Hotz and Miller, 1993, Rust, 1994, Su and Judd, 2012]. However, existing methodologies with theoretical precision guarantees suffer from the curse of dimensionality [Geng et al., 2023]: computational complexity exponentially grows as state dimension increases. Most importantly, in both IRL and DDC literature, theoretical guarantees of precise reward estimation have been limited to linear reward structures [Zeng et al., 2023] or monotone value function structure [Feng et al., 2020]. This motivates us to ask the following question:

Can we propose a scalable gradient-based method to infer rewards (or Q^* function) while provably ensuring global optimality with no assumption on reward structure/transition function knowledge?

Our contributions. In this paper, we propose an Empirical Risk Minimization (ERM)–based gradient-based method for IRL/DDC as an inverse Q-learning method. This method provably finds the true Q^* function (up to statistical error, which diminishes at an $O(1/\sqrt{N})$ rate with N samples) with $O(1/T)$ rate of convergence, where T is the number of gradient iterations. In addition, the true reward function can be computed from estimated Q^* with no extra statistical or computational cost given the estimated Q^* function. In developing this method, we make the following technical contributions:

- We propose an empirical risk minimization (ERM) problem formulation, which we refer to as ERM-IRL in the IRL literature and ERM-DDC in the DDC literature, reflecting the shared problem. This formulation allows us to circumvent the need for explicit transition function estimation.

[2]Refer to Section C.5 for the equivalence between Offline Maximum Entropy IRL (MaxEnt-IRL) and DDC.

- We show that the objective function of the ERM-IRL satisfies the Polyak-Łojasiewicz (PL) condition, which is a weaker but equally useful alternative to strong convexity for providing theoretical convergence guarantees. This is enabled by showing that each of its two components – expected negative log-likelihood and mean squared Bellman error – satisfy PL condition[3].
- Since the mean squared Bellman error term is a solution to a strongly concave inner maximization problem [Dai et al., 2018, Patterson et al., 2022], minimization of the ERM-IRL objective becomes a mini-max problem with two-sided PL condition [Yang et al., 2020]. Using this idea, we propose an alternating gradient ascent-decent algorithm that provably converges to the true Q^*, which is the unique saddle point of the problem.

In addition to establishing theoretical global convergence guarantees, we demonstrate the empirical effectiveness of the algorithm through standard benchmark simulation experiments. Specifically, we evaluate using a series of simulations: (1) The Rust bus engine replacement problem [Rust, 1987], which is the standard framework for evaluation used in the dynamic discrete choice literature, and (2) A high-dimensional variant of the Rust bus-engine problem, where we allow a very large state space. In both settings, we show that our algorithm outperforms/matches the performance of existing approaches. It is particularly valuable in large state-space settings, where many of the standard algorithms become infeasible due to their need to estimate state-transition probabilities. We expect our approach to be applicable to a variety of business and economic problems where the state and action space are infinitely large, and firms/policy-makers do not have *a priori* knowledge of the parametric form of the reward function and/or state transitions.

The remainder of the paper is organized as follows. In Section 2, we discuss related work in greater detail. Section 3 introduces the problem setup and provides the necessary background. In Section 4, we present the ERM-IRL framework, followed by an algorithm for solving it in Section 5. Section 6 establishes the global convergence guarantees of the proposed algorithm. Finally, Section 7 presents experimental results demonstrating the effectiveness of our approach.

2 Related works

The equivalence between Dynamic Discrete Choice (DDC) models and entropy-regularized Inverse Reinforcement Learning (MaxEnt-IRL) was first identified by Ermon et al. [2015]. We provide a detailed exposition and the proof of equivalence in Section C.

2.1 Dynamic discrete choice model estimation

In the econometrics literature, stochastic decision-making behaviors are usually considered to come from the random utility model [McFadden, 2001], which often assumes that the effect of unobserved covariates appear in the form of additive and conditionally independent randomness in agent utilities Rust [1994]. The seminal paper by Rust [Rust, 1987] pioneered this literature, demonstrating that a DDC model can be solved by solving a maximum likelihood estimation problem that runs above iterative dynamic programming. As discussed in the introduction, this method suffers computational intractability in terms of number of state dimensions.

Hotz and Miller [1993] introduced a method which is often called the two-step method conditional choice probability (CCP) method, where the CCPs and transition probabilities estimation step is followed by the reward estimation step. The reward estimation step avoids dynamic programming by combining simulation with the insight that differences in value function values can be directly inferred from data without solving Bellman equations. However, simulation methods are in principle trajectory-based numerical integration methods which also suffer scalability issues. Fortunately,

[3] Sum of two PL functions are is necessary PL; in the proof, we show that our case is an exception.

Method	Transition Estimation-Free	Anchor Action	Non-Parametric	Gradient-Based	Reward Estimation	Scalability	Globally Convergent
Rust [1987]		✓			✓		△(Linear only)
CCP [1993]		✓			✓		△(Linear only)
MPEC [2012]		✓			✓		△(Linear only)
BC [2018]	✓		✓	✓		✓	
AVI [2019]	✓	✓	✓		✓		
Semi-gradient 2019	✓	✓		✓	✓	✓	
IQ-Learn [2021]	✓		✓	✓		✓	
SAmQ [2023]	✓		✓		✓	△	
RP [2022]			✓		✓	✓	
Clare [2023]			✓	✓		✓	
ML-IRL [2023]			✓	✓	✓	✓	△(Linear only)
Ours	✓	✓	✓	✓	✓	✓	✓

Table 1. Comparison of IRL and DDC methods. 'Transition Estimation-Free" indicates whether the method avoids explicit transition function estimation. A method is 'Scalable" if it handles state spaces of at least 20^{10}. "Reward Estimation" excludes occupancy matching-based methods (e.g., IQ-Learn, Clare); see Appendix D.

we can sometimes avoid simulation altogether by utilizing the problem structure such as regenerative/terminal actions (known as finite dependence [Arcidiacono and Miller, 2011]). Still, this method requires explicit estimation of the transition function, which is not the case in our paper. This paper established an insight that there exists a one-to-one correspondence between the CCPs and the differences in Q^*-function values, which was formalized as the identification result by Magnac and Thesmar [2002].

Su and Judd [2012] propose that we can avoid dynamic programming or simulation by formulating a nested linear programming problem with Bellman equations as constraints of a linear program. This formulation is based on the observation that Bellman equations constitute a convex polyhedral constraint set. While this linear programming formulation significantly increases the computation speed, it is still not scalable in terms of state dimensions.

As the above methods suffer scalability issues, methods based on parametric/nonparametric approximation have been developed. Parametric policy iteration [Benitez-Silva et al., 2000] and sieve value function iteration [Arcidiacono et al., 2013] parametrize the value function by imposing a flexible functional form. Norets [2012] proposed that neural network-based function approximation reduces the computational burden of Markov Chain Monte Carlo (MCMC) estimation, thereby enhancing the efficiency and scalability. Geng et al. [2020] proposed that the one-to-one correspondence insight of Hotz and Miller [1993] enables us to avoid reward parameterization and directly (non-parametrically) estimate value functions, along with solving a much smaller number of soft-Bellman equations which do not require reward parametrization to solve them. Barzegary and Yoganarasimhan [2022] and Geng et al. [2023] independently proposed state aggregation/partition methods that significantly reduce the computational burden of running dynamic programming with the cost of optimality. While Geng et al. [2023] uses k-means clustering [Kodinariya et al., 2013, Sinaga and Yang, 2020], Barzegary and Yoganarasimhan [2022] uses recursive partitioning (RP) [Athey and Imbens, 2016]. As discussed earlier, combining approximation with dynamic programming induces unstable convergence except when the true reward function is linear Jiang and Xie [2024].

Adusumilli and Eckardt [2019] proposed how to adapt two popular temporal difference (TD)-based methods (an approximate dynamic programming-based method and a semi-gradient descent method

based on Tsitsiklis and Van Roy [1996]) for DDC. As discussed earlier, approximate dynamic programming-based methods are known to suffer from a lack of provable convergence beyond linear reward models [Jiang and Xie, 2024, Wang et al., 2021]; the semi-gradient method is a popular, efficient approximation method that has limited theoretical assurance of convergence beyond linear value function approximation [Sutton and Barto, 2018]. Feng et al. [2020] showed global concavity of value function under certain transition functions and monotonicity of value functions in terms of one-dimensional state, both of which are easily satisfied for applications in social science problems. However, those conditions are limitedly satisfied for the problems with larger dimensional state space.

2.2 Offline inverse reinforcement learning

In computer science literature, stochastic decision-making behaviors are modeled as 'random choice'. That is, they assume that agents randomize their actions. The most widely used inverse reinforcement learning model, Maximum-Entropy inverse reinforcement learning (MaxEnt-IRL), assumes that the random choice happens due to agents choosing the optimal policy after penalization of the policy by its Shannon entropy [Ermon et al., 2015]. In addition to the equivalence of MaxEnt-IRL to DDC (See Ermon et al. [2015], also in Section C.5), the identifiability condition for DDC [Magnac and Thesmar, 2002] was rediscovered by Cao et al. [2021] for MaxEnt-IRL. Zeng et al. [2023] proposes a two-step maximum likelihood-based method that can be considered as a conservative version of CCP method of Hotz and Miller [1993][4]. Despite that their method is proven to be convergent, its global convergence was limitedly proven only for linear reward functions.

Finn et al. [2016] and Fu et al. [2017] showed that a myopic[5] version of MaxEnt-IRL can be solved by the Generative Adversarial Network (GAN) training framework [Goodfellow et al., 2020]. This approach has been extended to Q-estimation methods that use fixed point iteration [Geng et al., 2020, 2023]. Ni et al. [2021] have shown that the idea of training an adversarial network can also be used to calculate the gradient direction for minimizing the myopic version of negative log likelihood[6]. As the adversarial formulation of IRL is inherently myopic, it is limited suited for the task of reward inference. However, it is known to work well for behavioral cloning tasks [Torabi et al., 2018].

A family of methods starting from Ho and Ermon [2016] tries to address the inverse reinforcement learning problem from the perspective of occupancy matching, i.e., finding a policy that best matches the behavior of data. Garg et al. [2021] proposed how to extend the occupancy matching approach of Ho and Ermon [2016] to directly estimate Q-function instead of r. Given the assumption that the Bellman equation holds, this approach allows a simple gradient-based solution, as the occupancy matching objective function they maximize becomes concave. Yue et al. [2023] modifies Ho and Ermon [2016] to conservatively deal with the uncertainty of transition function. Despite their simplicity, one caveat of occupancy matching approaches is that whether the estimated Q from occupancy matching satisfies the Bellman equation is not trivial. In addition, computing r from Q using the Bellman equation is also not valid.

[4]When there is no uncertainty in the transition function, approximated trajectory gradient of Offline IRL method degenerates to forward simulation-based gradient in CCP estimator method of Hotz and Miller [1993].

[5]See Cao et al. [2021] for more discussion on this.

[6]Minimizing negative log-likelihood is equivalent to minimizing KL divergence. See the Proof of Lemma B.2.

3 Problem set-up and backgrounds

3.1 Offline Inverse Reinforcement Learning (Offline IRL) problem[7]

Consider a single-agent Markov Decision Process (MDP) defined as a tuple $(\mathcal{S}, \mathcal{A}, P, \nu_0, r, \beta)$ where $\mathcal{S}$ denotes the state space and $\mathcal{A}$ denotes a finite action space, $P \in \Delta_{\mathcal{S}}^{\mathcal{S}\times\mathcal{A}}$ is a Markovian transition kernel, $\nu_0 \in \Delta_{\mathcal{S}}$ is the initial state distribution over $\mathcal{S}$, $r \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is a reward function and $\beta \in (0,1)$ a discount factor. Given a stationary Markov policy $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$, an agent starts from initial state s_0 and takes an action $a_h \in \mathcal{A}$ at state $s_h \in \mathcal{S}$ according to $a_h \sim \pi(\cdot \mid s_h)$ at each period h. Given an initial state s_0, we define the distribution of state-action sequences for policy π over the sample space $(\mathcal{S}\times\mathcal{A})^\infty = \{(s_0, a_0, s_1, a_1, \ldots) : s_h \in \mathcal{S}, a_h \in \mathcal{A}, h \in \mathbb{N}\}$ as $\mathbb{P}^\pi$. We also use $\mathbb{E}_\pi$ to denote the expectation with respect to $\mathbb{P}^\pi$. Following existing literature [Fu et al., 2017, Geng et al., 2020, Ho and Ermon, 2016], we consider the *entropy-regularized* optimal policy, which is defined as

$$\pi^* := \text{argmax}_{\pi\in\Delta_{\mathcal{A}}^{\mathcal{S}}} \mathbb{E}_\pi\Big[\sum_{h=0}^{\infty} \beta^h \big(r(s_h, a_h) + \lambda\mathcal{H}(\pi(\cdot \mid s_h))\big)\Big]$$

where $\mathcal{H}$ denotes the Shannon entropy and λ is the regularization coefficient. Throughout, we make the following assumption on agents' decisions.

ASSUMPTION 3.1. *When interacting with the MDP $(\mathcal{S}, \mathcal{A}, P, \nu_0, r, \beta)$, each agent follows the entropy-regularized optimal stationary policy π^*.*

Throughout the paper, we use $\lambda = 1$, the setting which is equivalent to dynamic discrete choice (DDC) model with mean zero T1EV distribution (Appendix C.5); all the results of this paper easily generalize to other values of λ. Given π^*, we define the *value function* V^* as:

$$V^*(s) := \mathbb{E}_{\pi^*}\left[\sum_{h=0}^{\infty} \beta^h \big(r(s_h, a_h) + \mathcal{H}(\pi^*(\cdot \mid s_h))\big) \;\middle|\; s_0 = s\right].$$

Similarly, we define the Q^* function as follows:

$$Q^*(s,a) := r(s,a) + \beta \cdot \mathbb{E}_{s'\sim P(s,a)}\left[V^*(s') \mid s, a\right]$$

Given state s and policy π^*, let $\mathbf{q} = [q_1 \ldots q_{|\mathcal{A}|}]$ denote the probability distribution over the action space $\mathcal{A}$, such that:

$$q_a = \frac{\exp(Q^*(s,a))}{\sum_{a'\in\mathcal{A}} \exp(Q^*(s,a'))} \text{ for } a \in \mathcal{A} \tag{1}$$

Then, according to Assumption 3.1, the value function V^* must satisfy the recursive relationship defined by the *Bellman equation* as follows:

$$V^*(s) = \max_{\mathbf{q}\in\Delta_{\mathcal{A}}} \{\mathbb{E}_{a\sim\mathbf{q}}\big[r(s,a) + \beta\mathbb{E}_{s'\sim P(s,a)}[V^*(s') \mid s, a]\big] + \mathcal{H}(\mathbf{q})\}.$$

Further, we can show that (see Appendix C.3):

$$V^*(s) = \ln\left[\sum_{a\in\mathcal{A}} \exp(Q^*(s,a))\right], \quad \pi^*(a \mid s) = \frac{\exp(Q^*(s,a))}{\sum_{a'\in\mathcal{A}} \exp(Q^*(s,a'))} \text{ for } a \in \mathcal{A}$$

Throughout, we define a function V_Q as

$$V_Q(s) := \ln\left[\sum_{a\in\mathcal{A}} \exp(Q(s,a))\right]$$

[7] Entropy regularized IRL's equivalence with dynamic discrete choice (DDC) model is discussed in Appendix C.5.

where $V_{Q^*} = V^*$. Further, we note that the Bellman equation can be stated in terms of Q^* function:

$$\begin{aligned} Q^*(s,a) &= r(s,a) + \beta \cdot \mathbb{E}_{s' \sim P(s,a)}\left[V^*(s') \mid s, a\right] \\ &= r(s,a) + \beta \cdot \mathbb{E}_{s' \sim P(s,a), a' \sim \pi^*(\cdot|s')}\left[Q^*(s',a') - \log \pi^*(a' \mid s') \mid s, a\right] \\ &= r(s,a) + \beta \cdot \mathbb{E}_{s' \sim P(s,a)}\left[\log \sum_{a' \in \mathcal{A}} \exp(Q^*(s',a')) \mid s, a\right] \end{aligned} \tag{2}$$

Suppose that we are given the state-action-next state sequences of offline data on an agent's behaviors we denote as $\mathcal{D} := \left((s_0, a_0, s_0'), (s_1, a_1, s_1'), \ldots, (s_N, a_N, s_N')\right)$. Following Assumption 3.1, we assume that the data was generated by the agent playing the optimal policy π^* when interacting with the MDP $(\mathcal{S}, \mathcal{A}, P, \nu_0, r, \beta)$.

DEFINITION 3.1 (OFFLINE INVERSE REINFORCEMENT LEARNING PROBLEM). *The objective of offline IRL can be defined as learning a function $\hat{r} \in \mathcal{R} \subseteq \mathbb{R}^{\bar{\mathcal{S}} \times \mathcal{A}}$ that minimizes the mean squared prediction error with respect to data distribution[8] from offline data $\mathcal{D}$ such that:*

$$\underset{\hat{r} \in \mathcal{R}}{\operatorname{argmin}}\, \mathbb{E}_{(s,a) \sim \nu_0, \pi^*}\left[(\hat{r}(s,a) - r(s,a))^2\right] \tag{3}$$

where

$$\overline{\mathcal{S}} := \{s \in \mathcal{S} \mid \Pr\left(s_h = s \mid s_0 \sim \nu_0, \pi^*\right) > 0 \text{ for some } h \geq 0\}$$

defines the expert policy's coverage[9].

3.2 Identification

As we defined in Definition 3.1, our goal is to learn the agent's reward function $r(s,a)$ given offline data $\mathcal{D}$. However, without additional assumptions on the reward structure, this problem is ill-defined because many reward functions can explain the optimal policy [Fu et al., 2017, Ng et al., 1999]. To address this issue, following the DDC literature [Hotz and Miller, 1993, Magnac and Thesmar, 2002, Rust, 1994] and recent IRL literature [Geng et al., 2020], we assume that there is an *anchor action* a_s in each state s, such that the reward for each of state - anchor action combinations are known.

ASSUMPTION 3.2. *For all $s \in \mathcal{S}$, there exists an action $a_s \in \mathcal{A}$ such that $r(s, a_s)$ is known.*

Note that the optimal policy remains the same irrespective of the choice of the anchor action a_s and the reward value at the anchor action $r(s, a_s)$ (at any given s). As such, Assumption 3.2 only helps with identification and does not materially affect the estimation procedure. That is, we can arbitrarily choose a_s and arbitrarily set $r(s, a_s)$ for all $s \in \mathcal{S}$. In Theorem 3.1, we formally establish that Assumptions 3.1 and 3.2 uniquely identify Q^* and r. (See Section B.2 for the proof.)

THEOREM 3.1 ([MAGNAC AND THESMAR, 2002]). *Given discount factor β, transition kernel $P \in \Delta_{\mathcal{S}}^{\mathcal{S} \times \mathcal{A}}$ and optimal policy $\pi^* \in \Delta_{\mathcal{A}}^{\mathcal{S}}$, under Assumptions 3.1 and 3.2, the solution to the following system of equations:*

$$\begin{cases} \dfrac{\exp(Q(s,a))}{\sum_{a' \in \mathcal{A}} \exp(Q(s,a'))} = \pi^*(a \mid s) \quad \forall s \in \mathcal{S}, a \in \mathcal{A} \\ r(s,a_s) + \beta \cdot \mathbb{E}_{s' \sim P(s,a_s)}\left[V_Q(s') \mid s, a_s\right] - Q(s,a_s) = 0 \quad \forall s \in \mathcal{S} \end{cases}$$

gives a unique $Q = Q^*$. Furthermore, r is obtained by solving:

$$r(s,a) = Q^*(s,a) - \beta \cdot \mathbb{E}_{s' \sim P(s,a)}\left[V_{Q^*}(s') \mid s, a\right]. \tag{4}$$

for all $s \in \mathcal{S}, a \in \mathcal{A}$.

[8] This choice of distribution for computing MSE is natural under Assumption 3.1.

[9] Under Assumption 3.1, all actions of $s \in \bar{\mathcal{S}}$ is covered, i.e., occurs with probability strictly larger than 0.

In the first part of the theorem, we show that, after constraining the reward functions for anchor actions, we can recover the unique Q^*-function for the optimal policy from the observed choices and the Bellman equation for the anchor-action (written in terms of log-sum-exp of Q-values). The second step follows naturally, where we can show that reward functions are then uniquely recovered from Q^*-functions using the Bellman equation.

3.3 Bellman error and Temporal difference (TD) error

There are two key concepts used for describing a gradient-based algorithm for IRL/DDC: the Bellman error and the Temporal difference (TD) error. In this section, we define each of them and discuss their relationship. Let us start from defining $\mathcal{Q} = \{Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \mid \|Q\|_\infty < \infty\}$. By Rust [1994], $\beta < 1$ implies $Q^* \in \mathcal{Q}$. Next, we define the *Bellman operator* as $\mathcal{T} : \mathcal{Q} \mapsto \mathcal{Q}$ as follows:

$$\mathcal{T}Q(s,a) := r(s,a) + \beta \cdot \mathbb{E}_{s' \sim P(s,a)}\left[V_Q(s')\right]$$

According to the Bellman equation shown in Equation (2), Q^* satisfies $\mathcal{T}Q^*(s,a) - Q^*(s,a) = 0$; in fact, Q^* is the unique solution to $\mathcal{T}Q(s,a) - Q(s,a) = 0$; see [Rust, 1994]. Based on this observation, we define the following notions of error.

DEFINITION 3.2. *We define the Bellman error for $Q \in \mathcal{Q}$ at (s,a) as $\mathcal{T}Q(s,a) - Q(s,a)$. Furthermore, we define the Squared Bellman error and the Expected squared Bellman error as*

$$\mathcal{L}_{BE}(Q)(s,a) := (\mathcal{T}Q(s,a) - Q(s,a))^2$$

$$\overline{\mathcal{L}_{BE}}(Q) = \mathbb{E}_{(s,a)\sim\pi^*,\, \nu_0}\left[\mathcal{L}_{BE}(Q)(s,a)\right]$$

In practice, we don't have direct access to $\mathcal{T}$ unless we know (or have a consistent estimate of) the transition kernel $P \in \Delta_{\mathcal{S}}^{\mathcal{S}\times\mathcal{A}}$. Instead, we can compute an empirical *Sampled Bellman operator* $\hat{\mathcal{T}}$, defined as

$$\hat{\mathcal{T}}Q(s,a,s') = r(s,a) + \beta \cdot V_Q(s').$$

DEFINITION 3.3. *We define Temporal-Difference (TD) error for Q at the transition (s,a,s'), Squared TD error, and Expected squared TD error as follows:*

$$\delta_Q(s,a,s') := \hat{\mathcal{T}}Q(s,a,s') - Q(s,a)$$

$$\mathcal{L}_{TD}(Q)(s,a,s') := \left(\hat{\mathcal{T}}Q(s,a,s') - Q(s,a)\right)^2$$

$$\overline{\mathcal{L}_{TD}}(Q) := \mathbb{E}_{(s,a)\sim\pi^*,\, \nu_0}\left[\mathbb{E}_{s'\sim P(s,a)}\left[\mathcal{L}_{TD}(Q)(s,a,s')\right]\right]$$

Lemma 3.2 states the relationship between the TD error terms and Bellman error terms.

LEMMA 3.2 (EXPECTATION OF TD ERROR IS EQUIVALENT TO BE ERROR).

$$\mathbb{E}_{s'\sim P(s,a)}\left[\hat{\mathcal{T}}Q(s,a,s')\right] = \mathcal{T}Q(s,a) \text{ and } \mathbb{E}_{s'\sim P(s,a)}\left[\delta_Q(s,a,s')\right] = \mathcal{T}Q(s,a) - Q(s,a).$$

4 ERM-IRL (ERM-DDC) framework

4.1 Identification through ERM

We now propose a one-shot Empirical Risk Minimization framework (ERM-IRL/ERM-DDC) to solve the IRL problem stated in Definition 3.1. We recast the IRL problem as an ERM problem as follows:

$$\min_{Q\in\mathcal{Q}} \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\mathcal{L}_{NLL}(Q)(s,a) + \lambda \mathbb{1}_{a=a_s}\mathcal{L}_{BE}(Q)(s,a)\right], \tag{5}$$

where the first term, $\mathcal{L}_{NLL}(Q)(s,a) := -\log\left(\hat{p}_Q(a \mid s)\right)$, denotes the Negative Log-Likelihood (NLL) of Q with $\hat{p}_Q(a \mid s) := \frac{Q(s,a)}{\sum_{\tilde{a}\in\mathcal{A}} Q(s,\tilde{a})}$, the second term, $\mathcal{L}_{BE}(s,a)$, is the squared Bellman error, is from Definition 3.2, and a_s is defined in Assumption 3.2. We can thus write Equation (5) as follows:

$$\min_{Q\in\mathcal{Q}} \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[-\log\left(\hat{p}_Q(a \mid s)\right) + \lambda \mathbb{1}_{a=a_s}\left(\mathcal{T}Q(s,a) - Q(s,a)\right)^2\right]$$

Remark. The joint minimization of the NLL term and BE term is the key novelty in our approach. Prior work on the IRL and DDC literature [Hotz and Miller, 1993, Zeng et al., 2023] typically minimizes the log-likelihood of the observed choice probabilities (the NLL term), given observed or estimated state transition probabilities. The standard solution is to first estimate/assume state transition probabilities, then obtain estimates of future value functions, plug them into the choice probability, and then minimize NLL term. In contrast, our recast problem avoids the estimation of state-transition probabilities and instead jointly minimizes the NLL term along with the Bellman-error term. This is particularly helpful in large-state spaces since the estimation of state-transition probabilities can be infeasible/costly in such settings. In Theorem 4.1, we show that the solution to our recast problem in Equation (5) identifies the reward function.

THEOREM 4.1 (IDENTIFICATION THROUGH ERM).
The solution to the ERM problem (Equation (5)*) with any $\lambda > 0$ uniquely identifies Q^* up to $s \in \bar{\mathcal{S}}$ and $a \in \mathcal{A}$, i.e., finds $\widehat{Q}$ that satisfies $\widehat{Q}(s,a) = Q^*(s,a)$ for $s \in \bar{\mathcal{S}}$ and $a \in \mathcal{A}$. Furthermore, we can uniquely identify r up to $s \in \bar{\mathcal{S}}$ and $a \in \mathcal{A}$ by $r(s,a) = \widehat{Q}(s,a) - \beta \cdot \mathbb{E}_{s'\sim P(s,a)}\left[V_{\widehat{Q}}\right]$.*

Essentially, Theorem 4.1 ensures that solving Equation (5) gives the exact r and Q^* up to $\bar{\mathcal{S}}$ and thus provides the solution to the IRL problem defined in Definition 3.1. See Appendix B.3 for the proof.

4.2 Empirical challenge and the minimax resolution

While the idea of ERM-IRL framework – minimizing Equation (5) – is straightforward, its main challenge comes from the difficulty of empirically approximating $\mathcal{L}_{BE}(Q)(s,a) = (\mathcal{T}Q(s,a) - Q(s,a))^2$ and its gradient. As discussed in Section 3.3, $\mathcal{T}Q$ is not available unless we know the transition function.

As a result, we have to rely on an estimate of $\mathcal{T}$. A natural choice, common in TD-methods, is $\hat{\mathcal{T}}Q(s,a,s') = r(s,a) + \beta \cdot V_Q(s')$ which is computable given Q and data $\mathcal{D}$. Thus, a natural proxy objective to minimize is:

$$\mathbb{E}_{s'\sim P(s,a)}\left[\mathcal{L}_{\mathrm{TD}}(Q)(s,a,s')\right] := \mathbb{E}_{s'\sim P(s,a)}\left[\left(\hat{\mathcal{T}}Q(s,a,s') - Q(s,a)\right)^2\right]$$

Temporal Difference (TD) methods typically use stochastic approximation to obtain an estimate of this proxy objective [Adusumilli and Eckardt, 2019, Tesauro et al., 1995]. However, the issue with TD methods is that minimizing the proxy objective will not minimize the Bellman error in general (see Appendix B.1 for details), because of the extra variance term, as shown below.

$$\mathbb{E}_{s'\sim P(s,a)}\left[\mathcal{L}_{TD}(Q)(s,a,s')\right] = \mathcal{L}_{BE}(Q)(s,a) + \mathbb{E}_{s'\sim P(s,a)}\left[\left(\mathcal{T}Q(s,a) - \hat{\mathcal{T}}Q(s,a,s')\right)^2\right]$$

As defined, $\hat{\mathcal{T}}$ is a one-step estimator, and the second term in the above equation does not vanish even in infinite data regimes. So, simply using the TD approach to approximate squared Bellman error provides a biased estimate. Intuitively, this problem happens because expectation and square are not exchangeable, i.e., $\mathbb{E}_{s'\sim P(s,a)}\left[\delta_Q(s,a,s') \mid s,a\right]^2 \neq \mathbb{E}_{s'\sim P(s,a)}\left[\delta_Q(s,a,s')^2 \mid s,a\right]$. To remove this problematic square term, we employ an approach often referred to as the "Bi-Conjugate Trick"

which replaces a square function by a linear function called the bi-conjugate:

$$\begin{aligned}\mathcal{L}_{BE}(s,a)(Q) &:= \mathbb{E}_{s'\sim P(s,a)}\left[\delta_Q(s,a,s') \mid s,a\right]^2 \\ &= \max_{h\in\mathbb{R}} 2\cdot \mathbb{E}_{s'\sim P(s,a)}\left[\delta_Q(s,a,s') \mid s,a\right]\cdot h - h^2\end{aligned}$$

By further re-parametrizing h using $\zeta = h - r + Q(s,a)$, after some algebra, we arrive at Lemma 4.2. (See Appendix B.1 for the detailed derivation.)

LEMMA 4.2.
(a) We can express the squared Bellman error as

$$\begin{aligned}\mathcal{L}_{BE}(Q)(s,a) &:= (\mathcal{T}Q(s,a) - Q(s,a))^2 \\ &= \mathbb{E}_{s'\sim P(s,a)}\left[\mathcal{L}_{TD}(s,a,s')(Q)\right] - \beta^2 D(Q)(s,a)\end{aligned} \tag{6}$$

where

$$D(Q)(s,a) := \min_{\zeta\in\mathbb{R}} \mathbb{E}_{s'\sim P(s,a)}\left[\left(V_Q(s') - \zeta\right)^2 \mid s,a\right] \tag{7}$$

(b) Define the minimizer (over all states and actions) of objective (7) *as*

$$\zeta^* : (s,a) \mapsto \arg\min_{\zeta\in\mathbb{R}} \mathbb{E}_{s'\sim P(s,a)}\left[(V^*(s') - \zeta)^2 \mid s,a\right]$$

then $r(s,a) = Q^*(s,a) - \beta\zeta^*(s,a)$.

The reformulation of $\mathcal{L}_{BE}$ proposed in Lemma 4.2 enjoys the advantage of minimizing the squared TD-error ($\mathcal{L}_{TD}$) but without bias. Combining Theorem 4.1 and Lemma 4.2, we arrive at the following Theorem 4.3, which gives the ERM formulation of IRL we propose.

THEOREM 4.3 (ERM FORMULATION OF IRL). *Q^* uniquely identified as a solution to*

$$\begin{aligned}&\min_{Q\in\mathcal{Q}} \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\mathcal{L}_{NLL}(Q)(s,a) + \lambda\mathbb{1}_{a=a_s}\left\{\mathbb{E}_{s'\sim P(s,a)}\left[\mathcal{L}_{TD}(Q)(s,a,s')\right] - \beta^2 D(Q)(s,a)\right\}\right] \\ &= \min_{Q\in\mathcal{Q}}\max_{\zeta\in\mathbb{R}^{S\times A}} \mathbb{E}_{(s,a)\sim\pi^*,\nu_0,s'\sim P(s,a)}\Big[\underbrace{-\log\left(\hat{p}_Q(a\mid s)\right)}_{1)} + \lambda\mathbb{1}_{a=a_s}\Big\{\underbrace{\left(\hat{\mathcal{T}}Q(s,a,s') - Q(s,a)\right)^2}_{2)} \\ &\quad - \beta^2\underbrace{\left(\left(V_Q(s') - \zeta(s,a)\right)^2\right)}_{3)}\Big\}\Big]\end{aligned} \tag{8}$$

Furthermore, $r(s,a) = Q^*(s,a) - \beta\zeta^*(s,a)$ where ζ^* is defined in Lemma 4.2.

Equation (8) in Theorem 4.3 is a mini-max problem in terms of $Q \in \mathcal{Q}$ and the introduced dual function $\zeta \in \mathbb{R}^{S\times A}$. To summarize, term 1) is the negative log-likelihood equation, term 2) is the TD error, and term 3) introduces a dual function ζ. The introduction of the dual function ζ in term 3) may seem a bit strange. In particular, note that $\arg\max_{\zeta\in\mathbb{R}} -\mathbb{E}_{s'\sim P(s,a)}\left[\left(V_Q(s') - \zeta\right)^2 \mid s,a\right]$ is just $\zeta = \mathbb{E}_{s'\sim P(s,a)}[V(s') \mid s,a]$. However, we do not have access to the transition kernel and the state and action spaces may be large. Instead, we think of ζ as a function of states and actions, $\zeta(s,a)$ as introduced in Lemma 4.2. This parametrization allows us to optimize over a class of functions containing $\zeta(s,a)$ directly.

In the next section, we discuss an algorithm to do this optimization.

5 GLADIUS: Algorithm for ERM-IRL (ERM-DDC)

Denote $N := |\mathcal{D}|$ where $\mathcal{D}$ is a finite dataset. We define the *empirical* ERM-IRL objective, the finite-sample approximation of the ERM-IRL objective defined as equation (8), as follows:

$$\begin{aligned}
&\min_{Q\in\mathcal{Q}} \max_{\zeta\in\mathbb{R}^{S\times A}} \frac{1}{N} \sum_{(s,a,s')\in\mathcal{D}} \\
&\qquad \left[-\log\left(\hat{p}_Q(a \mid s)\right) + \lambda \mathbb{1}_{a=a_s}\left\{\left(\hat{\mathcal{T}}Q(s,a,s') - Q(s,a)\right)^2 - \beta^2\left(V_Q(s') - \zeta(s,a)\right)^2\right\}\right] \\
&= \min_{Q\in\mathcal{Q}} \frac{1}{N} \sum_{(s,a,s')\in\mathcal{D}} \left[-\log\left(\hat{p}_Q(a \mid s)\right)\right. \\
&\qquad \left. + \lambda \mathbb{1}_{a=a_s}\left\{\left(\hat{\mathcal{T}}Q(s,a,s') - Q(s,a)\right)^2 - \beta^2 \min_{\zeta\in\mathbb{R}^{S\times A}} \sum_{(s,a,s')\in\mathcal{D}} \left(V_Q(s') - \zeta(s,a)\right)^2\right\}\right] \qquad (9)
\end{aligned}$$

Algorithm 1 solves the equation (9) through an alternating gradient ascent descent algorithm we call Gradient-based Learning with Ascent-Descent for Inverse Utility learning from Samples (GLADIUS). Given the function class $\mathcal{Q}$ of value functions, let $Q_{\theta_1} \in \mathcal{Q}$ and $\zeta_{\theta_2} \in \mathbb{R}^{S\times A}$ denote the functional representation of Q and ζ. Our goal is to learn the parameters $\boldsymbol{\theta}^* = \{\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*\}$, that together characterize $\hat{Q}$ and $\hat{\zeta}$. Each iteration in the GLADIUS algorithm consists of the following two steps:

(1) Gradient Ascent: Update ζ_{θ_2} based on the current value of Q_{θ_1}.
(2) Gradient Descent: Update $\boldsymbol{\theta}_1$ based on the current value of ζ_{θ_2}.

After a fixed number of gradient steps of Q_{θ_1} and ζ_{θ_2} (which we can denote as $\hat{Q}$ and $\hat{\zeta}$), we can compute the reward prediction $\hat{r}$ as $\hat{r}(s,a) = \hat{Q}(s,a) - \beta\hat{\zeta}(s,a)$ due to Theorem 4.3.

ALGORITHM 1: **G**radient-based **L**earning with **A**scent-**D**escent for **I**nverse **U**tility learning from **S**amples (GLADIUS)

Input: Offline dataset $\mathcal{D} = \{(s,a,s')\}$, time horizon T
Output: $\widehat{r}$, $\widehat{Q}$
Initialize Q_{θ_1}, ζ_{θ_2}, iteration $\leftarrow 1$
while $t \le T$ **do**
  Draw batches B_1, B_2 from $\mathcal{D}$
  [Ascent Step: Update ζ_{θ_1}, fixing Q_{θ_2} and V_{θ_2}]
  $D_{\theta_1} \leftarrow \sum_{(s,a,s')\in B_2} \left(V_{\theta_2}(s') - \zeta_{\theta_1}(s,a)\right)^2$ **where** $V_{\theta}(s') := \log\sum_{\tilde{a}\in\mathcal{A}} \exp(Q_{\theta}(s',\tilde{a}))$
  $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_1 - \tau_1 \nabla_{\boldsymbol{\theta}_1} D_{\boldsymbol{\theta}_1}$
  [Descent Step: Update Q_{θ_2} and V_{θ_2}, fixing ζ_{θ_1}]
  $\overline{\mathcal{L}_{NLL}} \leftarrow \sum_{(s,a,s')\in B_2} -\log(\hat{p}_{\theta_2}(a \mid s))$
  $\overline{\mathcal{L}_{\text{BE}}} \leftarrow \sum_{(s,a,s')\in B_1} \mathbb{1}_{a=a_s}\left[\mathcal{L}_{TD}(Q)(s,a,s') - \beta^2(V_{\theta_2}(s') - \zeta_{\theta_1}(s,a))^2\right]$
    where $\mathcal{L}_{TD}(Q)(s,a,s') := \left(\hat{\mathcal{T}}Q(s,a,s') - Q(s,a)\right)^2$
  $\mathcal{L}_{\theta_2} \leftarrow \overline{\mathcal{L}_{\text{NLL}}} + \lambda\overline{\mathcal{L}_{\text{BE}}}$
  $\boldsymbol{\theta}_2 \leftarrow \boldsymbol{\theta}_2 - \tau_1 \nabla_{\boldsymbol{\theta}_2} \mathcal{L}_{\theta_2}$
  iteration $\leftarrow$ iteration + 1
end
$\widehat{\zeta} \leftarrow \zeta_{\theta_1}$, $\widehat{Q} \leftarrow Q_{\theta_2}$
$\widehat{r}(s,a) \leftarrow \widehat{Q}(s,a) - \beta \cdot \widehat{\zeta}(s,a)$

Special Case: Deterministic Transitions

When the transition function is *deterministic* (e.g., in Guo et al. [2025], Rafailov et al. [2024], Zhong et al. [2024]) meaning that for any state-action pair (s, a), the next state s' is uniquely determined, the ascent step involving ζ is no longer required. This is because the term $\left(V_Q(s') - \zeta(s,a)\right)^2$ (highlighted in orange in equation (8) and (9)) becomes redundant in the empirical ERM-IRL objective, because $\max_{\zeta \in \mathbb{R}} -\mathbb{E}_{s' \sim P(s,a)} \left[\left(V_Q(s') - \zeta\right)^2 \mid s, a \right]$ is always 0. Consequently, the optimization simplifies to:

$$\min_{Q \in \mathcal{Q}} \frac{1}{N} \sum_{(s,a,s') \in \mathcal{D}} \left[-\log\left(\hat{p}_Q(a \mid s)\right) + \lambda \mathbb{1}_{a=a_s} \left(\hat{\mathcal{T}} Q(s,a,s') - Q(s,a)\right)^2 \right] \tag{10}$$

Under deterministic transitions, the **GLADIUS** algorithm reduces to a single gradient descent update step for Q_θ, eliminating the need for the alternating ascent-descent updates. Consequently, the estimated reward function is computed as:

$$\hat{r}(s,a) = \hat{Q}(s,a) - \beta V_Q(s')$$

The updated algorithm proceeds as follows:

ALGORITHM 2: GLADIUS under Deterministic Transitions

Input: Offline dataset $\mathcal{D} = \{(s, a, s')\}$, time horizon T
Output: $\widehat{r}$, $\widehat{Q}$
Initialize Q_θ, iteration $\leftarrow 1$
while $t \leq T$ **do**
  Draw batch B from $\mathcal{D}$
  $\overline{\mathcal{L}_{NLL}} \leftarrow \sum_{(s,a,s') \in B} -\log(\hat{p}_\theta(a \mid s))$
  $\overline{\mathcal{L}_{\text{BE}}} \leftarrow \sum_{(s,a,s') \in B} \mathbb{1}_{a=a_s} \mathcal{L}_{TD}(Q)(s,a,s')$
    where $\mathcal{L}_{TD}(Q)(s,a,s') := \left(\hat{\mathcal{T}} Q(s,a,s') - Q(s,a)\right)^2$
  $\mathcal{L}_\theta \leftarrow \overline{\mathcal{L}_{\text{NLL}}} + \lambda \overline{\mathcal{L}_{\text{BE}}}$
  $\theta \leftarrow \theta - \tau \nabla_\theta \mathcal{L}_\theta$
  iteration $\leftarrow$ iteration + 1
end
$\widehat{Q} \leftarrow Q_\theta$
$\widehat{r}(s,a) \leftarrow \widehat{Q}(s,a) - \beta \log \sum_{\tilde{a} \in \mathcal{A}} \exp(\widehat{Q}(s', \tilde{a}))$

Key Differences in the Deterministic Case:

- No Ascent Step: The ascent step for ζ is removed since the term $\left(V_Q(s') - \zeta(s,a)\right)^2$ disappears.
- Single Gradient Descent: The algorithm updates Q_θ via a single gradient descent step per iteration.
- Reward Computation: The reward function is computed as $\hat{r}(s,a) = \hat{Q}(s,a) - \beta V_Q(s')$.

This modification makes GLADIUS computationally more efficient when applied to deterministic environments while maintaining the correct theoretical formulation of the Q^* and reward functions.

6 Theory and analysis of Algorithm

As discussed in the previous section, Equation (9) represents a mini-max optimization problem. Such problems are known to be globally solvable by a simple gradient ascent-descent algorithm when it is a concave-convex mini-max problem. However, the challenge is that Equation (8) is not a concave-convex mini-max problem. Given Q, it determines ζ that serves as the Bayes-optimal estimator for $\mathbb{E}_{s' \sim P(s,a)} \left[V_Q(s') \mid s, a \right]$. This implies that $-\mathbb{E}_{s' \sim P(s,a)} \left[\left(V_Q(s') - \zeta\right)^2 \mid s, a \right]$ is strongly

concave in ζ. On the other hand, given such an optimal ζ, $\mathcal{L}_{BE}(Q)(s,a)$ term is *not* convex in Q [Bas-Serrano et al., 2021]. The key result in this section is proving that both $\mathcal{L}_{BE}(Q)(s,a)$ and $\mathcal{L}_{\text{NLL}}(Q)(s,a) = \left[-\log\left(\hat{p}_Q(a \mid s)\right)\right]$ satisfies the *Polyak-Łojasiewicz* (PL) condition under certain assumptions, which is enough for Algorithm 1 to converge to global optima.

6.1 Polyak-Łojasiewicz (PL) in terms of Q

The Polyak-Łojasiewicz (PL) condition prevents the gradient from vanishing prematurely, keeping optimization progress steady. That is, it nearly possesses the smooth, fast convergence behavior of strongly convex functions. Throughout, we use $\|Q\|_{L^2(\pi^*,\nu_0)} := \left(\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[Q(s,a)^2\right]\right)^{1/2}$ for $Q \in \mathcal{Q}$.

DEFINITION 6.1 (POLYAK-ŁOJASIEWICZ (PL) CONDITION WITH RESPECT TO L^2 NORM). *A function $f : \mathcal{Q} \mapsto \mathbb{R}$ is is said to satisfy the Polyak-Łojasiewicz (PL) condition with respect to L^2 norm with measure μ if f has a nonempty solution set and a finite minimal value $f(Q^*)$ for $Q^* \in \mathcal{Q}$, and there exists some $c > 0$ such that $\frac{1}{2}\|\nabla_Q f(Q)\|^2_{L^2(\mu)} \geq c(f(Q) - f(Q^*)), \forall x \in \mathcal{X}$.*

Remark. Note, in this definition, we are identifying $\mathcal{Q}$ as a subset of $\mathbb{R}^{S\times A}$ hence the derivative is defined appropriately.

To prove PL, we need the following Lemmas which describes the behavior of $\overline{\mathcal{L}_{\text{NLL}}}(Q)$ and $\overline{\mathcal{L}_{\text{BE}}}(Q)$.

LEMMA 6.1. $\overline{\mathcal{L}_{\text{NLL}}}(Q) := \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[-\log\left(\hat{p}_Q(a \mid s)\right)\right]$ *is convex and Lipschitz smooth in Q in terms of $L^2(\pi^*, \nu_0)$ norm.*

LEMMA 6.2. $\overline{\mathcal{L}_{\text{BE}}}(Q) := \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\mathcal{L}_{\text{BE}}(Q)(s,a)\right]$ *is of C^2 and Lipschitz smooth in Q in terms of $L^2(\pi^*, \nu_0)$ norm.*

Given Lemma 6.1 and 6.2, the following Theorems that $\overline{\mathcal{L}_{\text{NLL}}}(Q)$ and $\overline{\mathcal{L}_{\text{BE}}}(Q)$ satisfies PL condition.

THEOREM 6.3. *For given $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $\mathcal{L}_{BE}(Q)(s,a)$ satisfies PL condition with respect to Q. Furthermore, $\overline{\mathcal{L}_{BE}}(Q)$ satisfies the PL condition with respect to Q in terms of $L^2(\pi^*, \nu_0)$.*

THEOREM 6.4. *For given $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $\mathcal{L}_{NLL}(Q)(s,a)$ satisfies PL condition with respect to Q. Furthermore, $\overline{\mathcal{L}_{NLL}}(Q)$ satisfies the PL condition with respect to Q in terms of $L^2(\pi^*, \nu_0)$.*

In general, the sum of two PL functions is not necessarily PL. However, according to the following Lemma 6.5, our problem is a special case where such property holds,

LEMMA 6.5. $\overline{\mathcal{L}_{\text{NLL}}}(Q) + \lambda \mathbb{1}_{a=a_s}\overline{\mathcal{L}_{\text{BE}}}(Q)$ *satisfies PL in terms of Q in terms of $L^2(\pi^*, \nu_0)$.*

We remark that this result by itself, establishes the PL condition in the *tabular* setting with finite states and actions where the Q function is parametrized as a vector/matrix in $\mathbb{R}^{S\times A}$. In the next section, we extend this to a more general hypothesis class.

6.2 Polyak-Łojasiewicz (PL) in terms of θ

We now extend previous section to cases where the underlying state and action spaces are potentially featurized, i.e. $\mathcal{S} = \mathbb{R}^{\dim(\mathcal{S})}$ and $\mathcal{A} = \mathbb{R}^{\dim(\mathcal{A})}$. (In this case, $L^2(\pi^*, \nu_0)$ norm is reduced to the (weighted) euclidean norm with dimension $\dim(\mathcal{S}) + \dim(\mathcal{A})$.) When $\dim(\mathcal{S}) + \dim(\mathcal{A})$ is very large, it is often preferable/necessary to approximate Q^* using a set of parametrized functions $\mathcal{Q} = \left\{Q_\theta : \mathbb{R}^{\dim(\mathcal{S})+\dim(\mathcal{A})} \to \mathbb{R} \mid \theta \in \Theta \subseteq \mathbb{R}^d, Q_\theta \in \mathcal{F}\right\}$, where $\mathcal{F}$ denotes a class of functions such as linear, polynomial or deep neural network function class that is parametrized by θ. In this case, we make the following assumption often called the *realizability* assumption.

ASSUMPTION 6.1 (REALIZABILITY). *$\mathcal{Q}$ contains an optimal function Q^*, meaning there exists $\theta^* \in \Theta$ such that $Q_{\theta^*} = Q^*$.*

Under this parametrization, the ERM-IRL problem (the equation (5)) becomes

$$\min_{\boldsymbol{\theta}\in\Theta\subseteq\mathbb{R}^d} \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\mathcal{L}_{NLL}(Q_{\boldsymbol{\theta}})(s,a) + \lambda \mathbb{1}_{a=a_s}\mathcal{L}_{BE}(Q_{\boldsymbol{\theta}})(s,a)\right]$$

Our next question is whether our previous result – showing that the equation (5) being PL in terms of Q – can ensure that this new equation is also PL in terms of $\boldsymbol{\theta}$, which is defined as follows.

DEFINITION 6.2 (POLYAK-ŁOJASIEWICZ (PL) CONDITION WITH RESPECT TO ℓ_2 NORM). *Given $\Theta \in \mathbb{R}^d$, a function $h : \Theta \mapsto \mathbb{R}$ is is said to satisfy the Polyak-Łojasiewicz (PL) condition with respect to ℓ_2 norm if h has a nonempty solution set and a finite minimal value $h(\boldsymbol{\theta}^*)$ for $\boldsymbol{\theta}^* \in \Theta \subseteq \mathbb{R}^d$, and there exists some $c > 0$ such that $\frac{1}{2}\|\nabla h(\boldsymbol{\theta})\|_2^2 \geq c(h(\boldsymbol{\theta}) - h(\boldsymbol{\theta}^*)), \forall \boldsymbol{\theta} \in \Theta$.*

In this paper, we restrict our attention to the function class $\mathcal{Q}$ which satisfies the Assumption 6.2.

ASSUMPTION 6.2. *For $Q_{\boldsymbol{\theta}} \in \mathcal{Q}$,*
1. $Q_\theta(s,a)$ is continuously differentiable with respect to $\boldsymbol{\theta}$, meaning its Jacobian

$$DQ_{\boldsymbol{\theta}} := \frac{\partial Q_{\boldsymbol{\theta}}(s,a)}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{(\dim(\mathcal{S})+\dim(\mathcal{A}))\times d}$$

exists and is well-defined.
2. There exists a constant $m > 0$ such that for all $\boldsymbol{\theta} \in \Theta$,

$$\sigma_{\min}(DQ_{\boldsymbol{\theta}}) \geq m$$

where $\sigma_{\min}(DQ_{\boldsymbol{\theta}})$ is the smallest singular value of $DQ_{\boldsymbol{\theta}}$.

The two lemmas show that Assumption 6.2 is easy to satisfy by popular function classes such as linear and the neural network function class.

LEMMA 6.6. *Let $Q_{\boldsymbol{\theta}}(s,a) = \boldsymbol{\theta}^\top \phi(s,a)$, where the known feature mapping $\phi : \mathcal{S}\times\mathcal{A} \mapsto \mathbb{R}^d$, satisfies $\|\phi(s,a)\| \leq B$ almost surely with respect to $(s,a) \sim (\pi^*, \nu_0)$ for some $B > 0$. Then dataset size $|\mathcal{D}| \geq Cd$ implies that Assumption 6.2 holds with probability at least $1 - e^{-C|\mathcal{D}|}$.*

LEMMA 6.7 ([PENNINGTON ET AL., 2017]). *Let $Q_{\boldsymbol{\theta}}$ be a deep nonlinear neural network composed of smooth activation functions (e.g., sigmoid) and linear layers parameterized by $\boldsymbol{\theta}$. When initialized using orthogonal weight initialization, $Q_{\boldsymbol{\theta}}$ satisfies Assumption 6.2.*

The following Theorem 6.8 shows that satisfying Assumption 6.1 and 6.2 is enough to achieve *PL* condition in terms of $\boldsymbol{\theta}$. That is, linear, polynomial, neural network-parametrization satisfies PL. This also subsumes the previous result in the tabular case with $d = S \times A$, the states encoded as standard basis vectors $\theta = \{Q(s,a)\}_{s,a\in S\times A}$.

THEOREM 6.8. *Suppose that Assumption 6.1 and 6.2 are satisfied for Θ. Then the loss function $\overline{\mathcal{L}_{\mathrm{NLL}}}(Q_{\boldsymbol{\theta}}) + \lambda \mathbb{1}_{a=a_s}\overline{\mathcal{L}_{\mathrm{BE}}}(Q_{\boldsymbol{\theta}})$ satisfies the Polyak–Łojasiewicz (PL) condition in terms of $\boldsymbol{\theta}$.*

6.3 Global convergence of GLADIUS

Denote $N := |\mathcal{D}|$ where $\mathcal{D}$ is a finite dataset. We define Q_N^* as the solution to the empirical ERM-IRL objective (equation (9)), i.e.,

$$Q_N^* \in \arg\min_{Q\in\mathcal{Q}} \max_{\zeta\in\mathbb{R}^{S\times A}} \frac{1}{N}\sum_{(s,a,s')\in\mathcal{D}} \left[-\log\left(\hat{p}_Q(a\mid s)\right) + \lambda \mathbb{1}_{a=a_s}\left\{\left(\hat{\mathcal{T}}Q(s,a,s') - Q(s,a)\right)^2 - \beta^2\left(\left(V_Q(s') - \zeta(s,a)\right)^2\right)\right\}\right]$$

Note that Q^*_N approaches to Q^* as $N \to \infty$, the solution to the ERM-IRL objective (equation (8)), by the law of large numbers. Formally, we define the *statistical error*, $\epsilon_{\text{stat}}(N)$, as

$$\epsilon_{\text{stat}}(N) := \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\left(Q^*(s,a) - Q^*_N(s,a)\right)^2\right]$$

Also, define $\hat{Q}_T$ be the T iteration outcome of Algorithm 1. Then we can define the *optimization error* of GLADIUS (Algorithm 1) at its Tth iteration given data $\mathcal{D}$, $\epsilon_{\text{opt}}(T)$, as

$$\epsilon_{\text{opt}}(T) := \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\left(Q^*_N(s,a) - \hat{Q}_T(s,a)\right)^2\right]$$

PROPOSITION 6.1 (GLOBAL CONVERGENCE). *Define $\hat{Q}_T$ and $\hat{r}_T$ be the estimators of Q^* and r after T iterations of GLADIUS algorithm on the N-sample offline data. Then under Assumption 6.2 and Assumption 6.1, $\epsilon_{opt}(T) \le O(1/T)$ and $\epsilon_{stat}(N) \le O(1/\sqrt{N})$. This implies that*

$$\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\left(Q^*(s,a) - \hat{Q}_T(s,a)\right)^2\right] \le \epsilon_{opt}(T) + \epsilon_{stat}(N) \le O(1/T) + O(1/\sqrt{N})$$

Remark. To the best of our knowledge, no prior work has proposed an algorithm that guarantees *global* optimum convergence of the minimization problem that involves $\mathcal{L}_{BE}(Q)(s,a)$ term.[10]. With this regard, Theorem 6.3 and 6.8 have an important implication for Offline reinforcement learning (Offline RL) [Jiang and Xie, 2024]. Gradient-based Offline reinforcement learning [Antos et al., 2008, Dai et al., 2018], which minimizes $\overline{\mathcal{L}_{\text{BE}}}(Q_\theta)$ in the same way as GLADIUS does, has been been proven to be convergent. However, its global convergence guarantee has not yet been established. Theorem 6.3 and 6.8 establishes that gradient-based Offline RL is indeed globally convergent for important function classes such as tabular, linear, polynomial and neural network function classes.

7 Simulation experiments

We now present results from simulation experiments, in which we compare the performance of our approach against a series of benchmark algorithms.

In the main text, we use the high-dimensional version of the canonical bus engine replacement problem ([Rust, 1994]) as the setting for our experiments. This setting has been extensively used as the standard benchmark for the reward learning problem in the DDC literature in economics [Aguirregabiria and Mira, 2002, Arcidiacono and Ellickson, 2011, Arcidiacono and Miller, 2011, Barzegary and Yoganarasimhan, 2022, Chernozhukov et al., 2022, Chiong et al., 2016, Geng et al., 2023, Hotz and Miller, 1993, Kasahara and Shimotsu, 2009, Norets, 2009, Reich, 2018, Su and Judd, 2012, Yang, 2024].

In addition, in Appendix A, for completeness, we also provide OpenAI gym benchmark environment experiments with a discrete action space (Lunar Lander, Acrobot, and Cartpole) [Brockman, 2016] as in Garg et al. [2021] for the related, but easier problem of imitation learning.

7.1 Experimental Setup

The bus engine replacement problem [Rust, 1987] is a simple regenerative optimal stopping problem. In this setting, the manager of a bus company operates many identical buses. As a bus accumulates mileage, its per-period maintenance cost increases. The manager can replace the engine in any period (which then becomes as good, and this replacement decision re-sets the mileage to one). However, the replacement decision comes with a high fixed cost. Each period, the manager makes a dynamic trade-off between either replacing the engine or continuing with maintenance. We observe the manager's decisions for a fixed set of buses, i.e., a series of states, decisions, and state transitions. Our goal is to

[10] Some studies, such as [Dai et al., 2018], have demonstrated convergence to a stationary point of this mini-max problem.

learn the manager's reward function from these observed trajectories under the assumption that he made these decisions optimally.

Dataset. There are N independent and identical buses (trajectories) indexed by j, each of which has 100 periods over which we observe them, i.e., $h \in \{1 \dots 100\}$. Each bus's trajectory starts with an initial mileage of 1. The only reward-relevant state variable at period h is the mileage of bus $x_{jh} \in \{1, 2, \dots 20\}$.

Decisions and rewards. There are two possible decisions at each period, replacement or continuation, denoted by $d_{jh} = \{0, 1\}$. $d_{jh} = 1$ denotes replacement, and there is a fixed cost θ_1 of replacement. Replacement resets the mileage to 1, i.e., the engine is as good as new. $d_{jh} = 0$ denotes maintenance, and the cost of maintaining the engine depends on the mileage as follows: $\theta_0 x_{jh}$. Intuitively, the manager can pay a high fixed cost θ_1 for replacing an engine in this period but doing so reduces future maintenance costs since the mileage is reset to 1. In all our experiments, we set $\theta_0 = 1$ (maintenance cost) and $\theta_1 = 5$ (replacement cost). Additionally, we set the discount factor to $\beta = 0.95$.

State transitions at each period. If the manager chooses maintenance, the mileage advances by 1, 2, 3, or 4 with a $1/4$ probability each. If the manager chooses to replace the engine, then the mileage is reset to 1. That is, $\mathbb{P}(\{x_{j(h+1)} = x_{jh} + k\} \mid d_{jh} = 0) = 1/4$, $k \in \{1, 2, 3, 4\}$ and $\mathbb{P}\{x_{j(h+1)} = 1 \mid d_{jh} = 1\} = 1$. When the bus reaches the maximum mileage of 20, we assume that mileage remains at 20 even if the manager continues to choose maintenance.

High-dimensional setup. In some simulations, we consider a high-dimensional version of the problem, where we now modify the basic set-up described above to include a set of K high-dimensional state variables, similar to Geng et al. [2023]. Assume that we have access to an additional set of K state variables $\{s^1_{jh}, s^2_{jh}, s^3_{jh} \dots s^K_{jh}\}$, where each s^k_{jh} is an i.i.d random draw from $\{-10, -9, \dots, 9, 10\}$. We vary K from 2 to 100 in our empirical experiments to test the sensitivity of our approach to the dimensionality of the problem. Further, we assume that these high-dimensional state variables s^k_{jh}s do not affect the reward function or the mileage transition probabilities. However, the researcher does not know this. So, they are included in the state space, and ideally, our algorithm should be able to infer that these state variables do not affect rewards and/or value function and recover the true reward function.

Traing/testing split. Throughout, we keep 80% of the trajectories in any experiment for training/learning the reward function, and the remaining 20% is used for evaluation/testing.

Functional form. For all non-parametric estimation methods (including ours), we used a multi-layer perception (MLP) with two hidden layers and 10 perceptrons for each hidden layer for the estimation of Q-function. For parametric-oracle methods, we use the reward functions' true parametric form (as described earlier).

7.2 Benchmark Algorithms

We compare our algorithm against a series of standard, or state-of-art benchmark algorithms in the DDC and IRL settings.

Rust (Oracle) Rust is an oracle-like fixed point iteration baseline that uses the nested fixed point algorithm [Rust, 1987]. It assumes the knowledge of: (1) linear parametrization of rewards by θ_1 and θ_2 as described above, and (2) the exact transition probabilities.

ML-IRL (Oracle) ML-IRL from Zeng et al. [2023] is the state-of-art offline IRL algorithm that minimizes negative log-likelihood of choice (i.e., the first term in Equation (5)). This method requires a separate estimation of transition probabilities, which is challenging in high-dimensional settings. So, we make the same Oracle assumptions as we did for Rust (Oracle), i.e., assume that transition probabilities are known. Additionally, to further improve this method, we leverage the finite dependence property of the problem [Arcidiacono and Miller, 2011], which helps avoid roll-outs.

SAmQ SAmQ [Geng et al., 2023] fits approximated soft-max Value Iteration (VI) to the observed data. We use the SAmQ implementation provided by the authors[11]; their code did not scale due to a memory overflow issue for the trajectories larger than 2500, i.e., 250,000 samples.
IQ-learn IQ-learn is a popular gradient-based method, maximizing occupancy matching objective (which does not guarantee that the Bellman equation is satisfied – See Section D).
BC Behavioral Cloning (BC) simply minimizes the expected negative log-likelihood. This simple algorithm outperforms [Zeng et al., 2023, Ziniu et al., 2022] many recent algorithms such as ValueDICE [Kostrikov et al., 2019]. For details, refer to Appendix A.2.1.

7.3 Experiment results

7.3.1 Performance results for the standard bus engine setting. Table 2 provides a table of simulation experiment results without dummy variables, i.e., with only mileage (x_{jh}) as the relevant state variable. The performance of algorithms was compared in terms of *mean absolute percentage error (MAPE)* of r estimation, which is defined as $\frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{r}_i - r_i}{r_i}\right| \times 100$, where N is the total number of samples from expert policy π^* and $\hat{r}_i$ is each algorithm's estimator for the true reward r_i.[12][13]

No. of Trajectories (H=100)	Oracle Baselines		Nonparametric, No Knowledge of Transition Probabilities			
	Rust	ML-IRL	**GLADIUS**	SAmQ	IQ-learn	BC
	MAPE (SE)	MAPE (SE)	MAPE (SE)	MAPE (SE)	MAPE (SE)	MAPE (SE)
50	3.62 (1.70)	3.62 (1.74)	**3.44** (1.28)	4.92 (1.20)	114.13 (26.60)	80.55 (12.82)
250	1.37 (0.77)	1.10 (0.78)	**0.84** (0.51)	3.65 (1.00)	112.86 (27.31)	72.04 (13.21)
500	0.90 (0.56)	0.84 (0.59)	**0.55** (0.20)	3.13 (0.86)	113.27 (25.54)	71.92 (12.44)
1000	0.71 (0.49)	0.64 (0.48)	**0.52** (0.22)	1.55 (0.46)	112.98 (24.12)	72.17 (12.11)
2500	0.68 (0.22)	0.62 (0.35)	**0.13** (0.06)	N/A	111.77 (23.99)	62.61 (10.75)
5000	0.40 (0.06)	0.43 (0.26)	**0.12** (0.06)	N/A	119.18 (22.55)	46.45 (8.22)

Based on 20 repetitions. Oracle baselines (Rust, MLIRL) were based on bootstrap repetition of 100.

Table 2. Mean Absolute Percentage Error (MAPE) (%) of r Estimation. (# dummy = 0)

We find that GLADIUS performs much better than non-oracle baselines and performs at least on par with, or slightly better than oracle baselines. A natural question here is why the Oracle baselines that leverage the exact transition function and the precise linear parametrization do not beat our approach. The primary reason for this outcome is the imbalance of state-action distribution from expert policy: (See Table 3 and Appendix A.1)

(1) All trajectories start from mileage 1. In addition, the replacement action (action 0) resets the mileage to 1. Therefore, most states observed in the expert data are within mileage 1-5. This imbalance leads to a small instability in parametric estimation [Fithian and Hastie, 2014]. This makes Oracle baseline predictions for states with mileage 1-5 slightly worse than GLADIUS.
(2) Since we evaluate MAPE on the police played in the data, this implies that our evaluation mostly samples mileages 1–5, and GLADIUS's weakness in extrapolation for mileage 6-10 matters less than the slight imprecision of parametric methods in mileages 1–5.

Finally, it is not surprising to see IQ-learn and BC underperform in the reward function estimation task since they do not require/ensure that the Bellman condition holds. See Appendix A.2 for a detailed discussion.

[11] https://github.com/gengsinong/SAmQ

[12] In the simulation we consider, we don't have a state-action pair with true reward near 0.

[13] As we assume that the data was collected from agents following (entropy regularized) optimal policy π^* (Assumption 3.1), the distribution of states and actions in the data is the best data distribution choice.

Mileages	1	2	3	4	5	6	7	8	9	10
Frequency	7994	1409	1060	543	274	35	8	1	0	0
True reward	-1.000	-2.000	-3.000	-4.000	-5.000	-6.000	-7.000	-8.000	-9.000	-10.000
ML-IRL	-1.013	-2.026	-3.039	-4.052	-5.065	-6.078	-7.091	-8.104	-9.117	-10.130
Rust	-1.012	-2.023	-3.035	-4.047	-5.058	-6.070	-7.082	-8.093	-9.105	-10.117
GLADIUS	-1.000	-1.935	-2.966	-3.998	-4.966	-5.904	-6.769	-7.633	-8.497	-9.361

Table 3. Estimated rewards and frequency values for 1,000 trajectories for action 0.

7.3.2 Performance results for the high-dimensional set-up. Figure 1 (below) presents high-dimensional experiments, where states were appended with dummy variables. Each dummy variable is of dimension 20. Note that a state space of dimensionality 20^{10} (10 dummy variables with 20 possible values each) is equivalent to 10^{13}, which is infeasible for existing exact methods (e.g., Rust) and methods that require transition probability estimation (e.g., ML-IRL). Therefore, we only present comparisons to the non-oracle methods.

Dummy Variables	**GLADIUS**	SAmQ	IQ-learn	BC
	MAPE (SE)	MAPE (SE)	MAPE (SE)	MAPE (SE)
2	**1.24** (0.45)	1.79 (0.37)	112.0 (14.8)	150.9 (29.1)
5	**2.51** (1.19)	2.77 (0.58)	192.2 (19.2)	171.1 (37.3)
20	**6.07** (3.25)	N/A	180.1 (15.6)	180.0 (33.7)
50	**9.76** (3.68)	N/A	282.2 (25.2)	205.1 (35.3)
100	**11.35** (4.24)	N/A	321.1 (23.1)	288.8 (42.9)

Based on 10 repetitions. For SAmQ, N/A means that the algorithm did not scale.

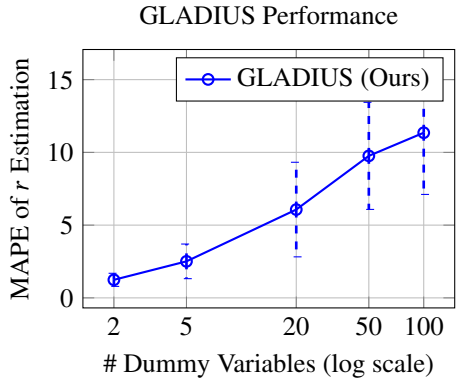

Fig. 1. MAPE of r estimation. The left panel shows the MAPE values in a tabular format, and the right panel visualizes the GLADIUS's performance on a log-scaled x-axis. 1000 trajectories were used for all experiments. Smaller is better; the best value in each row is highlighted.

We find that our approach outperforms benchmark algorithms, including SAmQ, IQ-learn, and BC (see Figure 1). Further, as shown in the right panel of Figure 1, the MAPE error exhibits sub-linear scaling with respect to the state dimension size (note that the x-axis is in log scale). This suggests that the algorithm can scale well to applications with large dimensional state space.

8 Conclusion

In this paper, we propose a provably globally convergent empirical risk minimization framework that combines non-parametric estimation methods (e.g., machine learning methods) with IRL/DDC models. This method's convergence to global optima stems from our new theoretical finding that the Bellman error (i.e., Bellman residual) satisfies the Polyak-Łojasiewicz (PL) condition, which is a weaker but almost equally useful condition as strong convexity for providing theoretical assurances.

The three key advantages of our method are: (1) it is easily applicable to high-dimensional state spaces, (2) it can operate without the knowledge of (or requiring the estimation of) state-transition probabilities, and (3) it is applicable to infinite state and action spaces. These three properties make our algorithm practically applicable and useful in high-dimensional, infinite-size state and action spaces

that are common in business and economics applications. We demonstrate our approach's empirical performance through extensive simulation experiments (covering both low and high-dimensional settings). We find that, on average, our method performs quite well in recovering rewards in both low and high-dimensional settings. Further, it has better/on-par performance compared to other benchmark algorithms in this area (including algorithms that assume the parametric form of the reward function and knowledge of state transition probabilities) and is able to recover rewards even in settings where other algorithms are not viable.

References

Karun Adusumilli and Dita Eckardt. 2019. Temporal-Difference estimation of dynamic discrete choice models. *arXiv preprint arXiv:1912.09509* (2019).

Victor Aguirregabiria and Pedro Mira. 2002. Swapping the nested fixed point algorithm: A class of estimators for discrete Markov decision models. *Econometrica* 70, 4 (2002), 1519–1543.

Victor Aguirregabiria and Pedro Mira. 2007. Sequential estimation of dynamic discrete games. *Econometrica* 75, 1 (2007), 1–53.

András Antos, Csaba Szepesvári, and Rémi Munos. 2008. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning* 71 (2008), 89–129.

Peter Arcidiacono, Patrick Bayer, Federico A Bugni, and Jonathan James. 2013. Approximating high-dimensional dynamic models: Sieve value function iteration. In *Structural Econometric Models*. Emerald Group Publishing Limited, 45–95.

Peter Arcidiacono and Paul B Ellickson. 2011. Practical methods for estimation of dynamic discrete choice models. *Annu. Rev. Econ.* 3, 1 (2011), 363–394.

Peter Arcidiacono and Robert A Miller. 2011. Conditional choice probability estimation of dynamic discrete choice models with unobserved heterogeneity. *Econometrica* 79, 6 (2011), 1823–1867.

Susan Athey and Guido Imbens. 2016. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences* 113, 27 (2016), 7353–7360.

Ebrahim Barzegary and Hema Yoganarasimhan. 2022. A Recursive Partitioning Approach for Dynamic Discrete Choice Modeling in High Dimensional Settings. *arXiv preprint arXiv:2208.01476* (2022).

Joan Bas-Serrano, Sebastian Curi, Andreas Krause, and Gergely Neu. 2021. Logistic Q-learning. In *International conference on artificial intelligence and statistics*. PMLR, 3610–3618.

Hugo Benitez-Silva, George Hall, Günter J Hitsch, Giorgio Pauletto, and John Rust. 2000. A comparison of discrete and parametric approximation methods for continuous-state dynamic programming problems. *manuscript, Yale University* (2000).

Jérôme Bolte, Tam Le, and Edouard Pauwels. 2023. Subgradient sampling for nonsmooth nonconvex minimization. *SIAM Journal on Optimization* 33, 4 (2023), 2542–2569.

G Brockman. 2016. OpenAI Gym. *arXiv preprint arXiv:1606.01540* (2016).

Haoyang Cao, Samuel Cohen, and Lukasz Szpruch. 2021. Identifiability in inverse reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 12362–12373.

Victor Chernozhukov, Juan Carlos Escanciano, Hidehiko Ichimura, Whitney K Newey, and James M Robins. 2022. Locally robust semiparametric estimation. *Econometrica* 90, 4 (2022), 1501–1535.

Khai Xiang Chiong, Alfred Galichon, and Matt Shum. 2016. Duality in dynamic discrete-choice models. *Quantitative Economics* 7, 1 (2016), 83–115.

Richard Combes. 2024. An extension of mcdiarmid's inequality. In *2024 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 79–84.

Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. 2018. SBEED: Convergent reinforcement learning with nonlinear function approximation. In *International conference on machine learning*. PMLR, 1125–1134.

Stefano Ermon, Yexiang Xue, Russell Toth, Bistra Dilkina, Richard Bernstein, Theodoros Damoulas, Patrick Clark, Steve DeGloria, Andrew Mude, Christopher Barrett, et al. 2015. Learning large-scale dynamic discrete choice models of spatio-temporal preferences with application to migratory pastoralism in East Africa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

Yiding Feng, Ekaterina Khmelnitskaya, and Denis Nekipelov. 2020. Global concavity and optimization in a class of dynamic discrete choice models. In *International Conference on Machine Learning*. PMLR, 3082–3091.

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. 2016. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852* (2016).

William Fithian and Trevor Hastie. 2014. Local case-control sampling: Efficient subsampling in imbalanced data sets. *Annals of statistics* 42, 5 (2014), 1693.

Dylan J Foster, Akshay Krishnamurthy, David Simchi-Levi, and Yunzong Xu. 2021. Offline reinforcement learning: Fundamental barriers for value function approximation. *arXiv preprint arXiv:2111.10919* (2021).

Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248* (2017).

Bolin Gao and Lacra Pavel. 2017. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805* (2017).

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. 2021. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems* 34 (2021), 4028–4039.

Sinong Geng, Houssam Nassif, Carlos Manzanares, Max Reppen, and Ronnie Sircar. 2020. Deep PQR: Solving inverse reinforcement learning using anchor actions. In *International Conference on Machine Learning*. PMLR, 3431–3441.

Sinong Geng, Houssam Nassif, and Carlos A Manzanares. 2023. A data-driven state aggregation approach for dynamic discrete choice models. In *Uncertainty in Artificial Intelligence*. PMLR, 647–657.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

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

V Joseph Hotz and Robert A Miller. 1993. Conditional choice probabilities and the estimation of dynamic models. *The Review of Economic Studies* 60, 3 (1993), 497–529.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. 2017. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*. PMLR, 1704–1713.

Nan Jiang and Tengyang Xie. 2024. Offline Reinforcement Learning in Large State Spaces: Algorithms and Guarantees. *Statist. Sci.* (2024).

Kelvin Kan, James G. Nagy, and Lars Ruthotto. 2023. LSEMINK: A Modified Newton-Krylov Method for Log-Sum-Exp Minimization. arXiv:2307.04871 [math.OC] https://arxiv.org/abs/2307.04871

Hiroyuki Kasahara and Katsumi Shimotsu. 2009. Nonparametric identification of finite mixture models of dynamic discrete choices. *Econometrica* 77, 1 (2009), 135–175.

Trupti M Kodinariya, Prashant R Makwana, et al. 2013. Review on determining number of Cluster in K-Means Clustering. *International Journal* 1, 6 (2013), 90–95.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. 2019. Imitation learning via off-policy distribution matching. *arXiv preprint arXiv:1912.05032* (2019).

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

Feng-Yi Liao, Lijun Ding, and Yang Zheng. 2024. Error bounds, PL condition, and quadratic growth for weakly convex functions, and linear convergences of proximal point methods. In *6th Annual Learning for Dynamics & Control Conference*. PMLR, 993–1005.

Thierry Magnac and David Thesmar. 2002. Identifying dynamic discrete decision processes. *Econometrica* 70, 2 (2002), 801–816.

Daniel McFadden. 2001. Economic choices. *American economic review* 91, 3 (2001), 351–378.

Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, Vol. 99. 278–287.

Tianwei Ni, Harshit Sikchi, Yufei Wang, Tejus Gupta, Lisa Lee, and Ben Eysenbach. 2021. f-irl: Inverse reinforcement learning via state marginal matching. In *Conference on Robot Learning*. PMLR, 529–551.

Andriy Norets. 2009. Inference in dynamic discrete choice models with serially orrelated unobserved state variables. *Econometrica* 77, 5 (2009), 1665–1682.

Andriy Norets. 2012. Estimation of dynamic discrete choice models using artificial neural network approximations. *Econometric Reviews* 31, 1 (2012), 84–106.

Andrew Patterson, Adam White, and Martha White. 2022. A generalized projected bellman error for off-policy value estimation in reinforcement learning. *Journal of Machine Learning Research* 23, 145 (2022), 1–61.

Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. 2017. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems* 30 (2017).

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024. From r to $Q*$: Your Language Model is Secretly a Q-Function. *arXiv preprint arXiv:2404.12358* (2024).

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research* 22, 268 (2021), 1–8.

Quentin Rebjock and Nicolas Boumal. 2023. Fast convergence to non-isolated minima: four equivalent conditions for C^2 functions. *arXiv preprint arXiv:2303.00096* (2023).

Gregor Reich. 2018. Divide and conquer: recursive likelihood function integration for hidden Markov models with continuous latent variables. *Operations research* 66, 6 (2018), 1457–1470.

Mark Rudelson and Roman Vershynin. 2009. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 62, 12 (2009), 1707–1739.

John Rust. 1987. Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. *Econometrica: Journal of the Econometric Society* (1987), 999–1033.

John Rust. 1994. Structural estimation of Markov decision processes. *Handbook of econometrics* 4 (1994), 3081–3143.

Andrzej Ruszczyński and Shangzhe Yang. 2024. A Functional Model Method for Nonconvex Nonsmooth Conditional Stochastic Optimization. *SIAM Journal on Optimization* 34, 3 (2024), 3064–3087.

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

Kristina P Sinaga and Miin-Shen Yang. 2020. Unsupervised K-means clustering algorithm. *IEEE access* 8 (2020), 80716–80727.

Che-Lin Su and Kenneth L Judd. 2012. Constrained optimization approaches to estimation of structural models. *Econometrica* 80, 5 (2012), 2213–2230.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*.

Gerald Tesauro et al. 1995. Temporal difference learning and TD-Gammon. *Commun. ACM* 38, 3 (1995), 58–68.

Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954* (2018).

Ahmed Touati and Pascal Vincent. 2020. Sharp Analysis of Smoothed Bellman Error Embedding. *arXiv preprint arXiv:2007.03749* (2020).

John Tsitsiklis and Benjamin Van Roy. 1996. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems* 9 (1996).

Masatoshi Uehara, Jiawei Huang, and Nan Jiang. 2020. Minimax weight and q-function learning for off-policy evaluation. In *International Conference on Machine Learning*. PMLR, 9659–9668.

Ruosong Wang, Yifan Wu, Ruslan Salakhutdinov, and Sham Kakade. 2021. Instabilities of offline rl with pre-trained neural representation. In *International Conference on Machine Learning*. PMLR, 10948–10960.

Junchi Yang, Negar Kiyavash, and Niao He. 2020. Global convergence and variance-reduced optimization for a class of nonconvex-nonconcave minimax problems. *arXiv preprint arXiv:2002.09621* (2020).

Yingyao Hu Fangzhu Yang. 2024. Estimation of Dynamic Discrete Choice Models with Unobserved State Variables Using Reinforcement Learning. (2024).

Sheng Yue, Guanbo Wang, Wei Shao, Zhaofeng Zhang, Sen Lin, Ju Ren, and Junshan Zhang. 2023. Clare: Conservative model-based reward learning for offline inverse reinforcement learning. *arXiv preprint arXiv:2302.04782* (2023).

Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. 2023. Understanding expertise through demonstrations: A maximum likelihood framework for offline inverse reinforcement learning. *arXiv preprint arXiv:2302.07457* (2023).

Han Zhong, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. 2024. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922* (2024).

Li Ziniu, Xu Tian, Yu Yang, and Luo Zhi-Quan. 2022. Rethinking ValueDice - Does It Really Improve Performance?. In *ICLR Blog Track*. https://iclr-blog-track.github.io/2022/03/25/rethinking-valuedice/ https://iclr-blog-track.github.io/2022/03/25/rethinking-valuedice/.

Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyu Wang, Yusuf Aytar, Misha Denil, Nando de Freitas, and Scott Reed. 2020. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885* (2020).

A Extended experiment discussions

A.1 More discussions on Bus engine replacement experiments

Figure 2, 3 and Table 4 - 7 shown below present the estimated results for reward and Q^* using 50 trajectories (5,000 transitions) and 1,000 trajectories (100,000 transitions). As you can see in Figure 2 and 3, Rust and ML-IRL , which know the exact transition probabilities and employ correct parameterization (i.e., linear), demonstrate strong extrapolation capabilities for [Mileage, action] pairs that are rarely observed or entirely missing from the dataset (mileage 6-10). In contrast, GLADIUS, a non-parametric method, struggles with these underrepresented pairs.

However, as we saw in the main text's Table 2, GLADIUS achieves par or lower Mean Absolute Percentage Error (MAPE), which is defined as $\frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{r}_i - r_i}{r_i}\right| \times 100$ where N is the total number of samples from expert policy π^* and $\hat{r}_i$ is each algorithm's estimator for the true reward r_i. This is because it overall outperforms predicting r values for the [Mileage, action] pairs that appear most frequently and therefore contribute most significantly to the error calculation, as indicated by the visibility of the yellow shading in the tables below. (Higher visibility implies larger frequency.)

Results for 50 trajectories (absolute error plot, r prediction, Q^* prediction)

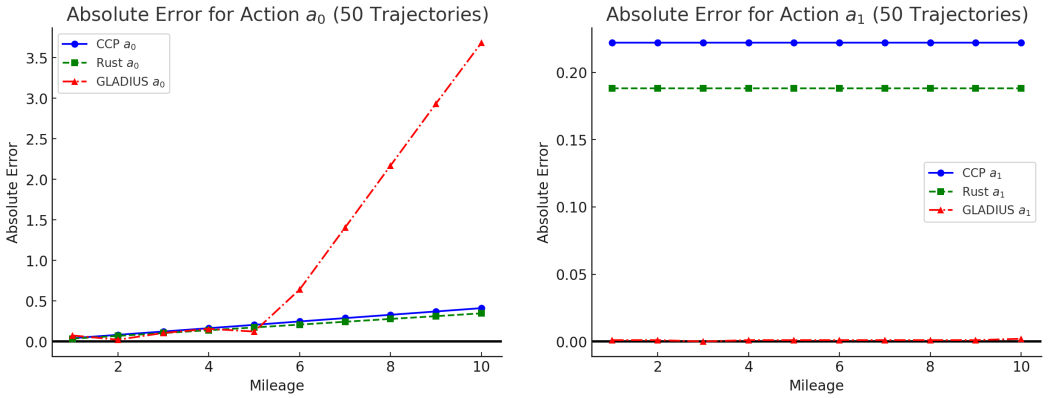

Fig. 2. Reward estimation error comparison for 50 trajectories. Closer to 0 (black line) is better.

Mileage	Frequency		Ground Truth r		ML-IRL		Rust		GLADIUS	
	a_0	a_1	a_0	a_1	a_0	a_1	a_0	a_1	a_0	a_1
1	412	37	-1.000	-5.000	-0.959	-4.777	-0.965	-4.812	-1.074	-4.999
2	65	18	-2.000	-5.000	-1.918	-4.777	-1.931	-4.812	-1.978	-5.001
3	43	80	-3.000	-5.000	-2.877	-4.777	-2.896	-4.812	-3.105	-5.000
4	24	101	-4.000	-5.000	-3.836	-4.777	-3.861	-4.812	-3.844	-5.001
5	8	134	-5.000	-5.000	-4.795	-4.777	-4.827	-4.812	-4.878	-5.001
6	4	37	-6.000	-5.000	-5.753	-4.777	-5.792	-4.812	-6.642	-5.001
7	1	26	-7.000	-5.000	-6.712	-4.777	-6.757	-4.812	-8.406	-5.001
8	0	7	-8.000	-5.000	-7.671	-4.777	-7.722	-4.812	-10.170	-5.001
9	0	2	-9.000	-5.000	-8.630	-4.777	-8.688	-4.812	-11.934	-5.001
10	0	1	-10.000	-5.000	-9.589	-4.777	-9.653	-4.812	-13.684	-5.002

Table 4. Reward estimation for 50 trajectories. Color indicates appearance frequencies.

Mileage	Frequency		Ground Truth Q		ML-IRL Q		Rust Q		GLADIUS Q	
	a_0	a_1	a_0	a_1	a_0	a_1	a_0	a_1	a_0	a_1
1	412	37	-52.534	-54.815	-49.916	-52.096	-50.327	-52.523	-53.059	-55.311
2	65	18	-53.834	-54.815	-51.165	-52.096	-51.584	-52.523	-54.270	-55.312
3	43	80	-54.977	-54.815	-52.266	-52.096	-52.691	-52.523	-55.548	-55.312
4	24	101	-56.037	-54.815	-53.286	-52.096	-53.718	-52.523	-56.356	-55.312
5	8	134	-57.060	-54.815	-54.270	-52.096	-54.708	-52.523	-57.419	-55.312
6	4	37	-58.069	-54.815	-55.239	-52.096	-55.683	-52.523	-59.184	-55.312
7	1	26	-59.072	-54.815	-56.202	-52.096	-56.652	-52.523	-60.950	-55.312
8	0	7	-60.074	-54.815	-57.162	-52.096	-57.619	-52.523	-62.715	-55.312
9	0	2	-61.074	-54.815	-58.122	-52.096	-58.585	-52.523	-64.481	-55.312
10	0	1	-62.074	-54.815	-59.081	-52.096	-59.550	-52.523	-66.228	-55.308

Table 5. Q^* estimation for 50 trajectories. Color indicates appearance frequencies.

Results for 1000 trajectories (absolute error plot, r prediction, Q^* prediction)

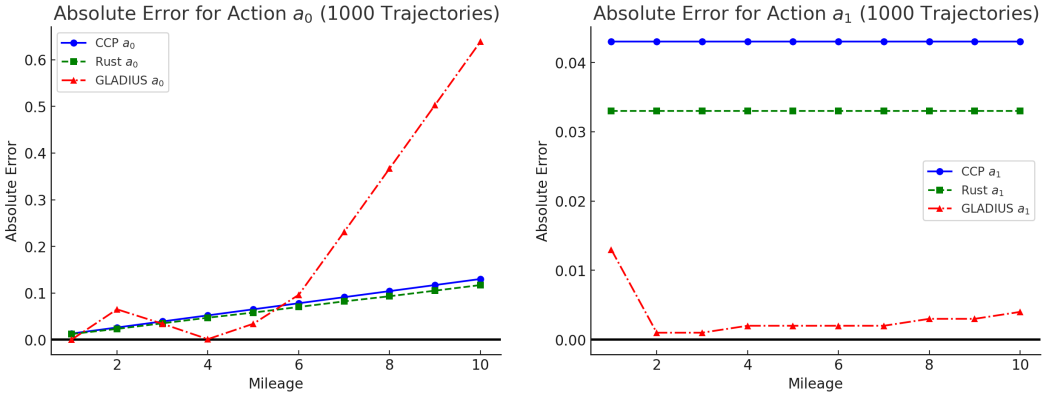

Fig. 3. Reward estimation error comparison for 1,000 trajectories. Closer to 0 (black line) is better.

Mileage	Frequency		Ground Truth r		ML-IRL		Rust		GLADIUS	
	a_0	a_1	a_0	a_1	a_0	a_1	a_0	a_1	a_0	a_1
1	7994	804	-1.000	-5.000	-1.013	-5.043	-1.012	-5.033	-1.000	-5.013
2	1409	541	-2.000	-5.000	-2.026	-5.043	-2.023	-5.033	-1.935	-5.001
3	1060	1296	-3.000	-5.000	-3.039	-5.043	-3.035	-5.033	-2.966	-5.000
4	543	1991	-4.000	-5.000	-4.052	-5.043	-4.047	-5.033	-3.998	-5.002
5	274	2435	-5.000	-5.000	-5.065	-5.043	-5.058	-5.033	-4.966	-5.002
6	35	829	-6.000	-5.000	-6.078	-5.043	-6.070	-5.033	-5.904	-5.002
7	8	476	-7.000	-5.000	-7.091	-5.043	-7.082	-5.033	-6.769	-5.002
8	1	218	-8.000	-5.000	-8.104	-5.043	-8.093	-5.033	-7.633	-5.003
9	0	73	-9.000	-5.000	-9.117	-5.043	-9.105	-5.033	-8.497	-5.003
10	0	10	-10.000	-5.000	-10.130	-5.043	-10.117	-5.033	-9.361	-5.004

Table 6. Reward estimation for 1,000 trajectories. Color indicates appearance frequencies.

Mileage	Frequency		Ground Truth Q		ML-IRL Q		Rust Q		GLADIUS Q	
	a_0	a_1	a_0	a_1	a_0	a_1	a_0	a_1	a_0	a_1
1	7994	804	-52.534	-54.815	-53.110	-55.405	-53.019	-55.309	-52.431	-54.733
2	1409	541	-53.834	-54.815	-54.423	-55.405	-54.330	-55.309	-53.680	-54.720
3	1060	1296	-54.977	-54.815	-55.578	-55.405	-55.483	-55.309	-54.852	-54.721
4	543	1991	-56.037	-54.815	-56.649	-55.405	-56.554	-55.309	-55.942	-54.721
5	274	2435	-57.060	-54.815	-57.684	-55.405	-57.588	-55.309	-56.932	-54.721
6	35	829	-58.069	-54.815	-58.705	-55.405	-58.608	-55.309	-57.886	-54.721
7	8	476	-59.072	-54.815	-59.721	-55.405	-59.623	-55.309	-58.745	-54.721
8	1	218	-60.074	-54.815	-60.735	-55.405	-60.636	-55.309	-59.604	-54.722
9	0	73	-61.074	-54.815	-61.748	-55.405	-61.648	-55.309	-60.463	-54.722
10	0	10	-62.074	-54.815	-62.760	-55.405	-62.660	-55.309	-61.322	-54.722

Table 7. Q^* estimation for 1,000 trajectories. Color indicates appearance frequencies.

A.2 Imitation learning experiments

In this section for completeness, we benchmark our method against other methods for *optimal policy learning*, i.e. imitation learning, not the more difficult problem of learning the true reward function.

The objective of imitation learning is often defined to as finding policy $\hat{p}$ with

$$\min_{\hat{p}} \mathbb{E}_{(s,a)\sim\pi^*} \left[\ell\left(\hat{p}(a \mid s), \pi^*(a \mid s)\right)\right], \ell \text{ is the cross-entropy loss}$$

or equivalently,

$$\min_{\hat{p}} \mathbb{E}_{(s,a)\sim\pi^*} \left[-\log \hat{p}(a|s)\right] \tag{11}$$

A.2.1 Behavioral Cloning (BC). Note that Equation (11) is exactly what a typical Behavioral Cloning (BC) algorithm [Torabi et al., 2018] minimizes under entropy regularization, as the objective of BC algorithm is

$$\min_{Q\in\mathcal{Q}} \mathbb{E}_{(s,a)\sim\pi^*} \left[-\log \hat{p}_Q(a|s)\right] \tag{12}$$

Furthermore, the solution set to Equation (12) includes the solution set to ERM-IRL objective (Theorem 4.1).

$$\min_{Q\in\mathcal{Q}} \mathbb{E}_{(s,a)\sim\pi^*} \left[-\log \hat{p}_Q(a|s) + \lambda \mathbb{1}_{a=a_s} \mathcal{L}_{BE}(Q)(s,a)\right] \qquad \text{(Equation (5))}$$

That is, ERM-IRL searches for the solution that satisfies the Bellman equation inside the solution set of BC. Therefore, we can expect that *the BC baseline will be computationally and statistically more efficient* than the ERM-IRL-based algorithm for Imitation learning tasks. (In other words, it is unreasonable to expect BC to perform well in reward estimation tasks.)

A.2.2 Experiments. As in Garg et al. [2021], we employ three OpenAI Gym environments for algorithms with discrete actions [Brockman, 2016] called Lunar Lander v2, Cartpole v1, and Acrobot v1. Here, we only include baselines that do not require the transition function estimation step and therefore excluded baselines such as Rust [Rust, 1987] and ML-IRL [Zeng et al., 2023].

Table 8 provides the baseline algorithms' performances measured in terms of *% optimality*, which is measured by dividing each episode's episodic reward by the mean of 1000 repetition of expert policy's episodic rewards. That is,

$$\%\text{ optimality of an episode} := \frac{\text{One episode's episodic reward of the algorithm}}{\text{Mean of 1,000 episodic rewards of the expert}} \times 100$$

In Table 8, We take the mean and variance of 100 episodic averages for each baseline. For expert policy, we used publicly available expert policy[14] trained via Proximal Policy Optimization (PPO) algorithm [Schulman et al., 2017] implementation available as part of stable-baselines3[Raffin et al., 2021] project. The 1000-episodic mean and standard deviation ([mean±std]) of the episodic reward of expert policy for each environment was $[232.77 \pm 73.77]$ for Lunar-Lander v2 (larger the better), $[-82.80 \pm 27.55]$ for Acrobot v1 (smaller the better), and $[500 \pm 0]$ for Cartpole v1 (larger the better).

Trajs	Lunar Lander v2 (%) (Larger % the better)			Cartpole v1 (%) (Larger % the better)			Acrobot v1 (%) (Smaller % the better)		
	Gladius	IQ-learn	BC	Gladius	IQ-learn	BC	Gladius	IQ-learn	BC
1	**107.30** (10.44)	83.78 (22.25)	103.38 (13.78)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	103.67 (32.78)	103.47 (55.44)	**100.56** (26.71)
3	**106.64** (11.11)	102.44 (20.66)	104.46 (11.57)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	102.19 (22.69)	101.28 (37.51)	**101.25** (36.42)
7	101.10 (16.28)	104.91 (13.98)	**105.99** (10.20)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	100.67 (22.30)	100.58 (30.09)	**98.08** (24.27)
10	104.46 (13.65)	105.13 (13.83)	**107.01** (10.75)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	99.07 (20.58)	101.10 (30.40)	**97.75** (16.67)
15	106.11 (10.65)	106.51 (14.10)	**107.42** (10.45)	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	96.50 (18.53)	95.34 (26.92)	**95.33** (15.42)

Based on 100 episodes for each baseline. Each baseline was trained for 5000 epochs.

Table 8. Mean and standard deviation of % optimality of 100 episodes

[14] https://huggingface.co/sb3/

B Technical Proofs

B.1 Theory of TD correction using biconjugate trick

PROOF OF LEMMA 4.2.

$$\begin{aligned}
&\mathcal{L}_{BE}(s,a)(Q) := \mathbb{E}_{s'\sim P(s,a)}\left[\delta_Q(s,a,s') \mid s,a\right]^2 \\
&= \max_{h\in\mathbb{R}} 2\cdot \mathbb{E}_{s'\sim P(s,a)}\left[\delta_Q(s,a,s') \mid s,a\right]\cdot h - h^2 && \text{(Biconjugate)} \\
&= \max_{h\in\mathbb{R}} 2\cdot \mathbb{E}_{s'\sim P(s,a)}\left[\hat{\mathcal{T}}Q - Q \mid s,a\right]\cdot \underbrace{h}_{=\rho - Q(s,a)} - h^2 \\
&= \max_{\rho(s,a)\in\mathbb{R}} \mathbb{E}_{s'\sim P(s,a)}\left[2\left(\hat{\mathcal{T}}Q - Q\right)(\rho - Q) - (\rho - Q)^2 \mid s,a\right] \\
&= \max_{\rho(s,a)\in\mathbb{R}} \mathbb{E}_{s'\sim P(s,a)}\left[\left(\hat{\mathcal{T}}Q - Q\right)^2 - \left(\hat{\mathcal{T}}Q - \rho\right)^2 \mid s,a\right] && (13)
\end{aligned}$$

where the unique maximum is with

$$\begin{aligned}
\rho^*(s,a) &= h^*(s,a) + Q(s,a) = \mathcal{T}Q(s,a) - Q(s,a) + Q(s,a) \\
&= \mathcal{T}Q(s,a)
\end{aligned}$$

and where the equality of 13 is from

$$\begin{aligned}
&2\left(\hat{\mathcal{T}}Q - Q\right)(\rho - Q) - (\rho - Q)^2 \\
&\quad = 2(\hat{\mathcal{T}}Q\rho - \hat{\mathcal{T}}QQ - \cancel{Q\rho} + Q^2) - (\rho^2 - \cancel{2Q\rho} + Q^2) \\
&\quad = 2\hat{\mathcal{T}}Q\rho - 2\hat{\mathcal{T}}QQ + \cancel{2}Q^2 - \rho^2 - \cancel{Q^2} \\
&\quad = \hat{\mathcal{T}}Q^2 - 2\hat{\mathcal{T}}QQ + Q^2 - \hat{\mathcal{T}}Q^2 + 2\hat{\mathcal{T}}Q\rho - \rho^2 \\
&\quad = \left(\hat{\mathcal{T}}Q - Q\right)^2 - \left(\hat{\mathcal{T}}Q - \rho\right)^2
\end{aligned}$$

Now note that

$$\begin{aligned}
&\mathcal{L}_{BE}(s,a)(Q) = \max_{\rho(s,a)\in\mathbb{R}} \mathbb{E}_{s'\sim P(s,a)}\left[\left(\hat{\mathcal{T}}Q - Q\right)^2 - \left(\hat{\mathcal{T}}Q - \rho\right)^2 \mid s,a\right]] && \text{(equation 13)} \\
&= \mathbb{E}_{s'\sim P(s,a)}\left[\left(\hat{\mathcal{T}}Q - Q\right)^2 \mid s,a\right] - \min_{\rho(s,a)\in\mathbb{R}} \mathbb{E}_{s'\sim P(s,a)}\left[\left(\hat{\mathcal{T}}Q - \underbrace{\rho}_{=r+\beta\zeta}\right)^2 \mid s,a\right] \\
&= \mathbb{E}_{s'\sim P(s,a)}\left[\mathcal{L}_{TD}(Q)(s,a,s')\right] - \beta^2 \min_{\zeta\in\mathbb{R}} \mathbb{E}_{s'\sim P(s,a)}\left[\left(\hat{V}(s') - \zeta\right)^2 \mid s,a\right] && (14) \\
&= \mathbb{E}_{s'\sim P(s,a)}\left[\mathcal{L}_{TD}(Q)(s,a,s')\right] - \beta^2 \mathbb{E}_{s'\sim P(s,a)}\left[\left(\hat{V}(s') - \mathbb{E}_{s'\sim P(s,a)}[\hat{V}(s') \mid s,a]\right)^2 \mid s,a\right] && (15)
\end{aligned}$$

where the equality of equation 15 comes from the fact that the ζ that maximize equation 14 is $\zeta^* := \mathbb{E}_{s'\sim P(s,a)}[\hat{V}(s') \mid s,a]$, because

$$\begin{aligned}
r(s,a) + \beta\cdot\zeta^*(s,a) &:= \rho^*(s,a) \\
&= \mathcal{T}Q(s,a) \\
&:= r(s,a) + \beta\cdot\mathbb{E}_{s'\sim P(s,a)}\left[\hat{V}(s') \mid s,a\right]
\end{aligned}$$

For Q^*, $\mathcal{T}Q^* = Q^*$ holds. Therefore, we get

$$\begin{aligned} r(s,a) + \beta \cdot \zeta^*(s,a) &:= \rho^*(s,a) \\ &= \mathcal{T}Q^*(s,a) = Q^*(s,a) \end{aligned}$$

□

B.2 Proof of Theorem 3.1

Proof. Suppose that the system of equations (Equation 4)

$$\begin{cases} \dfrac{\exp(Q(s,a))}{\sum_{a' \in \mathcal{A}} \exp(Q(s,a'))} = \pi^*(a \mid s) \quad \forall s \in \mathcal{S}, a \in \mathcal{A} \\ r(s,a_s) + \beta \cdot \mathbb{E}_{s' \sim P(s,a_s)} \left[\log(\sum_{a' \in \mathcal{A}} \exp Q(s',a')) \mid s, a_s \right] - Q(s,a_s) = 0 \quad \forall s \in \mathcal{S} \end{cases}$$

is satisfied for $Q \in \mathcal{Q}$, where $\mathcal{Q}$ denote the space of all Q functions. Then we have the following equivalent recharacterization of the second condition $\forall s \in \mathcal{S}$,

$$\begin{aligned} Q(s,a_s) &= r(s,a_s) + \beta \cdot \mathbb{E}_{s' \sim P(s,a_s)} \left[\log(\sum_{a' \in \mathcal{A}} \exp Q(s',a')) \mid s, a_s \right] \\ &= r(s,a_s) + \beta \cdot \mathbb{E}_{s' \sim P(s,a_s)} \left[Q(s',a') - \log \pi^*(a' \mid s') \mid s, a_s \right] \quad \forall a' \in \mathcal{A} \\ &= r(s,a_s) + \beta \cdot \mathbb{E}_{s' \sim P(s,a_s)} \left[Q(s',a_{s'}) - \log \pi^*(a_{s'} \mid s') \mid s, a_s \right] \end{aligned} \tag{16}$$

We will now show the existence and uniqueness of a solution using a standard fixed point argument on a Bellman operator. Let $\mathcal{F}$ be the space of functions $f : \mathcal{S} \to \mathbb{R}$ induced by elements of $\mathcal{Q}$, where each $Q \in \mathcal{Q}$ defines an element of $\mathcal{F}$ via

$$f_Q(s) := Q(s,a_s)$$

and define an operator $\mathcal{T}_f : \mathcal{F} \to \mathcal{F}$ that acts on functions f_Q :

$$\left(\mathcal{T}_f f_Q\right)(s) := r(s,a_s) + \beta \sum_{s'} P(s' \mid s,a_s) \left[f_Q(s') - \log \pi^*(a_{s'} \mid s') \right]$$

Then for $Q_1, Q_2 \in \mathcal{Q}$, We have

$$\left(\mathcal{T}_f f_{Q_1}\right)(s) := r(s,a_s) + \beta \sum_{s'} P(s' \mid s,a_s) \left[f_{Q_1}(s') - \log \pi^*(a_{s'} \mid s') \right]$$

$$\left(\mathcal{T}_f f_{Q_2}\right)(s) := r(s,a_s) + \beta \sum_{s'} P(s' \mid s,a_s) \left[f_{Q_2}(s') - \log \pi^*(a_{s'} \mid s') \right]$$

Substracting the two, we get

$$\begin{aligned} \left| \left(\mathcal{T}_f f_{Q_1}\right)(s) - \left(\mathcal{T}_f f_{Q_2}\right)(s) \right| &\le \beta \sum_{s'} P(s' \mid s,a_s) \left| f_{Q_1}(s') - f_{Q_2}(s') \right| \\ &\le \beta \left\| f_{Q_1} - f_{Q_2} \right\|_\infty \end{aligned}$$

Taking supremum norm over $s \in \mathcal{S}$, we get

$$\left\| \mathcal{T}_f f_{Q_1} - \mathcal{T}_f f_{Q_2} \right\|_\infty \le \beta \left\| f_{Q_1} - f_{Q_2} \right\|_\infty$$

This implies that $\mathcal{T}_f$ is a contraction mapping under supremum norm, with $\beta \in (0,1)$. Since $\mathcal{Q}$ is a Banach space under sup norm (Lemma B.1), we can apply Banach fixed point theorem to show that

there exists a unique f_Q that satisfies $\mathcal{T}_f(f_Q) = f_Q$, and by definition of f_Q there exists a unique Q that satisfies $\mathcal{T}_f(f_Q) = f_Q$, i.e.,

$$r(s, a_s) + \beta \cdot \mathbb{E}_{s' \sim P(s,a_s)} \left[\log \left(\sum_{a' \in \mathcal{A}} \exp Q(s', a') \right) \mid s, a_s \right] - Q(s, a_s) = 0 \quad \forall s \in \mathcal{S}$$

Since Q^* satisfies the system of equations 4, Q^* is the only solution to the system of equations.

Also, since $Q^* = \mathcal{T}Q^* = r(s,a) + \beta \cdot \mathbb{E}_{s' \sim P(s,a)} \left[\log(\sum_{a' \in \mathcal{A}} \exp Q^*(s', a')) \mid s, a \right]$ holds, we can identify r as

$$r(s,a) = Q^*(s,a) - \beta \cdot \mathbb{E}_{s' \sim P(s,a)} \left[\log(\sum_{a' \in \mathcal{A}} \exp Q^*(s', a')) \mid s, a \right]$$

□

LEMMA B.1. *Suppose that $\mathcal{Q}$ consists of bounded functions on $\mathcal{S} \times \mathcal{A}$. Then $\mathcal{Q}$ is a Banach space with supremum norm as the induced norm.*

PROOF. Suppose that a sequence of functions $\{Q_n\}$ in $\mathcal{Q}$ is Cauchy in the supremum norm. We need to show that $Q_n \to Q^*$ as $n \to \infty$ for some Q^* and Q^* is also bounded. Note that Q_n being Cauchy in sup norm implies that for every (s, a), the sequence $\{Q_n(s, a)\}$ is Cauchy in $\mathbb{R}$. Since $\mathbb{R}$ is a complete space, every Cauchy sequence of real numbers has a limit; this allows us to define function $Q^* : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ such that $Q^*(s, a) = \lim_{n \to \infty} Q_n(s, a)$. Then we can say that $Q_n(s, a) \to Q^*(s, a)$ for every $(s, a) \in \mathcal{S} \times \mathcal{A}$. Since each Q_n is bounded, we take the limit and obtain:

$$\sup_{s,a} |Q^*(s, a)| = \lim_{n \to \infty} \sup_{s,a} |Q_n(s, a)| \leq M$$

which implies $Q^* \in \mathcal{Q}$.

Now what's left is to show that the supremum norm

$$\|Q\|_\infty = \sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} |Q(s, a)|$$

induces the metric, i.e.,

$$d(Q_1, Q_2) := \|Q_1 - Q_2\|_\infty = \sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} |Q_1(s, a) - Q_2(s, a)|$$

The function d satisfies the properties of a metric:

- Non-negativity: $d(Q_1, Q_2) \geq 0$ and $d(Q_1, Q_2) = 0$ if and only if $Q_1 = Q_2$.
- Symmetry: $d(Q_1, Q_2) = d(Q_2, Q_1)$ by the absolute difference.
- Triangle inequality:

$$d(Q_1, Q_3) = \sup_{s,a} |Q_1(s, a) - Q_3(s, a)| \leq \sup_{s,a} |Q_1(s, a) - Q_2(s, a)| + \sup_{s,a} |Q_2(s, a) - Q_3(s, a)|$$

which shows $d(Q_1, Q_3) \leq d(Q_1, Q_2) + d(Q_2, Q_3)$.

□

B.3 Proof of Theorem 4.1

Define $\hat{Q}$ as

$$\hat{Q} \in \arg\min_{Q \in \mathcal{Q}} \ \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[-\log \left(\hat{p}_Q(a \mid s) \right) \right] + \lambda \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[\mathbb{1}_{a = a_s} \mathcal{L}_{BE}(Q)(s, a) \right] \qquad \text{(Equation 5)}$$

From Theorem 3.1, it is sufficient to show that $\hat{Q}$ satisfies the equations 4 of Theorem 3.1 for any $\lambda > 0$, i.e.,

$$\begin{cases} \dfrac{\exp(\hat{Q}(s,a))}{\sum_{a' \in \mathcal{A}} \exp(\hat{Q}(s,a'))} = \pi^*(a \mid s) \quad \forall s \in \bar{\mathcal{S}}, a \in \mathcal{A} \\ r(s,a_s) + \beta \cdot \mathbb{E}_{s' \sim P(s,a_s)} \left[\log(\sum_{a' \in \mathcal{A}} \exp \hat{Q}(s',a')) \mid s, a_s \right] - \hat{Q}(s,a_s) = 0 \quad \forall s \in \bar{\mathcal{S}} \end{cases} \tag{Equation 4}$$

where $\bar{\mathcal{S}}$ (the reachable states from ν_0, π^*) was defined as:

$$\bar{\mathcal{S}} = \{s \in \mathcal{S} \mid \Pr(s_t = s \mid s_0 \sim \nu_0, \pi^*) > 0 \text{ for some } t \geq 0\}$$

Now note that:

$$\begin{aligned} &\left\{Q \in \mathcal{Q} \mid \hat{p}_Q(\cdot \mid s) = \pi^*(\cdot \mid s) \quad \forall s \in \bar{\mathcal{S}} \quad \text{a.e.}\right\} \\ &= \arg\max_{Q \in \mathcal{Q}} \ \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[\log \left(\hat{p}_Q(\cdot \mid s) \right) \right] \qquad (\because \text{Lemma B.2}) \\ &= \arg\min_{Q \in \mathcal{Q}} \ \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[-\log \left(\hat{p}_Q(\cdot \mid s) \right) \right] \end{aligned}$$

and

$$\begin{aligned} &\left\{Q \in \mathcal{Q} \mid \mathcal{L}_{BE}(Q)(s,a_s) = 0 \quad \forall s \in \bar{\mathcal{S}}\right\} \\ &= \arg\min_{Q \in \mathcal{Q}} \ \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[\mathbb{1}_{a=a_s} \mathcal{L}_{BE}(Q)(s,a) \right] \end{aligned}$$

Therefore what we want to prove, equations 4, becomes the following equation 17:

$$\begin{cases} \hat{Q} \in \arg\min_{Q \in \mathcal{Q}} \ \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[-\log \left(\hat{p}_Q(\cdot \mid s) \right) \right] \\ \hat{Q} \in \arg\min_{Q \in \mathcal{Q}} \ \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[\mathbb{1}_{a=a_s} \mathcal{L}_{BE}(Q)(s,a) \right] \end{cases} \tag{17}$$

where its solution set is nonempty by Theorem 3.1, i.e.,

$$\arg\min_{Q \in \mathcal{Q}} \ \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[-\log \left(\hat{p}_Q(a \mid s) \right) \right] \ \cap \ \arg\min_{Q \in \mathcal{Q}} \ \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[\mathbb{1}_{a=a_s} \mathcal{L}_{BE}(\hat{Q})(s,a) \right] \ \neq \ \emptyset$$

Under this non-emptiness, according to Lemma B.3, $\hat{Q}$ satisfies equation 17. This implies that $\hat{Q}(s,a) = Q^*(s,a)$ for $s \in \bar{\mathcal{S}}$ and $a \in \mathcal{A}$, as the solution to set of equations 4 is Q^*. This implies that

$$r(s,a) = \hat{Q}(s,a) - \beta \cdot \mathbb{E}_{s' \sim P(s,a)} \left[\log \left(\sum_{a' \in \mathcal{A}} \exp \hat{Q}(s',a') \right) \mid s, a \right]$$

for $s \in \bar{\mathcal{S}}$ and $a \in \mathcal{A}$. □

LEMMA B.2.

$$\begin{aligned} &\arg\max_{Q \in \mathcal{Q}} \ \mathbb{E}_{(s,a) \sim \pi^*, \nu_0} \left[\log \left(\hat{p}_Q(\cdot \mid s) \right) \right] \\ &\quad = \left\{Q \in \mathcal{Q} \mid \hat{p}_Q(\cdot \mid s) = \pi^*(\cdot \mid s) \quad \forall s \in \bar{\mathcal{S}} \quad a.e.\right\} \\ &\quad = \left\{Q \in \mathcal{Q} \mid Q(s,a_1) - Q(s,a_2) = Q^*(s,a_1) - Q^*(s,a_2) \quad \forall a_1, a_2 \in \mathcal{A}, s \in \bar{\mathcal{S}}\right\} \end{aligned}$$

PROOF OF LEMMA B.2.

$$\begin{aligned}\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\log\left(\hat{p}_Q(\cdot\mid s)\right)\right] &= \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}[\log\hat{p}_Q(a|s) - \ln\pi^*(a|s) + \ln\pi^*(a|s)] \\ &= -\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\ln\frac{\pi^*(a|s)}{\hat{p}_Q(a|s)}\right] + \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}[\ln\pi^*(a|s)] \\ &= -\mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}(\pi^*(\cdot\mid s)\|\hat{p}_Q(\cdot\mid s))\right] + \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}[\ln\pi^*(a|s)]\end{aligned}$$

Therefore,

$$\begin{aligned}&\arg\max_{Q\in\mathcal{Q}}\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\log\left(\hat{p}_Q(\cdot\mid s)\right)\right] = \arg\min_{Q\in\mathcal{Q}}\mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}(\pi^*(\cdot\mid s)\|\hat{p}_Q(\cdot\mid s))\right] \\ &= \{Q\in\mathcal{Q}\mid D_{KL}(\pi^*(\cdot\mid s)\|\hat{p}_Q(\cdot\mid s)) = 0 \text{ for all } s\in\bar{\mathcal{S}}\} \\ &\qquad\qquad (\because\ Q^*\in\mathcal{Q} \text{ and } D_{KL}(\pi^*\|\pi^*) = 0) \\ &= \{Q\in\mathcal{Q}\mid \hat{p}_Q(\cdot\mid s) = \pi^*(\cdot\mid s) \text{ a.e. for all } s\in\bar{\mathcal{S}}\} \\ &= \{Q\in\mathcal{Q}\mid \frac{\hat{p}_Q(a_1\mid s)}{\hat{p}_Q(a_2\mid s)} = \frac{\pi^*(a_1\mid s)}{\pi^*(a_2\mid s)} \quad \forall a_1,a_2\in\mathcal{A}, s\in\bar{\mathcal{S}}\} \\ &= \left\{Q\in\mathcal{Q}\mid \exp(Q(s,a_1)-Q(s,a_2)) = \exp\left(Q^*(s,a_1)-Q^*(s,a_2)\right) \quad \forall a_1,a_2\in\mathcal{A}, s\in\bar{\mathcal{S}}\right\} \\ &= \left\{Q\in\mathcal{Q}\mid Q(s,a_1)-Q(s,a_2) = Q^*(s,a_1)-Q^*(s,a_2) \quad \forall a_1,a_2\in\mathcal{A}, s\in\bar{\mathcal{S}}\right\}\end{aligned}$$

□

LEMMA B.3. *Let $f_1:\mathcal{X}\to\mathbb{R}$ and $f_2:\mathcal{X}\to\mathbb{R}$ be two functions defined on a common domain $\mathcal{X}$. Suppose the sets of minimizers of f_1 and f_2 intersect, i.e.,*

$$\arg\min f_1 \cap \arg\min f_2 \neq \emptyset$$

Then, any minimizer of the sum f_1+f_2 is also a minimizer of both f_1 and f_2 individually. That is, if

$$x^*\in\arg\min(f_1+f_2)$$

then

$$x^*\in\arg\min f_1 \ \cap\ \arg\min f_2$$

PROOF. Since $\arg\min f_1\cap\arg\min f_2\neq\emptyset$, let $x^\dagger$ be a common minimizer such that

$$x^\dagger\in\arg\min f_1\cap\arg\min f_2$$

This implies that

$$\begin{aligned}f_1(x^\dagger) &= \min_{x\in\mathcal{X}} f_1(x) =: m_1, \\ f_2(x^\dagger) &= \min_{x\in\mathcal{X}} f_2(x) =: m_2.\end{aligned}$$

Now, let x^* be any minimizer of f_1+f_2, so

$$x^*\in\arg\min(f_1+f_2) \iff f_1(x^*)+f_2(x^*)\le f_1(x)+f_2(x), \quad \forall x\in\mathcal{X}.$$

Evaluating this at $x^\dagger$, we obtain

$$\begin{aligned}f_1(x^*)+f_2(x^*) &\le f_1(x^\dagger)+f_2(x^\dagger) \\ &= m_1+m_2.\end{aligned}$$

Now, suppose for contradiction that $x^*\notin\arg\min f_1$, meaning

$$f_1(x^*) > m_1$$

But then

$$\begin{aligned} f_2(x^*) &\leq m_1 + m_2 - f_1(x^*) \\ &< m_1 + m_2 - m_1 = m_2 \end{aligned}$$

This contradicts the fact that $m_2 = \min f_2$, so x^* must satisfy

$$f_1(x^*) = m_1$$

By symmetry, assuming $x^* \notin \arg\min f_2$ leads to the same contradiction, forcing

$$f_2(x^*) = m_2$$

Thus, we conclude

$$x^* \in \arg\min f_1 \cap \arg\min f_2$$

□

B.4 Proof of Lemma 6.1

PROOF OF LEMMA 6.1. Denote $Q(s,\cdot) = [Q(s,a')]_{a' \in \mathcal{A}}$. Then,

Convexity of $\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[-\log\left(\hat{p}_Q(\cdot \mid s)\right)\right]$ w.r.t. $Q \in \mathcal{Q}$

$\iff$ Concavity of $\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\ln \hat{p}_Q(\cdot \mid s)\right]$ w.r.t. $Q \in \mathcal{Q}$

$\impliedby$ Concavity of $\ln \hat{p}_Q(\cdot \mid s)$ w.r.t. $Q \in \mathcal{Q}$ for all $s \in \mathcal{S}$ ($\because$ linearity of expectation)

$\iff$ Concavity of $Q(s,\cdot) - log \sum_{a' \in \mathcal{A}} \exp(Q(s,a'))$ w.r.t. $Q(s,\cdot)$ for all $s \in \mathcal{S}$

$\iff$ Convexity of $log \sum_{a' \in \mathcal{A}} \exp(Q(s,a'))$ w.r.t. $Q(s,\cdot)$ for all $s \in \mathcal{S}$

Since the function logsumexp is a known convex function, we are done.

Lipschitz smoothness of $\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\log\left(\hat{p}_Q(\cdot \mid s)\right)\right]$ w.r.t. $Q \in \mathcal{Q}$

$\iff$ Lipschitz continuity of $\nabla_Q \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\ln \hat{p}_Q(\cdot \mid s)\right]$ w.r.t. $Q \in \mathcal{Q}$

$\iff$ Lipschitz continuity of $\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\delta_{a,a'} - \hat{p}_Q(a' \mid s)\right]_{a' \in \mathcal{A}}$

$\iff$ Lipschitz continuity of $\mathbb{E}_{s\sim\pi^*,\nu_0}\left[\pi^*(a' \mid s) - \hat{p}_Q(a' \mid s)\right]_{a' \in \mathcal{A}}$

$\iff$ $\exists\, c > 0$ *s.t.* $\|\mathbb{E}_{s\sim\pi^*,\nu_0}\left[\hat{p}_{Q'}(a' \mid s) - \hat{p}_Q(a' \mid s)\right]_{a' \in \mathcal{A}}\| \leq c\|Q - Q'\|_{L_2(\pi^*,\nu_0)} \quad \forall Q, Q' \in \mathcal{Q}$

Since softmax is 1-Lipschitz continuous for each $s \in \mathcal{S}$ with respect to ℓ_2 norm [Gao and Pavel, 2017], for all $s \in \mathcal{S}$ we have

$$\left\|\hat{p}_{Q'}(\cdot \mid s) - \hat{p}_Q(\cdot \mid s)\right\|_2 \leq \|Q'(s,\cdot) - Q(s,\cdot)\|_2$$

Therefore

$$\begin{aligned} \|\mathbb{E}_{s\sim\pi^*,\nu_0}\left[\hat{p}_{Q'}(\cdot \mid s) - \hat{p}_Q(\cdot \mid s)\right]\|_2 &\leq \mathbb{E}_{s\sim\pi^*,\nu_0}\left[\left\|\hat{p}_{Q'}(\cdot \mid s) - \hat{p}_Q(\cdot \mid s)\right\|_2\right] && \text{(Norm is convex)} \\ &\leq \mathbb{E}_{s\sim\pi^*,\nu_0}\left[\|Q'(s,\cdot) - Q(s,\cdot)\|_2\right] && \text{(Softmax is 1-Lipchitz)} \\ &\leq \left(\mathbb{E}_{s\sim\pi^*,\nu_0}\|Q'(s,\cdot) - Q(s,\cdot)\|_2^2\right)^{1/2} && (x^{1/2} \text{ is concave}) \\ &= \|Q - Q'\|_{L_2(\pi^*,\nu_0)} \end{aligned}$$

□

B.5 Proof of Lemma 6.2 (Properties of Bellman error)

For showing that $\overline{\mathcal{L}_{BE}}(Q)$ is of C^2 w.r.t. $Q \in \mathcal{Q}$,

$$
\begin{aligned}
&C^2 \text{ of } \overline{\mathcal{L}_{BE}}(Q) \text{ w.r.t. } Q \in \mathcal{Q} \\
&\Longleftarrow C^2 \text{ of } Q(s,a) - \left[R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|\cdot s,a)} \log \sum_{a'} \exp\left(Q\left(s', a'\right)\right) \right] \text{ w.r.t. } Q \in \mathcal{Q} \text{ for } s \in \mathcal{S} \\
&\Longleftarrow C^2 \text{ of } \log \sum_{a'} \exp\left(Q\left(s, a'\right)\right) \text{ w.r.t. } Q \in \mathcal{Q} \text{ for } s \in \mathcal{S}
\end{aligned}
$$

As it is known that logsumexp is of C^2 [Kan et al., 2023], we are done.

For Lipchitz smoothness,

$$
\begin{aligned}
&\text{Lipschitz smoothness of } \overline{\mathcal{L}_{BE}}(Q) \text{ w.r.t. } Q \in \mathcal{Q} \\
&\iff \text{Lipschitz continuity of } \nabla_Q \overline{\mathcal{L}_{BE}}(Q) \text{ w.r.t. } Q \in \mathcal{Q} \\
&\iff \text{Lipschitz continuity of } \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[2\delta_Q(s,a)\nabla_Q\delta_Q(s,a)\right] \text{ w.r.t. } Q \in \mathcal{Q}
\end{aligned}
$$

Now note that

$$
\begin{aligned}
&\|\mathbb{E}_{s,a\sim\pi^*,\nu_0}\left[2\delta_Q(s,a)\nabla_Q\delta_Q(s,a) - 2\delta_{Q'}(s,a)\nabla_{Q'}\delta_{Q'}(s,a)\right]\|_2 \\
&\le \mathbb{E}_{s,a\sim\pi^*,\nu_0}\left[\left\|2\delta_Q(s,a)\nabla_Q\delta_Q(s,a) - 2\delta_{Q'}(s,a)\nabla_{Q'}\delta_{Q'}(s,a)\right\|_2\right] && \text{(Norm is convex)} \\
&\le \mathbb{E}_{s,a\sim\pi^*,\nu_0}\left[\|Q'(s,a) - Q(s,a)\|_2\right] && \text{(Lemma B.4)} \\
&\le \left(\mathbb{E}_{s\sim\pi^*,\nu_0}\|Q'(s,a) - Q(s,a)\|_2^2\right)^{1/2} && (x^{1/2} \text{ is concave}) \\
&= \|Q - Q'\|_{L_2(\pi^*,\nu_0)}
\end{aligned}
$$

This proves Lipschitz continuity of $\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[2\delta_Q(s,a)\nabla_Q\delta_Q(s,a)\right]$ w.r.t. $Q \in \mathcal{Q}$. □

LEMMA B.4 ($\delta_Q(s,a)\nabla_Q\delta_Q(s,a)$ IS LIPCHITZ). *For given fixed* (s,a),

$$
\left\|2\delta_Q(s,a)\nabla_Q\delta_Q(s,a) - 2\delta_{Q'}(s,a)\nabla_{Q'}\delta_{Q'}(s,a)\right\|_2 \le \|Q'(s,a) - Q(s,a)\|_2
$$

holds for any $Q, Q' \in \mathcal{Q}$.

PROOF OF LEMMA B.4. Note that

$$
\begin{aligned}
&\left\|\delta_Q(s,a)\nabla_Q\delta_Q(s,a) - \delta_{Q'}(s,a)\nabla_{Q'}\delta_{Q'}(s,a)\right\|_2 \\
&\le \left\|\delta_Q(s,a)\right\|_2 \left\|\nabla_Q\delta_Q(s,a) - \nabla_{Q'}\delta_{Q'}(s,a)\right\|_2 + \left\|\delta_Q(s,a) - \delta_{Q'}(s,a)\right\|_2 \left\|\nabla_{Q'}\delta_{Q'}(s,a)\right\|_2
\end{aligned}
$$

Now what's left is to prove that for given fixed (s,a),

(1) $\left\|\delta_Q(s,a)\right\|_2$ is bounded
(2) $\left\|\nabla_{Q'}\delta_{Q'}(s,a)\right\|_2$ is bounded
(3) $\delta_Q(s,a)$ is Lipschitz in $Q(s,a)$
(4) $\nabla_{Q'}\delta_{Q'}(s,a)$ is Lipchitz in $Q(s,a)$

(1) Boundedness of $\delta_Q(s,a)$:

$$
\begin{aligned}
|\delta_Q(s,a)| &= |\mathcal{T}Q(s,a) - Q(s,a)| \\
&= \left|r(s,a) + \beta\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V_Q(s')\right] - Q(s,a)\right|.
\end{aligned}
$$

Since $V_Q(s') = \ln \sum_{b \in \mathcal{A}} \exp(Q(s', b))$, we use the bound:

$$\max_{b \in \mathcal{A}} Q(s', b) \leq V_Q(s') \leq \max_{b \in \mathcal{A}} Q(s', b) + \ln |\mathcal{A}|$$

Taking expectations preserves boundedness, so we conclude:

$$|\delta_Q(s, a)| \leq |r(s, a)| + \beta \max_{s' \in \mathcal{S}} \max_{b \in \mathcal{A}} |Q(s', b)| + \beta \ln |\mathcal{A}| + \max_{s,a} |Q(s, a)|$$

This shows $\delta_Q(s, a)$ is uniformly bounded as long as Q is bounded, which is assured by $\beta < 1$.
(2) Boundedness of $\nabla_Q \delta_Q(s, a)$: The gradient is given by:

$$\nabla_Q \delta_Q(s, a) = \nabla_Q \mathcal{T} Q(s, a) - e_{(s,a)}$$

where

$$\nabla_Q \mathcal{T} Q(s, a) = \beta \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\nabla_Q V_Q(s') \right]$$

Since the softmax function $\nabla_Q V_Q(s')$ satisfies

$$\sum_{b \in \mathcal{A}} \text{softmax}(s', b; Q) = 1, \quad 0 \leq \text{softmax}(s', b; Q) \leq 1$$

we obtain:

$$\|\nabla_Q \mathcal{T} Q(s, a)\|_2 \leq \beta$$

Thus,

$$\|\nabla_Q \delta_Q(s, a)\|_2 = \|\nabla_Q \mathcal{T} Q(s, a) - e_{(s,a)}\|_2 \leq \beta + 1$$

Hence, $\nabla_Q \delta_Q(s, a)$ is bounded.
(3) Lipschitz continuity of $\delta_Q(s, a)$: Consider two functions Q and Q', and their corresponding Bellman errors:

$$\begin{aligned} \left|\delta_Q(s, a) - \delta_{Q'}(s, a)\right| &= |\mathcal{T} Q(s, a) - Q(s, a) - \mathcal{T} Q'(s, a) + Q'(s, a)| \\ &= |\mathcal{T} Q(s, a) - \mathcal{T} Q'(s, a) - (Q(s, a) - Q'(s, a))| \\ &\leq |\mathcal{T} Q(s, a) - \mathcal{T} Q'(s, a)| + |Q(s, a) - Q'(s, a)| \end{aligned}$$

Since $\mathcal{T} Q(s, a)$ depends on Q only through $V_Q(s')$, we use the Lipschitz property of log-sum-exp:

$$|V_Q(s') - V_{Q'}(s')| \leq \max_{b \in \mathcal{A}} |Q(s', b) - Q'(s', b)|$$

Taking expectations, we get:

$$|\mathcal{T} Q(s, a) - \mathcal{T} Q'(s, a)| \leq \beta \max_{s', b} |Q(s', b) - Q'(s', b)|$$

Therefore,

$$|\delta_Q(s, a) - \delta_{Q'}(s, a)| \leq (1 + \beta) \max_{s', b} |Q(s', b) - Q'(s', b)|$$

This proves $\delta_Q(s, a)$ is Lipschitz in $Q(s, a)$ with Lipschitz constant $1 + \beta$.
(4) Lipschitz continuity of $\nabla_Q \delta_Q(s, a)$: From the expression:

$$\nabla_Q \delta_Q(s, a) = \nabla_Q \mathcal{T} Q(s, a) - e_{(s,a)}$$

we focus on $\nabla_Q \mathcal{T} Q(s, a)$, which satisfies:

$$\|\nabla_Q \mathcal{T} Q(s, a) - \nabla_Q \mathcal{T} Q'(s, a)\|_2 = \left\| \beta \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\nabla_Q V_Q(s') - \nabla_Q V_{Q'}(s') \right] \right\|_2$$

Using the Lipschitz property of softmax,

$$\|\nabla_Q V_Q(s') - \nabla_Q V_{Q'}(s')\|_2 \leq \|Q(s', \cdot) - Q'(s', \cdot)\|_2$$

Taking expectations, we get:

$$\|\nabla_Q \mathcal{T} Q(s,a) - \nabla_Q \mathcal{T} Q'(s,a)\|_2 \leq \beta \max_{s',b} |Q(s',b) - Q'(s',b)|$$

Since

$$\|\nabla_Q \delta_Q(s,a) - \nabla_{Q'} \delta_{Q'}(s,a)\|_2 \leq \|\nabla_Q \mathcal{T} Q(s,a) - \nabla_Q \mathcal{T} Q'(s,a)\|_2$$

we conclude that $\nabla_Q \delta_Q(s,a)$ is Lipschitz with constant at most β.

□

B.6 Proof of Theorem 6.3 (Bellman error satisfying the PL condition)

By Lemma B.5 (Below), $\mathcal{L}_{BE}(Q)(s,a)$ satisfies PL condition with respect to Q for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Now we would like to show that $\overline{\mathcal{L}_{\text{BE}}}(Q) \coloneqq \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}[\mathcal{L}_{\text{BE}}(Q)(s,a)]$ is also PL in terms of $L^2(\pi^*, \nu_0)$. Since $\overline{\mathcal{L}_{\text{BE}}}(Q)$ is of C^2, by [Rebjock and Boumal, 2023], showing PL is equivalent to showing to Quadratic Growth (QG), i.e., there exists $c' > 0$ such that

$$\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}[\mathcal{L}_{\text{BE}}(Q)(s,a)] - \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}[\mathcal{L}_{\text{BE}}(Q^*)(s,a)] \geq c' \|Q - Q^*\|^2_{L^2(\pi^*,\nu_0)}.$$

But note that

$$\begin{aligned}
&\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}[\mathcal{L}_{\text{BE}}(Q)(s,a)] - \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}[\mathcal{L}_{\text{BE}}(Q^*)(s,a)] \\
&= \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}[\mathcal{L}_{\text{BE}}(Q)(s,a) - \mathcal{L}_{\text{BE}}(Q^*)(s,a)] \\
&\geq \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[c(s,a)^2 (Q(s,a) - Q^*(s,a))^2\right] \qquad (18) \\
&= c^2 \|Q - Q^*\|^2_{L^2(\pi^*,\nu_0)}
\end{aligned}$$

where equation (18) is due to $\mathcal{L}_{BE}(Q)(s,a)$ being QG because it is smooth and therefore PL implies QG [Liao et al., 2024]. ($c(s,a) > 0$ is the QG constant for (s,a) and $c = \inf_{(s,a)\in\mathcal{S}\times\mathcal{A}} c(s,a)$.) This finishes the proof. □

LEMMA B.5. *For any given fixed $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $\mathcal{L}_{BE}(Q)(s,a)$ satisfies PL condition with respect to Q in terms of euclidean norm.*

PROOF OF LEMMA B.5. Throughout the proof, we extend [Ruszczyński and Yang, 2024] to deal with soft-max Bellman equation with infinite dimensional state space $\mathcal{S}$. Given that $|\mathcal{A}| < \infty$, for each $s \in \mathcal{S}$, $Q(s,\cdot)$ can be expressed as a finite-dimensional vector $[Q(s,a')]_{a'\in\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$; For convenience in notation, we define $q : \mathcal{S} \mapsto \mathbb{R}^{|A|}$ and

$$\mathcal{G}(s) : \{q(s) \in \mathbb{R}^{|\mathcal{A}|} \mid q(s) = [Q(s,a')]_{a'\in\mathcal{A}} \text{ for some } Q \in \mathcal{Q}\}$$

and use $q(s)$ instead of $Q(s,\cdot)$ and $q^*(s)$ instead of $Q^*(s,\cdot)$. We define

$$\Psi(s,a,q) \coloneqq r(s,a) + \beta \cdot \mathbb{E}_{s'\sim P(s,a)}\left[\log(\sum_{a'\in\mathcal{A}} \exp q(s')_{(a')}) \mid s,a\right] - q(s)_{(a)}$$

Now with $q^*(\cdot) \coloneqq [Q^*(\cdot,a)]_{a\in\mathcal{A}}$, let's define

$$f(s,a,q) \coloneqq \frac{1}{2}(\Psi(s,a,q^*) - \Psi(s,a,q))^2$$

Then, for $s \in \mathcal{S}$, with the choice of $q(\tau) := q^* + \tau(q - q^*)$,

$$\begin{aligned} f_q(s,a,q) := \partial_q f(s,a,q) &= -\Psi_q(s,a,q)(\Psi(s,a,q^*) - \Psi(s,a,q)) \\ &= -\Psi_q(s,a,q) \int_0^1 \Psi_q\left(s,a,q(\tau)\right)^\top (q^*(s) - q(s)) d\tau \qquad \text{(Theorem B.6)} \\ &= -\int_0^1 \Psi_q(s,a,q)\Psi_q\left(s,a,q(\tau)\right)^\top d\tau \cdot (q^*(s) - q(s)) \end{aligned}$$

By Lemma B.7,there exists $\tilde{\lambda}$ such that for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $\Psi_q(s,a,q')\Psi_q(s,a,q'')^\top \geq \tilde{\lambda} \cdot I$ for any choice of $q'(s), q''(s) \in \mathcal{G}(s)$. Therefore we have

$$\left\langle f_q(s,a,q), q(s) - q^*(s) \right\rangle \geq \tilde{\lambda} \|q(s) - q^*(s)\|_2^2.$$

This implies that

$$\begin{aligned} \|f_q(s,a,q)\|_2 = \max_{\|z\|=1} \left\langle f_q(s,a,q), z \right\rangle &\geq \left\langle f_q(s,a,q), \frac{q(s) - q^*(s)}{\|q(s) - q^*(s)\|_2} \right\rangle \\ \geq \tilde{\lambda} \|q(s) - q^*(s)\|_2 &\geq \tilde{\lambda} \|q(s) - q^*(s)\|_\infty \end{aligned} \tag{19}$$

Therefore,

$$\|f_q(s,a,q)\|_2 \geq \tilde{\lambda} \|q(s) - q^*(s)\|_\infty \tag{20}$$

(Note: Equation 20 is a regularity condition called sub-differential error bound.) Also, from Lemma B.8,

$$\begin{aligned} f(s,a,q) &= \frac{1}{2}(\Psi(s,a,q^*) - \Psi(s,a,q))^2 \\ &\leq \frac{1}{2}(1+\beta)^2 \|q(s) - q^*(s)\|_\infty^2 \end{aligned} \tag{21}$$

Combining equation 21 and 20, we get

$$f(s,a,q) \leq \frac{1}{2}\left(\frac{1+\beta}{\tilde{\lambda}}\right)^2 \|f_q(s,a,q)\|_2^2 \quad \text{for all } s \in \mathcal{S}, a \in \mathcal{A}$$

Since $\Psi\left(s,a,q^*\right) = 0$, $f(s,a,q) = \mathcal{L}_{BE}(Q)(s,a)$, where $q(s) = \left[Q\left(s,a'\right)\right]_{a' \in \mathcal{A}}$. This finishes the proof.

□

THEOREM B.6 ([BOLTE ET AL., 2023]). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. If a path $q : [0,\infty) \to \mathbb{R}^n$ is a absolutely continuous path in $\mathbb{R}^n$, f admits the chain rule on the path $q(t)$ as*

$$f(q(T)) - f(q(0)) = \int_0^T f_q(q(t))[\dot{q}(t)]dt$$

where $\dot{q}(t)$ is the derivative of the function path $q(t)$ with respect to t and $T > 0$.

LEMMA B.7 (POSITIVE SMALLEST EIGENVALUE). *Suppose that the discount factor $\beta < 1$. Then for there exists $\tilde{\lambda} > 0$ such that for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $\lambda_{\min}(\Psi_q\left(s,a,q'\right)\Psi_q\left(s,a,q''\right)^\top) > \tilde{\lambda}$ holds for any choice of $q', q'' \in \mathcal{G}(s)$.*

PROOF. First, note that we can define the policy $\pi_q(a|s) = \frac{\exp q_{(a)}}{\sum_{a'} \exp q_{(a')}}$ for $q \in \mathbb{R}^{|\mathcal{A}|}$, where $x_{(a)}$ implies the ath element of vector x.

$$\frac{\partial \Psi(s,a,q)}{\partial q_{(a')}} = \beta \mathbb{E}_{s' \sim P(s,a)}\left[\pi_q\left(a' \mid s'\right)\right] - \delta_{a,a'}$$

That is, $\Psi_q(s,a,q) = \beta\mu_q - e_q$, where $\mu_q = \mathbb{E}_{s'\sim P(s,a)}\left[\pi_q\left(a' \mid s'\right)\right]$ is a probability vector, as it's an expectation over probability distributions. Then for any choice of $q', q'' \in \mathcal{G}(s)$, denoting $\mu_{q'} = \mu'$ and $\mu_{q''} = \mu''$

$$\begin{aligned}
\lambda\left(\Psi_q\left(s,a,q'\right)\Psi_q\left(s,a,q''\right)^\top\right) &= \lambda\left(\left(\beta\mu' - e_a\right)\left(\beta\mu'' - e_a\right)^\top\right) \\
&= \left(\beta\mu' - e_a\right)^\top\left(\beta\mu'' - e_a\right) \\
&= \beta^2\left(\mu'\right)^\top\mu'' - \beta\mu'(a) - \beta\mu''(a) + 1 \\
&\geq \beta^2\mu'(a)\mu''(a) - \beta\mu'(a) - \beta\mu''(a) + 1 \\
&= (1-\beta\mu'(a))(1-\beta\mu''(a)) \qquad (22) \\
&\geq (1-\beta)^2
\end{aligned}$$

Since $\beta \in (0,1)$, $\tilde{\lambda} = (1-\beta)^2$ serves as the uniform lower bound of $\lambda_{\min}(\Psi_q\left(s,a,q'\right)\Psi_q\left(s,a,q''\right)^\top)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, for any choice of $q', q'' \in \mathcal{G}(s)$. □

LEMMA B.8. *$|(\mathcal{T}Q - Q)(s,a) - (\mathcal{T}Q^* - Q^*)(s,a)| \leq (1+\beta)\left\|Q\left(s',\cdot\right) - Q^*\left(s',\cdot\right)\right\|_\infty$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$.*

PROOF.

$$\begin{aligned}
&|(\mathcal{T}Q - Q)(s,a) - (\mathcal{T}Q^* - Q^*)(s,a)| \\
&= \left|\beta\cdot\mathbb{E}_{s'\sim P(s,a)}\left[\log\left(\sum_{a'\in\mathcal{A}}\exp Q\left(s',a'\right)\right) - \log\left(\sum_{a'\in\mathcal{A}}\exp Q^*\left(s',a'\right)\right) \mid s,a\right] + (Q^*(s,a) - Q(s,a))\right| \\
&\leq \left|\beta\cdot\mathbb{E}_{s'\sim P(s,a)}\left[\left\|Q\left(s',\cdot\right) - Q^*\left(s',\cdot\right)\right\|_\infty\right] + |Q^*(s,a) - Q(s,a)| \qquad \text{(logsumexp Liptshitz in 1)} \right. \\
&\leq (\beta+1)\left\|Q\left(s',\cdot\right) - Q^*\left(s',\cdot\right)\right\|_\infty
\end{aligned}$$

□

B.7 Proof of Theorem 6.4 (NLL loss satisfying the PL condition)

From Lemma B.2, we know

$$\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[-\log\left(\hat{p}_Q(a \mid s)\right)\right] = \mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_Q(\cdot \mid s)\right)\right] + \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\ln\pi^*(a \mid s)\right]$$

Note that the second term is not dependent on Q. Therefore, we will instead show that the PL condition holds for $\mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_Q(\cdot \mid s)\right)\right]$. Since $\mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_Q(\cdot \mid s)\right)\right]$ is convex, by [Liao et al., 2024], showing that $\mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_Q(\cdot \mid s)\right)\right]$ is PL is equivalent to showing that $\mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_Q(\cdot \mid s)\right)\right]$ satisfies Quadratic Growth (QG) condition, i.e., there exists $c' > 0$ such that

$$\mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_Q(\cdot \mid s)\right)\right] - \mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_{Q^*}(\cdot \mid s)\right)\right] \geq c'\left\|Q - Q^*\right\|^2_{L^2(\pi^*,\nu_0)}$$

But note that

$$\begin{aligned}
&\mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_Q(\cdot \mid s)\right)\right] - \mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_{Q^*}(\cdot \mid s)\right)\right] \\
&= \mathbb{E}_{s\sim\pi^*,\nu_0}\left[D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_Q(\cdot \mid s)\right) - D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_{Q^*}(\cdot \mid s)\right)\right] \\
&\geq \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[c(s,a)^2(Q(s,a) - Q^*(s,a))^2\right] \qquad \text{(Lemma B.9)} \\
&= c^2\|Q - Q^*\|^2_{L^2(\pi^*,\nu_0)}
\end{aligned}$$

where $c(s,a) > 0$ is the QG constant for (s,a) and $c = \inf_{(s,a)\in\mathcal{S}\times\mathcal{A}} c(s,a)$. Done. □

LEMMA B.9. *$D_{KL}\left(\pi^*(\cdot \mid s)\|\hat{p}_Q(\cdot \mid s)\right)$ satisfies the PL condition for each $s \in \mathcal{S}$.*

PROOF. Note that

$$\begin{aligned}\nabla_{Q(s,\cdot)} D_{KL}\left(\pi^*(\cdot \mid s) \| \hat{p}_Q(\cdot \mid s)\right) &= \nabla_{Q(s,\cdot)} \left(-\sum_a \pi^*(a \mid s) \log \hat{p}_Q(a \mid s)\right) \\ &= -\sum_a \pi^*(a \mid s) \left(\delta_{a,a'} - \hat{p}_Q\left(a' \mid s\right)\right) \\ &= -\left[\pi^*\left(a' \mid s\right) - \hat{p}_Q\left(a' \mid s\right) \sum_a \pi^*(a \mid s)\right]_{a' \in \mathcal{A}} \\ &= \left[\hat{p}_Q\left(a' \mid s\right) - \pi^*\left(a' \mid s\right)\right]_{a' \in \mathcal{A}}\end{aligned}$$

Then,

$$\begin{aligned}\|\nabla_{Q(s,)} D_{KL}\left(\pi^*(\cdot \mid s) \| \hat{p}_Q(\cdot \mid s)\right)\|^2 &= \| \left[\hat{p}_Q\left(a' \mid s\right) - \pi^*\left(a' \mid s\right)\right]_{a' \in \mathcal{A}} \|_2^2 \\ &\geq \frac{1}{|\mathcal{A}|} \| \left[\hat{p}_Q\left(a' \mid s\right) - \pi^*\left(a' \mid s\right)\right]_{a' \in \mathcal{A}} \|_1^2 \\ &= \frac{1}{|\mathcal{A}|} \mathrm{TV}\left(\hat{p}_Q\left(\cdot \mid s\right), \pi^*\left(\cdot \mid s\right)\right)^2 \\ &\geq \frac{\alpha_Q \ln 2}{|\mathcal{A}|} D_{KL}\left(\pi^*(\cdot \mid s) \| \hat{p}_Q(\cdot \mid s)\right)\end{aligned}$$

where,

- TV denotes the total variation distance.
- The last inequality is from Lemma B.10, where $\alpha_Q := \min_{a \in A_+} Q(s,a) > 0$ with $A_+ = \{a \in \mathcal{A} : Q(s,a) > 0\}$.

□

LEMMA B.10 (REVERSE PINSKER'S INEQUALITY).

$$\begin{aligned}D(P\|Q) = \sum_{a \in A_+} P(a) \log_2 \frac{P(a)}{Q(a)} &\leq \frac{1}{\ln 2} \sum_{a \in A_+} P(a) \left(\frac{P(a)}{Q(a)} - 1\right) \\ &= \frac{1}{\ln 2} \sum_{a \in A_+} \frac{(P(a) - Q(a))^2}{Q(a)} + \sum_{a \in A_+} (P(a) - Q(a)) \\ &= \frac{1}{\ln 2} \sum_{a \in A_+} \frac{(P(a) - Q(a))^2}{Q(a)} \\ &\leq \frac{d(P,Q)^2}{\alpha_Q \cdot \ln 2}\end{aligned}$$

LEMMA B.11. *Suppose that given fixed $z \in \mathcal{Z}$, a smooth function $f(x,z)$ 1) either satisfies convexity in x or of C^2 in x and 2) satisfies Polyak-Łojasiewicz condition in x with the coefficient $\mu_z > 0$, i.e.,*

$$\|\nabla_x f(x,z)\|_2^2 \geq 2\mu_z \left[f(x,z) - f_z^*\right]$$

where $f_z^* = \min_x f_z(x)$ and $\mu_z > 0$. In addition, suppose that $\arg\min_x f(x,z) = \arg\min_x f(x,z')$ for all $z, z' \in \mathcal{Z}$, where we define the common minimizer as x^*. Then $F(x) := \mathbb{E}_{z \sim \nu}[f(x,z)]$ satisfies Polyak-Łojasiewicz condition with respect to x, given that ν is a measure defined on $\mathcal{Z}$. That is,

$$\|\nabla_x F(x)\|_2^2 \geq 2\mu \left[F(x) - F^*\right],$$

where $F^* := \min_x F(x) = \mathbb{E}_{z \sim \nu}\left[f_z^*\right]$, and $\mu = \inf_{z \in \mathcal{Z}} \mu_z > 0$.

PROOF. Since f is smooth and satisfies PL condition with respect to x for given $z \in \mathcal{Z}$, it satisfies the Quadratic Growth (QG) condition [Liao et al., 2024], i.e., for fixed $z \in \mathcal{Z}$, there exists $\alpha_z > 0$ such that:

$$f(x,z) - f_z^* \geq \alpha_z \|x - x^*\|^2 \quad \forall x \in \mathcal{X}$$

Therefore,

$$\begin{aligned} F(x) - F^* &= \mathbb{E}_z \left[f(x,z) - f_z^* \right] \\ &\geq \mathbb{E}_z \left[\alpha_z \|x - x^*\|^2 \right] \\ &\geq \alpha \|x - x^*\|^2 \quad (\alpha := \inf_z \alpha_z > 0) \end{aligned}$$

This implies that $F(x)$ satisfies QG condition in x. If f satisfies convexity, then by Liao et al. [2024], Quadratic growth and PL are equivalent; if f is of C^2, then by Rebjock and Boumal [2023], Quadratic Growth and PL are equivalent. Therefore, $F(x)$ satisfies PL. □

B.8 Proof of Lemma 6.5

LEMMA B.12 (LEMMA 6.5).
Define $f_1(Q) = \mathbb{E}_{(s,a)\sim\pi^*,\nu_0} \left[-\log \left(\hat{p}_Q(\ a \mid s) \right) \right]$, $f_2(Q) = \mathbb{E}_{(s,a)\sim\pi^*,\nu_0} \left[\mathbb{1}_{a=a_s} \mathcal{L}_{BE}(Q)\ (s,a) \right]$. *Then* $f_1 + \lambda f_2$ *satisfies PL condition for any* $\lambda > 0$.

PROOF OF LEMMA 6.5. Without loss of generality, we prove that $f := f_1 + f_2$ satisfies PL condition. Recall that we say f satisfies μ-*PL* condition if $2\mu(f(Q) - f\ (Q^*)) \leq \|\nabla f(Q)\|^2$.

$$\begin{aligned} \|\nabla f(Q)\|^2 &= \|\nabla f_1(Q) + \nabla f_2(Q)\|^2 \\ &= \|\nabla f_1(Q)\|^2 + \|\nabla f_2(Q)\|^2 + 2\nabla f_1(Q)^\top \nabla f_2(Q) \\ &\geq 2\mu_1(f_1(Q) - f_1(Q^*)) + 2\mu_2(f_2(Q) - f_2(Q^*)) + 2\nabla f_1(Q)^\top \nabla f_2(Q) \\ &\geq 2\mu(f_1(Q) + f_2(Q) - f_1(Q^*) - f_2(Q^*)) + 2\nabla f_1(Q)^\top \nabla f_2(Q) \\ &= 2\mu(f(Q) - f(Q^*) + 2\nabla f_1(Q)^\top \nabla f_2(Q) \\ &\geq 2\mu(f(Q) - f(Q^*) \qquad \text{(Lemma B.13)} \end{aligned}$$

The last inequality is not trivial, and therefore requires Lemma B.13; it uses the fact that 1) f_1 and f_2 satisfies quadratic growth 2) Minimizer of $f_1 + f_2$ is unique.

□

LEMMA B.13. *For all* $Q \in \mathcal{Q}$, $\langle \nabla f_1(Q), \nabla f_2(Q) \rangle \geq 0$.

PROOF. Let $M_1 := \left\{ Q \in \mathcal{Q} : f_1(Q) = f_1^* \right\}$, and $M_2 := \left\{ Q \in \mathcal{Q} : f_2(Q) = f_2^* \right\}$. By Theorem 4.1, $f_1 + f_2$ has a unique minimizer Q^* that belongs to M_1 and therefore belongs to $M_1 \cap M_2$; otherwise, there would be a point in $M_1 \cap M_2$ (if non-empty) that attains the same or lower value for the sum, contradicting uniqueness.

Since f_1 and f_2 are both Lipschitz smooth and satisfies PL condition, they both satisfy Quadratic Growth (QG) condition [Liao et al., 2024], i.e., there exists $\alpha_1, \alpha_2 > 0$ such that:

$$f_1(Q) - f_1\ (Q^*) \geq \alpha_1 \|Q - Q^*\|^2 \quad \forall Q \in \mathcal{Q}$$

$$f_2(Q) - f_2\ (Q^*) \geq \alpha_2 \|Q - Q^*\|^2 \quad \forall Q \in \mathcal{Q}$$

Now suppose, for the purpose of contradiction, that there exists $\hat{Q} \in \mathcal{Q}$ such that $\left\langle \nabla f_1(\hat{Q}), \nabla f_2(\hat{Q}) \right\rangle < 0$. Consider the direction $d := -g_1 = -\nabla f_1(\hat{Q})$. Then $\nabla f_1(\hat{Q})^\top d = g_1^\top\ (-g_1) = -\|g_1\|^2 < 0$ holds. This implies that $f_1(\hat{Q} + \eta d) < f_1(\hat{Q})$. Then QG condition for f_1 implies that

$$\left\| \hat{Q} + \eta d - Q^* \right\| < \left\| \hat{Q} - Q^* \right\|$$

Now, note that $\nabla f_2(\hat{Q})^\top d = g_2^\top(-g_1) = -g_1^\top g_2$. Since $g_1^\top g_2 < 0$, $\nabla f_2(\hat{Q})^\top d > 0$. Therefore, $f_2(\hat{Q} + \eta d) > f_2(\hat{Q})$ for sufficiently small $\eta > 0$. That is, $f_2(\hat{Q} + \eta d) - f_2(Q^*) > f_2(\hat{Q}) - f_2(Q^*)$. By QG condition, this implies that $\left\|\hat{Q} + \eta d - Q^*\right\| > \left\|\hat{Q} - Q^*\right\|$. Contradiction.

□

B.9 Proof of Lemma 6.6

We consider the function class

$$Q_\theta(s,a) = \theta^\top \phi(s,a)$$

where $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is a known feature map with $\|\phi(s,a)\| \le B$ almost surely and $\theta \in \mathbb{R}^d$ is the parameter vector. Then for any unit vector $u \in \mathbb{R}^d$,

$$\left|u^\top \phi(s,a)\right| \le \|u\|\|\phi(s,a)\| = B$$

Then by using Hoeffding's Lemma, we have

$$\mathbb{E}\left[e^{\lambda u^\top \phi(s,a)}\right] \le \exp\left(\frac{\lambda^2 B^2}{2}\right)$$

Therefore we have

$$\mathbb{P}\left(\left|u^\top \phi(s,a)\right| \ge t\right) \le 2e^{-t^2/(2B^2)} \quad \forall t > 0$$

Now for the given dataset $\mathcal{D}$, define

$$M = \begin{bmatrix} \phi(s_1,a_1)^\top \\ \phi(s_2,a_2)^\top \\ \vdots \\ \phi\left(s_{|\mathcal{D}|}, a_{|\mathcal{D}|}\right)^\top \end{bmatrix} \in \mathbb{R}^{|\mathcal{D}| \times d}$$

Note that $DQ_\theta = M$. Then by Rudelson and Vershynin [2009], we have

$$\mathbb{P}\left(\sigma_{\min}(DQ_\theta) \ge \sqrt{|\mathcal{D}|} - \sqrt{d}\right) \ge 1 - e^{-C|\mathcal{D}|}$$

provided that the dataset size satisfies $|\mathcal{D}| \ge Cd$ with $C > 1$.

B.10 Proof of Theorem 6.8

Proof of Theorem 6.8. For convenience in notation,

$$f(Q_\theta) := \overline{\mathcal{L}_{\mathrm{NLL}}}(Q_\theta) + \lambda \mathbb{1}_{a=a_s} \overline{\mathcal{L}_{\mathrm{BE}}}(Q_\theta)$$

and denote $Q_\theta = Q(\theta)$. Set $h(\theta) := f(Q(\theta))$, where f is the loss in terms of the function Q. Then $h(\theta^*) = f(Q(\theta^*)) = f(Q^*)$ by realizability (Assumption 6.1). Then

$$\begin{aligned}
\|\nabla_\theta h(\theta)\|_2^2 &= \left\|DQ(\theta)^\top \nabla_Q f(Q(\theta))\right\|_2^2 \\
&\ge \sigma_{\min}^2\left(DQ(\theta)\right)\left\|\nabla_Q f\left(Q(\theta)\right)\right\|_2^2 && (\dim(\mathcal{S}), \dim(\mathcal{A}) < \infty) \\
&\ge m^2\left\|\nabla_Q f\left(Q(\theta)\right)\right\|_2^2 && \text{(Assumption 6.2)} \\
&\ge 2(m^2 c)(f(Q(\theta)) - f(Q^*)) && \text{(PL in terms of } Q) \\
&= 2(m^2 c)(h(\theta) - h(\theta^*))
\end{aligned}$$

□

B.11 Proof of Proposition 6.1 (Global optima convergence under ERM-IRL)

1. Optimization error analysis.

Define $f_1(Q) = \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[-\log\left(\hat{p}_Q(\ a \mid s)\right)\right]$ and $f_2(Q) = \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\mathbb{1}_{a=a_s}\mathcal{L}_{BE}(Q)\,(s,a)\right]$. By Theorem 4.1, there is a unique minimizer Q^* for $f_1 + \lambda f_2$, which is the same for all $\lambda > 0$. Also, $f_1 + \lambda f_2$ satisfies PL condition by Lemma 6.5.

In equation 5 of Theorem 4.1, we saw that $f_2(Q)$ is actually of form $\max_\zeta f_2(Q,\zeta)$. This implies that minimization of $f_1 + \lambda f_2$, a mini-max optimization problem that satisfies two-sided PL. (The inner maximization problem is trivially strongly convex, which implies PL).

Now denote

$$f_\lambda(Q,\zeta) := f_1(Q) + \lambda f_2(Q,\zeta)$$

$$g_\lambda(Q) := \max_\zeta f_\lambda(Q,\zeta)$$

$$g_\lambda^* = \min_Q g_\lambda(Q) = \min_Q \max_\zeta f_\lambda(Q,\zeta)$$

Note that

$$g_\lambda(Q) - g_\lambda^* \geq 0$$

$$g_\lambda(Q) - f_\lambda(Q,\zeta) \geq 0$$

for any (Q,ζ). Furthermore, they are both equal to 0 if and only if (Q,ζ) is a minimax point, which is Q^* and ζ^*. More precisely, we have

$$|f_\lambda(Q,\zeta) - g_\lambda^*| \leq (g_\lambda(Q) - g_\lambda^*) + (g_\lambda(Q) - f_\lambda(Q,\zeta))$$

Therefore, we would like to find Q,ζ that for $\alpha > 0$ $a_\lambda(Q) + \alpha b(Q,\zeta) = 0$, where

$$a_\lambda(Q) := g_\lambda(Q) - g_\lambda^*$$

$$b_\lambda(Q,\zeta) := g_\lambda(Q) - f_\lambda(Q,\zeta)$$

At iteration 0, algorithm starts from $\hat{Q}_0$ and $\zeta = \zeta_0$. We denote the Q,ζ value at iteration T as $\hat{Q}_T$ and ζ_T. Also, we define P_T as

$$P_T := a_\lambda(\hat{Q}_T) + \alpha b_\lambda(\hat{Q}_T,\zeta_T)$$

Set that $f_\lambda(Q,\zeta)$ satisfies μ_1-PL for Q and μ_2-PL for ζ. Let $\alpha = 1/10$, $\tau_1^T = \frac{\beta}{\gamma+T}, \tau_2^T = \frac{18l^2\beta}{\mu_2^2(\gamma+T)}$ for some $\beta > 2/\mu_1$, $L = l + l^2/\mu_2$, and $\gamma > 0$ such that $\tau_1^1 \leq \min\left\{1/L, \mu_2^2/18l^2\right\}$. Then the following Theorem holds.

THEOREM B.14 (THEOREM 3.3, YANG ET AL [YANG ET AL., 2020]). *Consider the setup where $\lambda > 0$ is fixed. Then applying Algorithm 1 using stochastic gradient descent (SGD), P_T satisfies*

$$P_T \leq \frac{\nu}{\gamma + T}$$

Note that a_λ satisfies PL condition with respect to Q and smoothness, since subtracting constant from PL is PL. Therefore, a_λ satisfies Quadratic Growth (QG) condition by [Liao et al., 2024], i.e.,

$$C \cdot \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\left(\hat{Q}_T(s,a) - \hat{Q}_N(s,a)\right)^2\right] \leq a_\lambda(Q) - 0 \leq O(1/T).$$

Since $a_\lambda \leq P_T$, we can conclude that $\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\left(\hat{Q}_T(s,a) - \hat{Q}_N(s,a)\right)^2\right]$ is $O(1/T)$.

2. Statistical error analysis.

LEMMA B.15. *When minimizing the mean squared Bellman error of form*

$$\overline{\mathcal{L}_{\text{BE}}}(Q) \coloneqq \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\mathcal{L}_{\text{BE}}(Q)(s,a)\right]$$

with

$$\begin{aligned}\mathcal{L}_{BE}(Q)(s,a) &= \left(\mathcal{T}Q(s,a) - Q(s,a)\right)^2 \\ &= \max_{\rho\in\mathbb{R}} \mathbb{E}_{s'\sim P(s,a)}\left[\mathcal{L}_{TD}(Q)\left(s,a,s'\right)\right] - \left(\hat{\mathcal{T}}Q\left(s,a,s'\right) - \rho\right)^2\Bigg]\end{aligned}$$

where

$$\hat{\mathcal{T}}Q\left(s,a,s'\right) = r(s,a) + \beta\cdot\log\sum_{a'\in\mathcal{A}}\exp\left(Q\left(s',a'\right)\right)$$

PROOF. By adapting Lemma E.5 of Touati and Vincent [2020] for expert training data and softmax policy, we have

$$\sqrt{\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\left(Q^*(s,a) - \hat{Q}_N(s,a)\right)^2\right]} = O(1/\sqrt{N})$$

This immediately implies that

$$\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\left(Q^*(s,a) - \hat{Q}_N(s,a)\right)^2\right] = O(1/N)$$

□

LEMMA B.16 ([COMBES, 2024]).
With $|\mathcal{D}| = N$, $\left|\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[-\log\left(\hat{p}_Q(a\mid s)\right)\right] - \frac{1}{|\mathcal{D}|}\sum_{(s,a)\in\mathcal{D}}(-\log(\hat{p}_Q(a\mid s)))\right| = O(1/\sqrt{N})$.

Since $\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[-\log\left(\hat{p}_Q(a\mid s)\right)\right]$ satisfies PL and smoothness, it satisfies Quadratic Growth condition ([Liao et al., 2024]). Therefore

$$\left|\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[-\log\left(\hat{p}_Q(a\mid s)\right)\right] - \frac{1}{|\mathcal{D}|}\sum_{(s,a)\in\mathcal{D}}(-\log(\hat{p}_Q(a\mid s)))\right| \geq C\cdot\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\left(Q(s,a) - \hat{Q}_N(s,a)\right)^2\right]$$

for some C. This implies that the sample complexity is $O(1/\sqrt{N})$.

Combining the above two Lemma, we conclude that

$$\mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\left(Q^*(s,a) - \hat{Q}_N(s,a)\right)^2\right] = O(1/\sqrt{N})$$

.

Below are the Lemmas used above. Throughout those Lemmas, we use $f_1(Q) = \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[-\log\left(\hat{p}_Q(a\mid s)\right)\right]$ and $f_2(Q) = \mathbb{E}_{(s,a)\sim\pi^*,\nu_0}\left[\mathbf{1}_{a=a_s}\mathcal{L}_{BE}(Q)(s,a)\right]$.

C Equivalence between Dynamic Discrete choice and Entropy regularized Inverse Reinforcement learning

C.1 Properties of Type 1 Extreme Value (T1EV) distribution

Type 1 Extreme Value (T1EV), or Gumbel distribution, has a location parameter and a scale parameter. The T1EV distribution with location parameter ν and scale parameter 1 is denoted as Gumbel $(\nu, 1)$ and has its CDF, PDF, and mean as follows:

$$\text{CDF: } F(x;\nu) = e^{-e^{-(x-\nu)}}$$
$$\text{PDF: } f(x;\nu) = e^{-\left((x-\nu)+e^{-(x-\nu)}\right)}$$
$$\text{Mean } = \nu + \gamma$$

Suppose that we are given a set of N independent Gumbel random variables G_i, each with their own parameter ν_i, i.e. $G_i \sim \text{Gumbel}\,(\nu_i, 1)$.

LEMMA C.1. *Let $Z = \max G_i$. Then $Z \sim \text{Gumbel}\,(\nu_Z = \log \sum_i e^{\nu_i}, 1)$.*

PROOF. $F_Z(x) = \prod_i F_{G_i}(x) = \prod_i e^{-e^{-(x-\nu_i)}} = e^{-\sum_i e^{-(x-\nu_i)}} = e^{-e^{-x}\sum_i e^{\nu_i}} = e^{-e^{-(x-\nu_Z)}}$ □

COROLLARY C.2. $P\left(G_k > \max_{i\neq k} G_i\right) = \frac{e^{\nu_k}}{\sum_i e^{\nu_i}}$.

PROOF.

$$\begin{aligned}
P\left(G_k > \max_{i\neq k} G_i\right) &= \int_{-\infty}^{\infty} P\left(\max_{i\neq k} G_i < x\right) f_{G_k}(x)dx \\
&= \int_{-\infty}^{\infty} e^{-\sum_{i\neq k} e^{-(x-\nu_i)}} e^{-(x-\nu_k)} e^{-e^{-(x-\nu_k)}} dx \\
&= e^{\nu_k} \int_{-\infty}^{\infty} e^{-e^{-x}\sum_i e^{\nu_i}} e^{-x} dx \\
&= e^{\nu_k} \int_{\infty}^{0} e^{-uS} u \left(-\frac{du}{u}\right) \quad \left(\text{Let } \sum_i e^{\nu_i} = S, u = e^{-x}\right) \\
&= e^{\nu_k} \int_{0}^{\infty} e^{-uS} du = e^{\nu_k} \left[-\frac{1}{S} e^{-uS}\right]_0^{\infty} = \frac{e^{\nu_k}}{S} \\
&= \frac{e^{\nu_k}}{\sum_i e^{\nu_i}}
\end{aligned}$$

□

LEMMA C.3. *Let $G_1 \sim$ Gumbel $(\nu_1, 1)$ and $G_2 \sim$ Gumbel $(\nu_2, 1)$. Then $\mathbb{E}\left[G_1 \mid G_1 \geq G_2\right] = \gamma + \log\left(1 + e^{(-(\nu_1-\nu_2))}\right)$ holds.*

PROOF. Let $\nu_1 - \nu_2 = c$. Then $\mathbb{E}\left[G_1 \mid G_1 \geq G_2\right]$ is equivalent to $\nu_1 + \frac{\int_{-\infty}^{+\infty} xF(x+c)f(x)\mathrm{d}x}{\int_{-\infty}^{+\infty} F(x+c)f(x)\mathrm{d}x}$, where the pdf f and cdf F are associated with Gumbel $(0, 1)$, because

$$P\left(G_1 \leq x, G_1 \geq G_2\right) = \int_{-\infty}^{x} F_{G_2}(t) f_{G_1}(t) dt = \int_{-\infty}^{x} F\left(t - v_2\right) f\left(t - v_1\right) dt$$

$$\begin{aligned}
\mathbb{E}\left[G_1 \mid G_1 \geq G_2\right] &= \frac{\int_{-\infty}^{\infty} xF\left(x + c - v_1\right) f\left(x - v_1\right) dx}{\int_{-\infty}^{\infty} F\left(x + c - v_1\right) f\left(x - v_1\right) dx} \\
&= \frac{\int_{-\infty}^{\infty} \left(y + v_1\right) F(y + c) f(y) dy}{\int_{-\infty}^{\infty} F(y + c) f(y) dy} \\
&= v_1 + \frac{\int_{-\infty}^{\infty} yF(y + c) f(y) dy}{\int_{-\infty}^{\infty} F(y + c) f(y) dy}
\end{aligned}$$

Now note that

$$\begin{aligned}
\int_{-\infty}^{+\infty} F(x + c) f(x) \mathrm{d}x &= \int_{-\infty}^{+\infty} \exp\{-\exp[-x - c]\} \exp\{-x\} \exp\{-\exp[-x]\} \mathrm{d}x \\
&\overset{a=e^{-c}}{=} \int_{-\infty}^{+\infty} \exp\{-(1 + a) \exp[-x]\} \exp\{-x\} \mathrm{d}x \\
&= \frac{1}{1 + a} \left[\exp\left\{-(1 + a) e^{-x}\right\}\right]_{-\infty}^{+\infty} \\
&= \frac{1}{1 + a}
\end{aligned}$$

and

$$\begin{aligned}
\int_{-\infty}^{+\infty} xF(x + c) f(x) \mathrm{d}x &= \int_{-\infty}^{+\infty} x \exp\{-(1 + a) \exp[-x]\} \exp\{-x\} \mathrm{d}x \\
&\overset{z=e^{-x}}{=} \int_{0}^{+\infty} \log(z) \exp\{-(1 + a) z\} \mathrm{d}z \\
&= \frac{-1}{1 + a} \left[\mathrm{Ei}(-(1 + a) z) - \log(z) e^{-(1+a)z}\right]_{0}^{\infty} \\
&= \frac{\gamma + \log(1 + a)}{1 + a}
\end{aligned}$$

Therefore, $\mathbb{E}\left[G_1 \mid G_1 \geq G_2\right] = \gamma + v_k + \log\left(1 + e^{(-(v_1 - v_2))}\right)$ holds. □

COROLLARY C.4. $\mathbb{E}\left[G_k \mid G_k = \max G_i\right] = \gamma + v_k - \log\left(\frac{e^{v_k}}{\sum_i e^{v_i}}\right)$.

PROOF.

$$\begin{aligned}
\mathbb{E}\left[G_k \mid G_k = \max G_i\right] &= \mathbb{E}\left[G_k \mid G_k \geq \max_{i \neq k} G_i\right] \\
&= \gamma + \nu_k + \log\left(1 + e^{\left(-\left(\nu_k - \log \sum_{i \neq k} e^{\nu_i}\right)\right)}\right) && \text{(Lemma C.3)} \\
&= \gamma + \nu_k + \log\left(1 + \frac{\sum_{i \neq k} e^{\nu_i}}{e^{-\nu_k}}\right) \\
&= \gamma + \nu_k + \log\left(\sum_i e^{\nu_i} / e^{\nu_k}\right) \\
&= \gamma + \nu_k - \log\left(e^{\nu_k} / \sum_i e^{\nu_i}\right)
\end{aligned}$$

□

C.2 Properties of entropy regularization

Suppose we have a choice out of discrete choice set $\mathcal{A} = \{x_i\}_{i=1}^{|\mathcal{A}|}$. A choice policy can be a deterministic policy such as $\text{argmax}_{i \in 1,\ldots,|\mathcal{A}|} x_i$, or stochastic policy that is characterized by $\mathbf{q} \in \triangle_{\mathcal{A}}$. When we want to enforce smoothness in choice, we can regularize choice by newly defining the choice rule

$$\arg\max_{\mathbf{q} \in \triangle_{\mathcal{A}}} \left(\langle \mathbf{q}, \mathbf{x} \rangle - \Omega(\mathbf{q})\right)$$

where Ω is a regularizing function.

LEMMA C.5. *When the regularizing function is constant $-\tau$ multiple of Shannon entropy $H(\mathbf{q}) = -\sum_{i=1}^{|\mathcal{A}|} q_i \log(q_i)$,*

$$\max_{\mathbf{q} \in \triangle_{\mathcal{A}}} \left(\langle \mathbf{q}, \mathbf{x} \rangle - \Omega(\mathbf{q})\right) = \tau \log\left(\sum_i \exp\left(x_i / \tau\right)\right)$$

and

$$\arg\max_{\mathbf{q} \in \triangle_{\mathcal{A}}} \left(\langle \mathbf{q}, \mathbf{x} \rangle - \Omega(\mathbf{q})\right) = \frac{\exp\left(\frac{x_i}{\tau}\right)}{\sum_{j=1}^{n} \exp\left(\frac{x_j}{\tau}\right)}$$

PROOF. In the following, I will assume $\tau > 0$. Let

$$\begin{aligned}
G(\mathbf{q}) &= \langle \mathbf{q}, \mathbf{x} \rangle - \Omega(\mathbf{q}) \\
&= \sum_{i=1}^{n} q_i x_i - \tau \sum_{i=1}^{n} q_i \log(q_i) \\
&= \sum_{i=1}^{n} q_i \left(x_i - \tau \log(q_i)\right)
\end{aligned}$$

We are going to find the max by computing the gradient and setting it to 0 . We have

$$\frac{\partial G}{\partial q_i} = x_i - \tau\left(\log(q_i) + 1\right)$$

and

$$\frac{\partial G}{\partial q_i \partial q_j} = \begin{cases} -\frac{\tau}{q_i}, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases}$$

This last equation states that the Hessian matrix is negative definite (since it is diagonal and $-\frac{\tau}{q_1} < 0$), and thus ensures that the stationary point we compute is actually the maximum. Setting the gradient to $\mathbf{0}$ yields $q_i^* = \exp\left(\frac{x_i}{\tau} - 1\right)$, however the resulting $\mathbf{q}^*$ might not be a probability distribution. To ensure $\sum_{i=1}^n q_i^* = 1$, we add a normalization:

$$q_i^* = \frac{\exp\left(\frac{x_i}{\tau} - 1\right)}{\sum_{j=1}^n \exp\left(\frac{x_j}{\tau} - 1\right)} = \frac{\exp\left(\frac{x_i}{\tau}\right)}{\sum_{j=1}^n \exp\left(\frac{x_j}{\tau}\right)}.$$

This new $\mathbf{q}^*$ is still a stationary point and belongs to the probability simplex, so it must be the maximum. Hence, you get

$$\begin{aligned}\max_{\tau H}(\mathbf{x}) = G(\mathbf{q}^*) &= \sum_{i=1}^n \frac{\exp\left(\frac{x_i}{\tau}\right)}{\sum_{j=1}^n \exp\left(\frac{x_j}{\tau}\right)} x_i - \tau \sum_{i=1}^n \frac{\exp\left(\frac{x_1}{\tau}\right)}{\sum_{j=1}^n \exp\left(\frac{x_j}{\tau}\right)} \left(\frac{x_i}{\tau} - \log \sum_{i=1}^n \exp\left(\frac{x_j}{\tau}\right)\right) \\ &= \tau \log \sum_{i=1}^n \exp\left(\frac{x_j}{\tau}\right)\end{aligned}$$

as desired. □

C.3 IRL with entropy regularization

Markov decision processes. Consider an MDP defined by the tuple $(\mathcal{S}, \mathcal{A}, P, \nu_0, r, \beta)$:

- $\mathcal{S}$ and $\mathcal{A}$ denote finite state and action spaces
- $P \in \Delta_{\mathcal{S}}^{\mathcal{S}\times\mathcal{A}}$ is a Markovian transition kernel, and $\nu_0 \in \Delta_{\mathcal{S}}$ is the initial state distribution.
- $r \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is a reward function.
- $\beta \in (0, 1)$ a discount factor

C.3.1 Agent behaviors. Denote the distribution of agent's initial state $s_0 \in \mathcal{S}$ as ν_0. Given a stationary Markov policy $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$, an agent starts from initial state s_0 and make an action $a_h \in \mathcal{A}$ at state $s_h \in \mathcal{S}$ according to $a_h \sim \pi(\cdot \mid s_h)$ at each period h. We use $\mathbb{P}_{\nu_0}^{\pi}$ to denote the distribution over the sample space $(\mathcal{S} \times \mathcal{A})^\infty = \{(s_0, a_0, s_1, a_1, \ldots) : s_h \in \mathcal{S}, a_h \in \mathcal{A}, h \in \mathbb{N}\}$ induced by the policy π and the initial distribution ν_0. We also use $\mathbb{E}_\pi$ to denote the expectation with respect to $\mathbb{P}_{\nu_0}^{\pi}$. Maximum entropy inverse reinforcement learning (MaxEnt-IRL) makes the following assumption:

ASSUMPTION C.1 (ASSUMPTION 3.1). *Agent follows the policy* $\pi^* = \operatorname{argmax}_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}} \mathbb{E}_\pi \left[\sum_{h=0}^\infty \beta^h (r(s_h, a_h) +$ *where* $\mathcal{H}$ *denotes the Shannon entropy and* λ *is the regularization parameter.*

For the rest of the section, we use $\lambda = 1$.

We define the function V as $V(s_{h'}) = \mathbb{E}_{\pi^*}\left[\sum_{h=h'}^\infty \beta^h (r(s_h, a_h) + \mathcal{H}(\pi^*(\cdot \mid s_h)))\right]$ and call it the *value function*. According to Assumption 3.1, the value function V must satisfy the Bellman equation,

i.e.,

$$
\begin{aligned}
V(s) &= \max_{\mathbf{q}\in\triangle_{\mathcal{A}}} \left\{\mathbb{E}_{a\sim\mathbf{q}}\left[r(s,a)+\beta\cdot\mathbb{E}\left[V(s')\mid s,a\right]\right]+\mathcal{H}(\mathbf{q})\right\} \\
&= \max_{\mathbf{q}\in\triangle_{\mathcal{A}}} \left\{\sum_{a\in\mathcal{A}} q_a\left(r(s,a)+\beta\cdot\mathbb{E}\left[V(s')\mid s,a\right]\right)+\mathcal{H}(\mathbf{q})\right\} = \max_{\mathbf{q}\in\triangle_{\mathcal{A}}} \left\{\sum_{a\in\mathcal{A}} q_a Q(s,a)+\mathcal{H}(\mathbf{q})\right\} \qquad (23)\\
&= \ln\left[\sum_{a\in\mathcal{A}}\exp\left(r(s,a)+\beta\cdot\mathbb{E}\left[V(s')\mid s,a\right]\right)\right] \qquad (24)\\
&= \ln\left[\sum_{a\in\mathcal{A}}\exp\left(Q(s,a)\right)\right] \qquad (25)
\end{aligned}
$$

and $\mathbf{q}^* := \arg\max_{\mathbf{q}\in\triangle_{\mathcal{A}}}\left\{\mathbb{E}_{a\sim\mathbf{q}}\left[r(s,a)+\beta\cdot\mathbb{E}\left[V(s')\mid s,a\right]\right]+\mathcal{H}(\mathbf{q})\right\}$ is characterized by

$$
\mathbf{q}^* = [q_1^*\ldots q_{|\mathcal{A}|}^*],\ \text{where } q_a^* = \frac{\exp\left(Q(s,a)\right)}{\sum_{a'\in\mathcal{A}}\exp\left(Q(s,a')\right)} \text{ for } a\in\mathcal{A} \qquad (26)
$$

where:

- $Q(s,a) := r(s,a)+\beta\cdot\mathbb{E}\left[V(s')\mid s,a\right]$
- Equality in equation 24 and equality in equation 26 is from Lemma C.5

This implies that

$$
\pi^*(a\mid s) = q_a^* = \frac{\exp\left(Q(s,a)\right)}{\sum_{a'\in\mathcal{A}}\exp\left(Q(s,a')\right)} \text{ for } a\in\mathcal{A}.
$$

In addition to Bellman equation in terms of value function V, Bellman equation in terms of choice-specific value function $Q(s,a)$ can be derived by combining $Q(s,a) := r(s,a)+\beta\cdot\mathbb{E}\left[V(s')\mid s,a\right]$ and equation 25:

$$
Q(s,a) = r(s,a)+\beta\mathbb{E}_{s'\sim P(s,a)}\left[\ln\left(\sum_{a'\in\mathcal{A}}\exp\left(Q(s',a')\right)\right)\mid s,a\right]
$$

We can also derive an alternative form of choice-specific value function $Q(s,a)$ by combining $Q(s,a) := r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a)}\left[V(s')\mid s,a\right]$ and equation 23:

$$
\begin{aligned}
Q(s,a) &= r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a)}\left[\max_{\mathbf{q}\in\triangle_{\mathcal{A}}}\left\{\sum_{a\in\mathcal{A}} q_a Q(s',a)+\mathcal{H}(\mathbf{q})\right\}\mid s,a\right] \\
&= r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a)}\left[\max_{\mathbf{q}\in\triangle_{\mathcal{A}}}\left\{\sum_{a\in\mathcal{A}} q_a\left(Q(s',a)-\log q_a\right)\right\}\mid s,a\right] \\
&= r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a),a'\sim\pi^*(a|\cdot)}\left[\left(Q(s',a')-\log\pi^*(a'\mid s')\right)\mid s,a\right] \qquad (27)\\
&= r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a)}\left[\left(Q(s',a')-\log\pi^*(a'\mid s')\right)\mid s,a\right]\ \text{for all } a'\in\mathcal{A}
\end{aligned}
$$

The last line comes from the fact that $Q(s',a')-\log\pi^*(a'\mid s')$ is equivalent to $\log\left(\sum_{a'\in\mathcal{A}}\exp\left(Q(s',a')\right)\right)$, which is a quantity that does not depend on the realization of specific action a'.

C.4 Single agent Dynamic Discrete Choice (DDC) model

Markov decision processes. Consider an MDP $\tau := (\mathcal{S}, \mathcal{A}, P, v_0, r, G(\delta, 1), \beta)$:

- $\mathcal{S}$ and $\mathcal{A}$ denote finite state and action spaces
- $P \in \Delta_{\mathcal{S}}^{\mathcal{S}\times\mathcal{A}}$ is a Markovian transition kernel, and $v_0 \in \Delta_{\mathcal{S}}$ is the initial state distribution.
- $r(s_h, a_h) + \epsilon_{ah}$ is the immediate reward (called the flow utility in the Discrete Choice Model literature) from taking action a_h at state s_h at time-step h, where:
  - $r \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is a deterministic reward function
  - $\epsilon_{ah} \overset{i.i.d.}{\sim} G(\delta, 1)$ is the random part of the reward, where G is Type 1 Extreme Value (T1EV) distribution (a.k.a. Gumbel distribution). The mean of $G(\delta, 1)$ is $\delta + \gamma$, where γ is the Euler constant.
  - In the econometrics literature, this reward setting is considered as a result of combination of two assumptions: conditional independence (CI) and additive separability (AS) [Magnac and Thesmar, 2002].

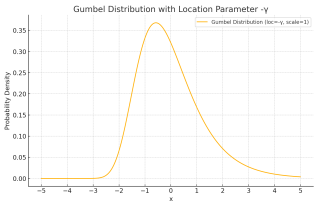

Fig. 4. Gumbel distribution $G(-\gamma, 1)$

- $\beta \in (0, 1)$ a discount factor

C.4.1 Agent behaviors. Denote the distribution of agent's initial state $s_0 \in \mathcal{S}$ as v_0. Given a stationary Markov policy $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$, an agent starts from initial state s_0 and make an action $a_h \in \mathcal{A}$ at state $s_h \in \mathcal{S}$ according to $a_h \sim \pi(\cdot \mid s_h)$ at each period h. We use $\mathbb{P}_{v_0}^{\pi}$ to denote the distribution over the sample space $(\mathcal{S} \times \mathcal{A})^{\infty} = \{(s_0, a_0, s_1, a_1, \ldots) : s_h \in \mathcal{S}, a_h \in \mathcal{A}, h \in \mathbb{N}\}$ induced by the policy π and the initial distribution v_0. We also use $\mathbb{E}_{\pi}$ to denote the expectation with respect to $\mathbb{P}_{v_0}^{\pi}$. As in Inverse Reinforcement learning (IRL), a Dynamic Discrete Choice (DDC) model make the following assumption:

ASSUMPTION C.2. *Agent makes decision according to the policy* $\operatorname{argmax}_{\pi\in\Delta_{\mathcal{A}}^{\mathcal{S}}} \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \beta^h (r(s_h, a_h) + \epsilon_{ah})\right]$.

As Assumption C.2 specifies the agent's policy, we omit π in the notations from now on. Define $\boldsymbol{\epsilon}_h = [\epsilon_{1h} \ldots \epsilon_{|\mathcal{A}|h}]$, where $\epsilon_{ih} \overset{i.i.d}{\sim} G(\delta, 1)$ for $i = 1 \ldots |\mathcal{A}|$. We define a function V as

$$V(s_{h'}, \boldsymbol{\epsilon}_{\boldsymbol{h'}}) = \mathbb{E}\left[\sum_{h=h'}^{\infty} \beta^h (r(s_h, a_h) + \epsilon_{ah}) \mid s_{h'}\right]$$

and call it the value function. According to Assumption C.2, the value function V must satisfy the Bellman equation, i.e.,

$$V(s, \boldsymbol{\epsilon}) = \max_{a\in\mathcal{A}} \left\{ r(s, a) + \epsilon_a + \beta \cdot \mathbb{E}_{s'\sim P(s,a), \boldsymbol{\epsilon}'\sim\boldsymbol{\epsilon}} \left[V(s', \boldsymbol{\epsilon}') \mid s, a \right] \right\}. \tag{28}$$

Define

$$\begin{aligned}\bar{V}(s) &\triangleq E_{\boldsymbol{\epsilon}}\left[V(s,\boldsymbol{\epsilon})\right]\\ Q(s,a) &\triangleq r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a)}\left[\bar{V}(s')\mid s,a\right]\end{aligned} \tag{29}$$

We call $\bar{V}$ the expected value function, and $Q(s,a)$ as the choice-specific value function. Then the Bellman equation can be written as

$$\bar{V}(s)=\mathbb{E}_{\boldsymbol{\epsilon}}\left[\max_{a\in\mathcal{A}}\left\{r(s,a)+\epsilon_a+\beta\cdot\mathbb{E}\left[\bar{V}(s')\mid s,a\right]\right\}\right] \tag{30}$$

$$=\ln\left[\sum_{a\in\mathcal{A}}\exp\left(r(s,a)+\beta\cdot\mathbb{E}\left[\bar{V}(s')\mid s,a\right]\right)\right]+\delta+\gamma \qquad (\because \text{Lemma C.1})$$

$$=\ln\left[\sum_{a\in\mathcal{A}}\exp\left(Q(s,a)\right)\right]+\delta+\gamma \tag{31}$$

Furthermore, Corollary C.2 characterizes that the agent's optimal policy is characterized by

$$\pi^*(a\mid s)=\frac{\exp\left(Q(s,a)\right)}{\sum_{a'\in\mathcal{A}}\exp\left(Q(s,a')\right)}\text{ for }a\in\mathcal{A} \tag{32}$$

In addition to Bellman equation in terms of value function V in equation 28, Bellman equation in terms of choice-specific value function Q comes from combining equation 29 and equation 31:

$$Q(s,a)=r(s,a)+\beta\mathbb{E}_{s'\sim P(s,a)}\left[\ln\left(\sum_{a'\in\mathcal{A}}\exp\left(Q\left(s',a'\right)\right)\right)\mid s,a\right]+\delta+\gamma \tag{33}$$

When $\delta=-\gamma$ (i.e., the Gumbel noise is mean 0), we have

$$Q(s,a)=r(s,a)+\beta\mathbb{E}_{s'\sim P(s,a)}\left[\ln\left(\sum_{a'\in\mathcal{A}}\exp\left(Q\left(s',a'\right)\right)\right)\mid s,a\right] \tag{34}$$

This Bellman equation can be also written in another form.

$$\begin{aligned}Q(s,a) &\triangleq r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a)}\left[\bar{V}(s')\mid s,a\right] && \text{(Equation 29)}\\ &=r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a),\boldsymbol{\epsilon}'\sim\boldsymbol{\epsilon}}\left[V(s',\boldsymbol{\epsilon}')\mid s,a\right]\\ &=r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a),\boldsymbol{\epsilon}'\sim\boldsymbol{\epsilon}}\left[\max_{a'\in\mathcal{A}}\left(Q(s',a')+\epsilon'_a\right)\mid s,a\right] && (35)\\ &=r(s,a)+\beta\cdot\mathbb{E}_{s'\sim P(s,a),a'\sim\pi^*(\cdot|s')}\left[Q(s',a')+\delta+\gamma-\log\pi^*(a'\mid s')\mid s,a\right] && \text{(Corollary C.4)}\\ & && (36)\end{aligned}$$

where $\pi^*(s,a)=\left(\frac{Q(s,a)}{\sum_{a'\in\mathcal{A}}Q(s,a')}\right)$.

C.5 Equivalence between DDC and Entropy regularized IRL

Equation 24, equation 26 and equation 27 characterizes the choice-specific value function's Bellman equation and optimal policy in entropy regularized IRL setting when regularizing coefficient is 1:

$$Q(s,a) = r(s,a) + \beta \mathbb{E}_{s' \sim P(s,a)} \left[\ln \left(\sum_{a' \in \mathcal{A}} \exp\left(Q\left(s', a'\right)\right) \right) \mid s, a \right]$$

$$\pi^*(a \mid s) = \frac{\exp\left(Q(s,a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(Q(s,a')\right)} \text{ for } a \in \mathcal{A}$$

$$Q(s,a) = r\left(s,a\right) + \beta \cdot \mathbb{E}_{s' \sim P(s,a), a' \sim \pi^*(\cdot \mid s')} \left[Q(s',a') - \log \pi^*(a' \mid s') \mid s, a \right]$$

Equation 32, equation 34, and equation 36 (when $\delta = -\gamma$) characterizes the choice-specific value function's Bellman equation and optimal policy of Dynamic Discrete Choice setting:

$$Q(s,a) = r(s,a) + \beta \mathbb{E}_{s' \sim P(s,a)} \left[\ln \left(\sum_{a' \in \mathcal{A}} \exp\left(Q\left(s', a'\right)\right) \right) \mid s, a \right]$$

$$\pi^*(a \mid s) = \frac{\exp\left(Q(s,a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(Q(s,a')\right)} \text{ for } a \in \mathcal{A}$$

$$Q(s,a) = r\left(s,a\right) + \beta \cdot \mathbb{E}_{s' \sim P(s,a), a' \sim \pi^*(\cdot \mid s')} \left[Q(s',a') - \log \pi^*(a' \mid s') \mid s, a \right]$$

Q that satisfies 32 is unique [Rust, 1994], and $Q - r$ forms a one-to-one relationship. Therefore, the exact equivalence between these two setups implies that the same reward function r and discount factor β will lead to the same choice-specific value function Q and the same optimal policy for the two problems.

D IRL with occupancy matching

[Ho and Ermon, 2016] defines another inverse reinforcement learning problem that is based on the notion of occupancy matching. Let ν_0 be the initial state distribution and d^π be the discounted state-action occupancy of π which is defined as $d^\pi = (1-\beta) \sum_{t=0}^{\infty} \beta^t d_t^\pi$, with $d_t^\pi(s,a) = \mathbb{P}_{\pi, \nu_0}\left[s_t = s, a_t = a\right]$. Note that $Q^\pi(s,a) := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \beta^t r(s_t, a_t) \mid s_0 = s, a_0 = a\right] = \sum_{t=0}^{\infty} \beta^t \mathbb{E}_{(\tilde{s},\tilde{a}) \sim d_t^\pi}\left[r(\tilde{s},\tilde{a}) \mid s_0 = a, a_0 = a\right]$. Defining the discounted state-action occupancy of the expert policy π^* as d^*, [Ho and Ermon, 2016] defines the inverse reinforcement learning problem as the following max-min problem:

$$\max_{r \in C} \min_{\pi \in \Pi} \left(\mathbb{E}_{d^*}[r(s,a)] - \mathbb{E}_{d^\pi}[r(s,a)] - \mathcal{H}(\pi) - \psi(r)\right) \tag{37}$$

where $\mathcal{H}$ is the Shannon entropy we used in MaxEnt-IRL formulation and ψ is the regularizer imposed on the reward model r.

Would occupancy matching find Q that satisfies the Bellman equation? Denote the policy as π^* and its corresponding discounted state-action occupancy measure as $d^* = (1-\beta) \sum_{t=0}^{\infty} \beta^t d_t^*$, with $d_t^*(s,a) = \mathbb{P}_{\pi^*, \nu_0}\left[s_t = s, a_t = a\right]$. We define the expert's action-value function as $Q^*(s,a) := \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty} \beta^t r(s_t, a_t) \mid s_0 = s, a_0 = a\right]$ and the Bellman operator of π^* as $\mathcal{T}^*$. Then we have the following Lemma D.1 showing that occupancy matching (even without regularization) may not minimize Bellman error for every state and action.

LEMMA D.1 (OCCUPANCY MATCHING IS EQUIVALENT TO NAIVE WEIGHTED BELLMAN ERROR SUM). *The perfect occupancy matching given the same* (s_0, a_0) *satisfies*

$$\mathbb{E}_{(s,a)\sim d^*}[r(s,a) \mid s_0, a_0] - \mathbb{E}_{(s,a)\sim d^\pi}[r(s,a) \mid s_0, a_0] = \mathbb{E}_{(s,a)\sim d^*}[(\mathcal{T}^* Q^\pi - Q^\pi)(s,a) \mid s_0, a_0]$$

PROOF. Note that $\mathbb{E}_{(s,a)\sim d^*}[r(s,a) \mid s_0, a_0] = \sum_{t=0}^{\infty} \beta^t \mathbb{E}_{(s,a)\sim d_t^*}[r(s,a) \mid s_0, a_0] = Q^*(s,a)$ and $\mathbb{E}_{(s,a)\sim d^\pi}[r(s,a) \mid s_0, a_0] = \sum_{t=0}^{\infty} \beta^t \mathbb{E}_{(s,a)\sim d_t^\pi}[r(s,a) \mid s_0, a_0] = Q^\pi(s,a)$. Therefore

$$\begin{aligned}
&\mathbb{E}_{(s,a)\sim d^*}[r(s,a) \mid s_0, a_0] - \mathbb{E}_{(s,a)\sim d^\pi}[r(s,a) \mid s_0, a_0] = (1-\beta)Q^*(s_0,a_0) - (1-\beta)Q^\pi(s_0,a_0) \\
&= (1-\beta)\frac{1}{1-\beta}\mathbb{E}_{(s,a)\sim d^*}[(\mathcal{T}^* Q^\pi - Q^\pi)(s,a) \mid s_0, a_0] && \text{(Lemma D.2)} \\
&= \mathbb{E}_{(s,a)\sim d^*}[(\mathcal{T}^* Q^\pi - Q^\pi)(s,a) \mid s_0, a_0]
\end{aligned}$$

□

Lemma D.1 implies that occupancy measure matching, even without reward regularization, does not necessarily imply Bellman errors being 0 for every state and action. In fact, what they minimize is the *average Bellman error* [Jiang et al., 2017, Uehara et al., 2020]. This implies that r cannot be inferred from Q using Bellman equation after deriving Q using occupancy matching.

LEMMA D.2 (BELLMAN ERROR TELESCOPING). *Let the Bellman operator* $\mathcal{T}^\pi$ *is defined to map* $f \in \mathbb{R}^{S\times A}$ *to* $\mathcal{T}^\pi f := r(s,a) + \mathbb{E}_{s'\sim P(s,a), a'\sim\pi(\cdot|s')}[f(s',a') \mid s,a]$. *For any* π, *and any* $f \in \mathbb{R}^{S\times A}$,

$$Q^\pi(s_0,a_0) - f(s_0,a_0) = \frac{1}{1-\beta}\mathbb{E}_{(s,a)\sim d^\pi}[(\mathcal{T}^\pi f - f)(s,a) \mid s_0, a_0].$$

PROOF. Note that the right hand side of the statement can be expanded as

$$\begin{aligned}
&r(s_0,a_0) + \beta \cancel{\mathbb{E}_{s'\sim P(s,a), a'\sim\pi(\cdot|s')}[f(s',a') \mid s,a]} - f(s_0,a_0) \\
&+ \beta\mathbb{E}_{(s,a)\sim d_1^\pi}\left[r(s,a) + \beta \cancel{\mathbb{E}_{s'\sim P(s,a), a'\sim\pi(\cdot|s')}[f(s',a') \mid s,a]} - \cancel{f(s,a)} \mid s_0, a_0\right] \\
&+ \beta^2\mathbb{E}_{(s,a)\sim d_2^\pi}\left[r(s,a) + \beta \cancel{\mathbb{E}_{s'\sim P(s,a), a'\sim\pi(\cdot|s')}[f(s',a') \mid s,a]} - \cancel{f(s,a)} \mid s_0, a_0\right] \\
&\qquad \cdots \\
&= Q^\pi(s_0,a_0) - f(s_0,a_0)
\end{aligned}$$

which is the left-hand side of the statement.

□