Factorized Decision Forecasting via Combining Value-based and Reward-based Estimation

Brian D. Ziebart
Carnegie Mellon University
Pittsburgh, PA 15213
bziebart@cs.cmu.edu

***Abstract*— A powerful recent perspective for predicting sequential decisions learns the parameters of decision problems that produce observed behavior as (near) optimal solutions. Under this perspective, behavior is explained in terms of utilities, which can often be defined as functions of state and action features to enable generalization across decision tasks. Two approaches have been proposed from this perspective: estimate a feature-based reward function and recursively compute values from it, or directly estimate a feature-based value function. In this work, we investigate the combination of these two approaches into a single learning task using directed information theory and the principle of maximum entropy. This enables uncovering which type of estimate is most appropriate—in terms of predictive accuracy and/or computational benefit—for different portions of the decision space.**

I. Introduction

For many tasks, human-generated decisions are the best examples of intelligent behavior available and are a valuable source for training machines to behave intelligently. Early approaches for learning to imitate intelligent behavior directly estimate observed policies as a *behavior cloning* task. Perhaps the most successful example is ALVINN [20], the autonomous vehicle trained to control the driving wheel's steering angle using a neural network with forward-mounted video feeds as input. This reduces decision prediction to a classification task. Though successful in keeping the vehicle safely on the roadway, this model of decision making as solely the response to immediate sensor data is limited in its capabilities. For example, it would not be successful in choosing a sequence of roadways leading to a particular new destination. In other words, it is incapable of higher-level decision making, like sequential planning.

More recent techniques for learning and predicting sequential decisions have connected the learning task to decision theory and control theory frameworks. In these frameworks, an instantaneous *reward* is received by taking a particular action in a particular state and it is related to the expected action *value* of cumulative future rewards for each state-action combination according to the Bellman equation [3]. *Inverse optimal control* [13], [4], [19], [1] attempts to find rewards and values that explain observed behavior, following Kalman's early question: "When is a linear control system optimal [13]?" Unfortunately, the question is ill-posed; many choices for the reward function will make observed sequences of decisions optimal, including degeneracies that make all decision sequences equally good [19], but are uninformative. A number of techniques have been developed that resolve the ambiguities of this original question. Most have focused on estimating the reward function based on features of the state and action [5], [1], [22], [2], [21], [18], [25], [27]. However, a recent approach has employed this same perspective to learn the value function based on state and action features [15]. Inverse optimal control approaches have been successfully employed for vehicle [29] and robotic navigation applications [30], [23], as well as cognitive science models [2].

In this paper, we investigate a new approach for combining reward-based and value-based decision learning. We assume that the desirability of each state's actions motivates observed behavior either as a value (a function of state-action or state features that is the ultimate motivator of actions) or as a cost (a combination of a direct function of state-action features and the influence of future expected features in the decision process—i.e., a future value). Portions of the state space that are value-based effectively factorize the decision forecasting task, since decision leading to those states do not consider the future beyond the value-based actions of those states. This factorization can dramatically improve the computational efficiency of inverse optimal control while retaining many of the advantages of non-myopic reasoning over smaller cost-based portions of the decision space.

We formulate the combination of reward-based and value-based inverse optimal control as a maximum causal entropy optimization [27]. We then investigate parameter and structure learning tasks, as well as prediction tasks, under the resulting model. We show that given the state-based factorization of the decision problem, reward and value parameters can be efficiently learned as a convex optimization. We then pose the problem of deciding whether each state influences decisions as a value-based or as a reward-based factor from a Bayesian perspective. We present structure learning algorithms for obtaining the posterior belief of value-based versus reward-based influence for each state from demonstrated behavior and known value and reward weights. To accomplish this, we introduce a technique based on Markov chain Monte Carlo simulation [8]. Combining these two procedures, we introduce an expectation-maximization [6] approach for learning the combined influence type of states and the corresponding reward and value parameters.

II. Background and Related Work

We begin by reviewing decision making frameworks, optimal decision criteria, and inverse optimal control learning techniques. Our combined value-based and reward-based maximum causal entropy inverse optimal control approach builds upon these concepts.

A. Decision Processes and Optimal Control

Markov decision processes (Definition 1) provide a flexible representation of sequential decision making.

Definition 1: A **Markov decision process** (MDP) is a tuple, $\mathcal{M}_{\text{MDP}} = (S, A, P(s'|s,a), R(s,a))$, of:

- A set of **states** ($s \in S$);
- A set of **actions** ($a \in A$) associated with states;
- Action-dependent **state transition dynamics** probability distributions ($P(s'|s,a)$) specifying a next state (s'); and
- A **reward function** ($R(s,a) \rightarrow \mathbb{R}$).

At each timestep t, the state (S_t) is generated from the transition probability distribution (based on the previous state S_{t-1} and previous action A_{t-1}). The state is known to the decision maker before the next action (A_t) is selected. The distribution from which decisions are drawn is the (stochastic) **policy**, $P(A|S)$ or $\pi(A|S)$.

The standard problem of interest, given an MDP, is to find the **optimal policy** which maximizes the cumulative expected reward:

$$\pi(A|S) = \underset{\pi(A|S)}{\operatorname{argmax}} \mathbb{E}_{P(\mathbf{A},\mathbf{S})} \left[\sum_t R(a_t, s_t) \middle| \pi \right].$$

The Bellman equation,

$$\pi(s) = \underset{a}{\operatorname{argmax}}\, Q(a,s) \tag{1}$$

$$Q(a,s) = R(a,s) + \mathbb{E}_{P(S'|s,a)}\left[V(s')\right] \tag{2}$$

$$V(s) = \max_a Q(a,s), \tag{3}$$

defines a fixed point for this optimal policy. The recurrences of Equation 2 and Equation 3 can be iteratively applied to compute the state-action values (also known as the reward-to-go), $Q(a,s)$, and the state values, $V(s)$, via the **value iteration** algorithm [3]. These values quantify the cumulative future expected reward received by the optimal policy from invoking a particular action in a particular state or from a particular state, respectively. Under this optimality criteria, a deterministic policy can always be obtained that provides the optimal values computed via the Bellman equation. Often, the t^{th} reward obtained in the MDP is discounted by a factor of γ^t ($0 \leq \gamma < 1$) to ensure fast convergence when applying the Bellman equation. However, this can be equivalently represented by scaling all transition probabilities $P(s'|a,s)$ by a factor of γ and terminating after each action with probability $(1-\gamma)$. Thus, we do not explicitly consider the discounted reward setting in out formulation.

In practice, computing the Bellman equation in decision spaces with large numbers of states and actions can be computationally burdensome. One approach to tame this complexity is to intelligently order the state updates of the Bellman equation that are applied and to employ heuristics that bound the value functions, limiting the size of the decision space needed to be considered. This approach is famous for its use in planning problems and is known as the A* search algorithm [10]. The value-based inverse optimal control portion of our approach can be viewed as providing related computational benefits.

B. Inverse Optimal Control

Inverse optimal control (IOC) investigates the problem of determining what reward function best explains demonstrated behavior sequences of states $\mathbf{s} = (s_1, \ldots, s_T) \in \boldsymbol{\mathcal{S}}$ and actions $\mathbf{a} = (a_1, \ldots, a_T) \in \boldsymbol{\mathcal{A}}$. Typically, the parameters of linear reward functions,

$$R_\theta(a,s) = \theta^\top \mathbf{f}_{a,s}, \tag{4}$$

which are defined in terms of real-valued feature vectors, $\mathbf{f}_{a,s} \in \mathbb{R}^K$, are learned. A broader view of inverse optimal control allows any of the underlying Bellman variables of Equations 1–3 to be estimated. This leads to three types of estimates:

- **Action value estimation**, $Q(a,s) = \phi_Q^\top \mathbf{f}_{a,s}$, from which the policy can be obtained via Equation 1.
- **State value estimation**, $V(s) = \phi_V^\top \mathbf{f}_s$, from which the policy can be obtained via Equation 1 and Equation 2.
- **Reward estimation**, $R(a,s) = \theta^\top \mathbf{f}_{s,a}$, from which the policy can be recursively obtained via Equations 1-3.

Unfortunately, the naïve problem formulation—find reward parameters θ or value parameters ϕ_Q or ϕ_V that make all demonstrated behavior optimal—is ill-posed; many choices of these parameters, including degeneracies will achieve this objective. Consider the zero weight vector, $\theta = \phi_Q = \phi_V = \mathbf{0}$. It makes all decision choices, including demonstrated decision sequences, have an equal value of zero. While this satisfies the naïve formulation, it is a completely uninformative solution.

Early approaches employed heuristics to select the more meaningful reward function weight solutions to the inverse optimal control problem [19]. However, the choice of heuristic is not particularly well-justified and often the degenerate solution is the only valid solution in this problem formulation when demonstrated sequences are not perfectly predictable. A key insight of Abbeel & Ng [1] is that to ensure that an estimated policy matches the performance of a demonstrated (sample from a) policy on a decision maker's unknown reward function, the expected features must match. Following linearity,

$$\mathbb{E}_{\hat{P}}\left[\sum_t \mathbf{f}_{a_t,s_t}\right] = \mathbb{E}_{\tilde{P}}\left[\sum_t \mathbf{f}_{a_t,s_t}\right]$$

$$\Leftrightarrow \forall \theta \in \mathbb{R}^K,\ \theta^\top \mathbb{E}_{\hat{P}}\left[\sum_t \mathbf{f}_{a_t,s_t}\right] = \theta^\top \mathbb{E}_{\tilde{P}}\left[\sum_t \mathbf{f}_{a_t,s_t}\right]$$

$$\Leftrightarrow \forall \theta \in \mathbb{R}^K,\ \mathbb{E}_{\hat{P}}\left[\sum_t R_\theta(a_t,s_t)\right] = \mathbb{E}_{\tilde{P}}\left[\sum_t R_\theta(a_t,s_t)\right],$$

we can see that this is indeed the case.

Unfortunately, matching features in expectation does not resolve all of the ambiguities in the inverse optimal control problem; when demonstrated decision sequences are not consistently explained as the optimal result of a single choice of reward parameters, θ, then there are many stochastic policies that match feature counts. This corresponds to the situation in which demonstrated decision sequences are not perfectly predictable and degenerate solutions are the only solutions to the naïve inverse optimal control formulation. When learning from human-generate decision sequences, this is often the case—not necessarily because people are non-optimal, but also because a MDP and state-action features are often a simplification of the decision task and its influences. Abbeel & Ng [1] mix a set of deterministic policies that are the optimal policies for a sequence of reward parameters, $\{\theta_i\}$:

$$\sum_i \mathbb{E}_{P(\mathbf{S},\mathbf{A})}\left[\sum_{t=1}^{T}\mathbf{f}_{s_t,a_t}\middle|\pi_{\theta_i}(A|S)\right] \approx \mathbb{E}_{\tilde{P}(\mathbf{S},\mathbf{A})}\left[\sum_{t=1}^{T}\mathbf{f}_{s_t,a_t}\right].$$

However, there are many other ways to find such a feature-matching policy (mixture), and many do not provide good predictive performance. Indeed, very simple examples have been shown to have infinite log-loss under the Abbeel & Ng approach [28].

Other approaches to inverse optimal control employ game-theoretic parameter selection criteria for obtaining reward function parameters [25], maximum margin techniques [22], or Bayesian posterior distributions of reward parameters [5]. A Boltzmann action distribution approach [2], [21], [18] is similar to the maximum entropy techniques we use in this paper. It employs a distribution over actions based on the state-action value,

$$P(a|s) \propto e^{Q_\theta(a,s)}, \tag{5}$$

where the $Q(a,s)$ action values are computed from the Bellman equation (Equation 1) with a linear reward function (Equation 4). Unfortunately, the data likelihood under this model is not convex, so optimization techniques for finding reward weights θ are subject to local optima. Additionally, the policy with the largest expected reward does not necessarily have the largest probability in this model [28].

C. Reward-Based Maximum Causal Entropy IOC

The principle of maximum entropy [12] prescribes probability distributions that are the most uncertain subject to constraints matching measured properties of data. This provides robust predictive log-loss minimization guarantees [9]. It has been recently extended to settings with interaction and feedback, making it applicable to inverse optimal control problems [27].

Definition 2: **Reward-based maximum causal entropy inverse optimal control** [27] is defined by the following optimization:

$$\begin{aligned} &\max H(\mathbf{A}||\mathbf{S}) \qquad (6)\\ \text{such that: } &\mathbb{E}_{P(\mathbf{S},\mathbf{A})}\left[\sum_t \mathbf{f}_{s_t,a_t}\right] = \mathbb{E}_{\tilde{P}(\mathbf{S},\mathbf{A})}\left[\sum_t \mathbf{f}_{s_t,a_t}\right],\\ &\forall \mathbf{s}\in\boldsymbol{\mathcal{S}}, \mathbf{a}\in\boldsymbol{\mathcal{A}},\ P(\mathbf{s},\mathbf{a}) = P(\mathbf{a}||\mathbf{s})\ P(\mathbf{s}^T||\mathbf{a}^{T-1})\\ &\forall \mathbf{s}\in\boldsymbol{\mathcal{S}}, \mathbf{a}\in\boldsymbol{\mathcal{A}},\ P(\mathbf{a}||\mathbf{s}) \geq 0 \text{ and}\\ &\forall \mathbf{s}\in\boldsymbol{\mathcal{S}},\ \sum_{\mathbf{a}} P(\mathbf{a}||\mathbf{s}) = 1. \end{aligned}$$

where the causally conditioned entropy [14] from the Marko-Massey theory of directed information [16], [17],

$$H(\mathbf{A}||\mathbf{S}) = \mathbb{E}_{P(\mathbf{A},\mathbf{S})}[-\log P(\mathbf{a}||\mathbf{s})], \tag{7}$$

is based on the causally conditioned probability distribution,

$$P(\mathbf{a}||\mathbf{s}) = \prod_{t=1}^{T} P(a_t|\mathbf{s}_{1:t},\mathbf{a}_{1:t-1}), \tag{8}$$

and its temporal complement,

$$P(\mathbf{s}^T||\mathbf{a}^{T-1}) = \prod_{t=1}^{T} P(s_t|\mathbf{s}_{1:t-1},\mathbf{a}_{1:t-1}). \tag{9}$$

These distributions crucially differ from the standard conditional probability,

$$P(\mathbf{a}|\mathbf{s}) = \prod_{t=1}^{T} P(a_t|\mathbf{s}_{1:T},\mathbf{a}_{1:t-1}), \tag{10}$$

in that each action is conditioned only on previously available state information, $\mathbf{s}_{1:t}$ and not future state information, $\mathbf{s}_{t+1:T}$. This difference is reflected in Markov decision processes where each state is only revealed to the decision maker after previous actions are invoked and policies that condition on future states cannot be executed. Together, the two causally conditioned distributions form the joint distribution: $P(\mathbf{a},\mathbf{s}) = P(\mathbf{a}||\mathbf{s})P(\mathbf{s}^T||\mathbf{a}^{T-1})$, which is explicitly enforced as a constraint in Equation 6. By maximizing this causal entropy measure (Equation 6), this approach provides a robust log-loss estimate for the policy $P(A|S)$ [27].

The solution to this optimization (Definition 2) factors into a stochastic policy with actions distributed according to $P(a|s) \propto e^{Q_\theta^{\text{soft}}(a,s)}$, where $Q_\theta^{\text{soft}}(a,s)$ is recursively defined as follows:

$$\begin{aligned} Q_\theta^{\text{soft}}(a,s) &= R_\theta(a,s) + \mathbb{E}_{P(S'|s,a)}\left[V_\theta^{\text{soft}}(s)\right] \qquad (11)\\ V_\theta^{\text{soft}}(s) &= \operatorname*{softmax}_a Q_\theta^{\text{soft}}(a,s), \end{aligned}$$

with reward function $R_\theta(a,s) = \theta^\top \mathbf{f}_{s,a}$ and $\operatorname{softmax}_x f(x) = \log\sum_x e^{f(x)}$. This recursive relationship can be viewed as a smooth relaxation of the Bellman equation (Equation 1). The value updates of Equation 11, like the Bellman equation, can be iteratively applied via dynamic programming to obtain the stochastic policy of the model. This distribution is equivalent to maximum likelihood estimation under a multivariate extreme-value noise distribution for reward values in an MDP [24], but can be applied more broadly to settings in which appropriate

noise terms are difficult to specify (*e.g.*, continuous control with linear dynamics and quadratic costs [27]).

The reward function parameters can be learned using standard gradient-based optimization techniques. The gradient is simply the difference between empirical and expected features:

$$\nabla_\theta \log L(\theta|\tilde{P}(\mathbf{S},\mathbf{A})) = \mathbb{E}_{\tilde{P}(\mathbf{S},\mathbf{A})}\left[\sum_t \mathbf{f}_{s_t,a_t}\right] - \mathbb{E}_{P_\theta(\mathbf{S},\mathbf{A})}\left[\sum_t \mathbf{f}_{s_t,a_t}\right].$$

Due to the convexity of the log-likelihood function, $\log L(\theta|\tilde{P}(\mathbf{S},\mathbf{A})) = \sum_{\mathbf{A},\mathbf{S}} \tilde{P}(\mathbf{A},\mathbf{S}) \log P_\theta(\mathbf{A}||\mathbf{S})$, reward parameters converging to a global optima are obtained using *e.g.*, the gradient ascent algorithm.

D. Value-based maximum entropy inverse optimal control

An alternate approach learns a value function rather than a reward function [15]. This approach can also be posed as a maximum entropy estimation task and is closely related to logistic regression.

Definition 3: **Action value-based maximum entropy inverse optimal control** is defined by the following optimization:

$$\begin{aligned} &\max H(A|S) \\ \text{such that: } &\mathbb{E}_{P(S,A)}\left[\mathbf{f}_{s,a}\right] = \mathbb{E}_{\tilde{P}(S,A)}\left[\mathbf{f}_{s,a}\right] \\ &\forall a \in \mathcal{A}, s \in \mathcal{S}\ P(a|s) \geq 0 \text{ and} \\ &\forall s \in \mathcal{S},\ \sum_{a\in\mathcal{A}} P(a|s) = 1, \end{aligned} \tag{12}$$

where the conditional entropy is: $H(A|S) = \mathbb{E}_P[-\log P(A|S)]$. The conditional probability distribution of actions is of the form: $P(a|s) \propto e^{Q(a,s)}$ where $Q(a,s) = \phi_Q^\top \mathbf{f}_{s,a}$. Note that the distribution is over a single state and action pair.

Action distributions from state-based values are also possible as a result of a maximum entropy formulation.

Definition 4: **State value-based maximum entropy inverse optimal control** [15] is defined by the following optimization:

$$\begin{aligned} &\max H(A|S) \\ \text{such that: } &\mathbb{E}_{P(S',S,A)}\left[\mathbf{f}_{s'}\right] = \mathbb{E}_{\tilde{P}(S',S,A)}\left[\mathbf{f}_{s'}\right] \\ &\forall a \in \mathcal{A}, s \in \mathcal{S}\ P(a|s) \geq 0 \text{ and} \\ &\forall s \in \mathcal{S},\ \sum_{a\in\mathcal{A}} P(a|s) = 1, \end{aligned} \tag{13}$$

where S' is the next state experienced after employing action A in state S, and is distributed according to the known decision process dynamics $P(S'|S,A)$. Both estimates can similarly be obtained using standard gradient-based optimization techniques that avoid local optima as a consequence of convexity.

A key advantage of the value-based estimates is that the policy can be directly obtained without requiring a potentially computationally expensive dynamic programming task (Equation 11). However, learned value functions can typically be less generally applied to *e.g.*, differences in the goal terminal state of the MDP, for which the reward-based approach is more appropriate.

E. Inverse Optimal Heuristic Control

In our previous work, we combined reward-based, IOC-style learning with instantaneous influence learning, similar to the value-based perspective, by augmenting learned reward functions having long-term influence with learned value functions having only instantaneous influence. Actions under this model are distributed according to:

$$P(a|s) \propto e^{Q_\theta^{\text{soft}}(a,s)+Q_\phi^{\text{inst}}(a,s)}, \tag{14}$$

where the state-action values, $Q_\theta^{\text{soft}}(a,s)$, are recursively computed from the dynamic program of Equation 11 and the instantaneous value is a linear function of state-action features, $Q_\phi^{\text{inst}}(a,s) = \phi^\top \mathbf{f}_{s,a}$, and is not recursively related to the state-action values $Q_\theta^{\text{soft}}(s,a)$ associated with the learned reward function. Richer features can be employed within the instantaneous value function (*e.g.*, characteristics of the previous state-action sequence) than can be efficiently incorporated in the reward function.

While this approach improves upon the predictive capabilities of reward-based maximum causal entropy inverse optimal control [27], many of the niceties of the maximum entropy formulation are lost in this formulation. First and foremost, learning reward and value function parameters θ and ϕ is no longer a convex optimization task. Therefore, practical algorithms for learning model parameters are susceptible to local optima. Additionally, it does not improve upon the computational complexity of the underlying maximum causal entropy approach it employs by combining the long-term and instantaneous rewards in this manner.

In contrast, the approach we introduce in this paper permits each state's actions to influence behavior as a reward or as a value, but not simultaneously a combination of both. This retains the convexity properties of the maximum entropy formulation. It also provides computational benefits, as the Bellman-like inference procedure is effectively restricted to smaller portions of the decision space.

III. A Unifying Statistical Estimation Framework for Reward-based and Value-based Inverse Optimal Control

We leverage the recently developed maximum causal entropy framework [27] to pose a state-dependent combination of reward-based and value-based inverse optimal control under a unified estimation task. We then introduce algorithms for inference and learning in the resulting model.

A. A Unified Estimation Formulation

We combine value-based and reward-based inverse optimal control by partitioning the states into three subsets: one of reward-based estimates, $\mathcal{S}_R$, one of action value-based estimates, $\mathcal{S}_Q$, and one of state value-based estimates, $\mathcal{S}_V$. Each state $s \in \mathcal{S}$ has a type, denoted type(s), indicating to which of these sets it belongs. We denote the set of types for all states as type$(\mathcal{S})$. Using these subsets of states, we

view decision sequences as shorter subsequences according to Definition 5[1].

Definition 5: A **value-state-segmented behavior sequence** is a transformation of a sequence of states and actions into a set of smaller subsequences in which only the final state of each sub-sequence can be a value-based state.

For example, a particular state-action sequence, $(s_a, a_x, s_b, a_y, s_c, a_x, s_d, a_z, s_a, a_y, s_b)$, with action value-based set $\mathcal{S}_Q = \{s_b\}$ and state value-based set $\mathcal{S}_V = \{s_d\}$ is segmented into sequences: (s_a, a_x, s_b, a_y), (s_c, a_x, s_d, a_z), and (s_a, a_y, s_b). Our view of demonstrated decision sequences and distributions over decision sequences considers only these segmented subsequences. We make use of the empirical distribution of the initial sub-sequence state, $\tilde{P}_{\text{subseq}}(S_1)$, which, in our example, would be: $\tilde{P}_{\text{subseq}}(S_1 = s_a) = \frac{2}{3}$ and $\tilde{P}_{\text{subseq}}(S_1 = s_c) = \frac{1}{3}$.

Definition 6: **Hybrid reward and value maximum causal entropy inverse optimal control** given reward-based states $\mathcal{S}_R$ and value-based states $\mathcal{S}_V$ and all state-action sequences segmented according to Definition 5 has actions with distributions obtained from the following optimization:

$$\{P(A|S)\} = \underset{\{P(A|S)\}}{\operatorname{argmax}}\, H(\mathbf{A}||\mathbf{S}) \tag{15}$$

$$\text{such that: } \mathbb{E}_{P(\mathbf{S},\mathbf{A})}\left[\sum_{t=1}^{T-1} \mathbf{f}_{s_t,a_t}\right] = \mathbb{E}_{\tilde{P}(\mathbf{S},\mathbf{A})}\left[\sum_{t=1}^{T-1} \mathbf{f}_{s_t,a_t}\right]$$

$$\mathbb{E}_{P(\mathbf{S},\mathbf{A})}\big[I_{\mathcal{S}_Q}(s_T)\mathbf{f}_{s_T,a_T}\big] = \mathbb{E}_{\tilde{P}(\mathbf{S},\mathbf{A})}\big[I_{\mathcal{S}_Q}(s_T)\mathbf{f}_{s_T,a_T}\big]$$

$$\mathbb{E}_{P(\mathbf{S},\mathbf{A})}\left[I_{\mathcal{S}_V}(s_T)\sum_{s'\in\mathcal{S}} P(s'|s_T,a_T)\,\mathbf{f}_{s'}\right]$$

$$= \mathbb{E}_{\tilde{P}(\mathbf{S},\mathbf{A})}\left[I_{\mathcal{S}_V}(s_T)\sum_{s'\in\mathcal{S}} P(s'|s_T,a_T)\,\mathbf{f}_{s'}\right]$$

$$\forall \mathbf{s}\in\boldsymbol{\mathcal{S}}, \mathbf{a}\in\boldsymbol{\mathcal{A}},\ P(\mathbf{s},\mathbf{a}) = P(\mathbf{a}||\mathbf{s})\ P(\mathbf{s}^T||\mathbf{a}^{T-1})$$

$$\forall \mathbf{s}\in\boldsymbol{\mathcal{S}}, \mathbf{a}\in\boldsymbol{\mathcal{A}}\ P(\mathbf{a}||\mathbf{s}) \geq 0,$$

$$\forall \mathbf{s}\in\boldsymbol{\mathcal{S}} \sum_{\mathbf{a}\in\boldsymbol{\mathcal{A}}} P(\mathbf{a}||\mathbf{s}) = 1 \text{ and}$$

$$\forall s_1 \in \mathcal{S},\ P(s_1) = \tilde{P}_{\text{subseq}}(s_1),$$

with indicator function $I_{\mathcal{A}}(a) = 1$ if $a \in \mathcal{A}$ and 0 otherwise.

We note that the causal entropy is concave and that the joint sequence distribution $P(\mathbf{S},\mathbf{A})$ is a linear function of unknown causally conditioned variables $P(\mathbf{A}||\mathbf{S})$. This enables the optimization of Definition 6 to be expressed as a convex program.

B. Dual Form and Inference Procedure

Though the convex program of Definition 6 can be directly optimized, the dual program enables a better understanding of the model and supports more compact optimization procedures.

Theorem 1: The hybrid IOC optimization of Definition 6 can be formulated as a softened version of the Bellman equation [3] via convex duality:

$$Q^{\text{hyb}}_{\theta,\phi}(a,s) = \begin{Bmatrix} \theta^\top \mathbf{f}_{s,a} + \mathbb{E}_{P(S'|s,a)}\left[V^{\text{hyb}}_{\theta,\phi}(s')\right] & \text{if } s\in\mathcal{S}_R \\ \phi_Q^\top\, \mathbf{f}_{s,a} & \text{if } s\in\mathcal{S}_Q \\ \sum_{s'\in\mathcal{S}} P(s'|s,a)\ \phi_V^\top\, \mathbf{f}_{s'} & \text{if } s\in\mathcal{S}_V \end{Bmatrix} \tag{16}$$

$$V^{\text{hyb}}_{\theta,\phi}(s) = \operatorname*{softmax}_a Q^{\text{hyb}}_{\theta,\phi}(a,s),$$

with $\operatorname{softmax}_x f(x) = \log\sum_x e^{f(x)}$, the reward function parameterized by reward weights θ, and the value functions parameterized by value weights $\phi = \{\phi_Q, \phi_V\}$.

A straight-forward procedure, Algorithm 1, follows from these softmax Bellman-like updates. For simplicity, we assume in our notation that the decision time-horizon is infinite and that the corresponding policies are time-invariant. Extensions to time-varying, fixed horizon decision settings are straight-forward, but notationally more cumbersome.

Algorithm 1 Reward/value-based IOC inference procedure

Require: MDP $\mathcal{M}_{\text{MDP}}$, reward parameters θ, action value parameters ϕ_Q, state value parameters ϕ_V, action value-based set $\mathcal{S}_Q$, state value-based set $\mathcal{S}_V$, and reward-based set $\mathcal{S}_R$.

Ensure: State-action values $Q_{\theta,\phi}(a,s)$ and state values $V_{\theta,\phi}(s)$ according to Equation 16

for all $s \in \mathcal{S}_Q$ **do**
  for all $a \in \mathcal{A}$ **do**
    $Q_{\theta,\phi}(a,s) \leftarrow \phi_Q^\top\, \mathbf{f}_{s,a}$
  end for
end for
for all $s \in \mathcal{S}_V$ **do**
  for all $a \in \mathcal{A}$ **do**
    $Q_{\theta,\phi}(a,s) \leftarrow \phi_V^\top\, \mathbf{f}_{s,a}$
  end for
end for
while not converged **do**
  for all $s \in \mathcal{S}_R$ **do**
    $V_{\theta,\phi}(s) \leftarrow \operatorname{softmax}_a Q(a,s)$
  end for
  for all $s \in \mathcal{S}_R$ **do**
    for all $a \in \mathcal{A}$ **do**
      $Q_{\theta,\phi}(a,s) \leftarrow \theta^\top\, \mathbf{f}_{s,a} + \mathbb{E}_{P(S'|s,a)}[V_{\theta,\phi}(s')|s,a]$
    end for
  end for
end while

The computational benefit of replacing some reward-based states with value-based states can be understood from considering the longest paths of the decision space. In the worst case, value iteration requires $\mathcal{O}(|S|^2|A|)$ time—quadratic in the number of states—to obtain the optimal infinite time-horizon policy. Intuitively, this is because the best policy could correspond to a path of length $\mathcal{O}(|S|)$ and propagating values over each state of that path requires $\mathcal{O}(|S||A|)$ time via value iteration. Replacing reward-based states with value-based states so that there are no paths consisting entirely

[1]For simplicity, we assume that the final state of each original sequence is a value state. However, this assumption can be relaxed with only additional notational complexity.

of reward-based states of length bounded below $\mathcal{O}(|S|^{\frac{1}{2}})$ or $\mathcal{O}(\log|S|)$ reduces the value iteration to $\mathcal{O}(|S|^{\frac{3}{2}}|A|)$ or $\mathcal{O}(|S||A|\log|S|)$ time.

The inference procedure (Algorithm 1) can be generalized to settings in which there is a belief about the value-based state set, $P(\mathcal{S}_V)$. Naïvely, the resulting policy can be obtained for each belief combination for value state sets:

$$P(a|s) = \sum_{\mathcal{S}_V \subseteq \mathcal{S}} P(\mathcal{S}_V) P_{\theta,\phi}(a|s, \mathcal{S}_V). \tag{17}$$

However, a more efficient algorithm is obtained when beliefs in individual state types, $P(\text{type}(s))$ are independent between states (Theorem 2) and we assume that decisions are made under this same uncertainty.

Theorem 2: When state type beliefs are independent, $P(\text{type}(\mathcal{S})) = \prod_{s\in\mathcal{S}} P(\text{type}(s))$, the state-action value of Equation 16 can be re-written as:

$$\begin{aligned} Q_{\theta,\phi}(a,s) =& P(s \in \mathcal{S}_R)\ (\theta^\top \mathbf{f}_{s,a} + \mathbb{E}[V(s')|s,a]), \\ &+ P(s \in \mathcal{S}_Q)\ \phi_Q^\top \mathbf{f}_{s,a} \\ &+ P(s \in \mathcal{S}_V) \sum_{s' \in \mathcal{S}} P(s'|s,a)\ \phi_V^\top \mathbf{f}_{s'} \end{aligned} \tag{18}$$

with the interpretation that at each visit to a state, a sample from $P(\text{type}(s))$ is drawn indicating whether the state is treated as a reward-based influence, an action value-based influence, or a state value-based influence.

C. Reward and Value Weight Learning from Known State Sets

We now investigate the task of learning reward weights θ and value weights $\phi = \{\phi_Q, \phi_V\}$ given the reward state set. Our objective is to maximize the log likelihood:

$$\begin{aligned} \{\theta^*, \phi_Q^*, \phi_V^*\} &= \underset{\theta,\ \phi_Q,\ \phi_V}{\text{argmax}} \log L(\theta, \phi_Q, \phi_V | \tilde{P}(\mathbf{a}, \mathbf{s})) \\ &= \underset{\theta,\ \phi_Q,\ \phi_Q}{\text{argmax}} \sum_{\mathbf{a}\in\boldsymbol{\mathcal{A}},\ \mathbf{s}\in\boldsymbol{\mathcal{S}}} \tilde{P}(\mathbf{a}, \mathbf{s}) \log P_{\theta,\phi}(\mathbf{a}|\mathbf{s}). \end{aligned}$$

The gradient of this optimization with respect to reward parameters and value parameters is:

$$\begin{aligned} &\nabla_\theta \log L(\theta, \phi_Q, \phi_V | \tilde{P}(\mathbf{S}, \mathbf{A})) = \\ &\quad \mathbb{E}_{\tilde{P}(\mathbf{S},\mathbf{A})}\left[\sum_{t=1}^{T-1} \mathbf{f}_{s_t,a_t}\right] - \mathbb{E}_{P_{\theta,\phi}(\mathbf{S},\mathbf{A})}\left[\sum_{t=1}^{T-1} \mathbf{f}_{s_t,a_t}\right] \end{aligned} \tag{19}$$

$$\begin{aligned} &\nabla_{\phi_Q} \log L(\theta, \phi_Q, \phi_V | \tilde{P}(\mathbf{S}, \mathbf{A})) = \\ &\quad \mathbb{E}_{\tilde{P}(\mathbf{S},\mathbf{A})}\left[I_{s_T \in \mathcal{S}_Q}\ \mathbf{f}_{s_T,a_T}\right] - \mathbb{E}_{P_{\theta,\phi}(\mathbf{S},\mathbf{A})}\left[I_{s_T \in \mathcal{S}_Q}\ \mathbf{f}_{s_T,a_T}\right] \end{aligned} \tag{20}$$

$$\begin{aligned} &\nabla_{\phi_Q} \log L(\theta, \phi_Q, \phi_V | \tilde{P}(\mathbf{S}, \mathbf{A})) = \\ &\quad \mathbb{E}_{\tilde{P}(\mathbf{S},\mathbf{A})}\left[I_{s_T \in \mathcal{S}_V} \sum_{s' \in \mathcal{S}} P(s'|s_T, a_T)\ \mathbf{f}_{s'}\right] \\ &\quad - \mathbb{E}_{P_{\theta,\phi}(\mathbf{S},\mathbf{A})}\left[I_{s_T \in \mathcal{S}_V} \sum_{s' \in \mathcal{S}} P(s'|s_T, a_T)\ \mathbf{f}_{s'}\right]. \end{aligned} \tag{21}$$

Standard gradient-based optimization techniques can be employed to find parameters that converge to a global optimum.

Algorithm 2 Reward/value-based IOC reward/value parameter learning procedure

Require: MDP $\mathcal{M}_{\text{MDP}}$, demonstrated sequences $\tilde{P}(\mathbf{S}, \mathbf{A})$, initial reward parameters θ_0, initial action value parameters $\phi_{Q,0}$, initial state value parameters $\phi_{V,0}$, state types type($\mathcal{S}$).

Ensure: Reward weight estimates $\hat{\theta}$ and value weight estimates $\hat{\phi}$ that are (approximately) optimal maximum likelihood estimates

$\hat{\theta} \leftarrow \theta_0$, $\hat{\phi}_Q \leftarrow \phi_{Q,0}$, $\hat{\phi}_V \leftarrow \phi_{V,0}$
while $\hat{\theta}$, $\hat{\phi}_Q$, and $\hat{\phi}_V$ not (approximately) converged **do**
  Compute $Q_{\hat{\theta},\hat{\phi}}(a,s)$ and $V_{\hat{\theta},\hat{\phi}}(s)$ given type($\mathcal{S}$) via Algorithm 1
  Compute policy $P(a|s) = e^{Q_{\hat{\theta},\hat{\phi}}(a,s) - V_{\hat{\theta},\hat{\phi}}(s)}$
  Compute gradient $\nabla_\theta \log L(\theta, \phi | \tilde{P}(\mathbf{S}, \mathbf{A}))|_{\theta=\hat{\theta}}$ via Equation 19
  Compute gradient $\nabla_{\phi_Q} \log L(\theta, \phi | \tilde{P}(\mathbf{S}, \mathbf{A}))|_{\phi_Q=\hat{\phi}_Q}$ via Equation 20
  Compute gradient $\nabla_{\phi_V} \log L(\theta, \phi | \tilde{P}(\mathbf{S}, \mathbf{A}))|_{\phi_V=\hat{\phi}_V}$ via Equation 21
  $\hat{\theta} \leftarrow \hat{\theta} + \eta_t \nabla_\theta \log L(\theta, \phi | \tilde{P}(\mathbf{S}, \mathbf{A}))$
  $\hat{\phi}_Q \leftarrow \hat{\phi}_Q + \eta_t \nabla_{\phi_Q} \log L(\theta, \phi | \tilde{P}(\mathbf{S}, \mathbf{A}))$
  $\hat{\phi}_V \leftarrow \hat{\phi}_V + \eta_t \nabla_{\phi_V} \log L(\theta, \phi | \tilde{P}(\mathbf{S}, \mathbf{A}))$
end while

One such procedure is shown as Algorithm 2. The learning rate parameters η_t are chosen to slowly decay to 0 at a rate of *e.g.*, $\Theta\left(\frac{1}{\sqrt{t}}\right)$.

D. Inferring the Influence Types of States

Inferring which states are value-based influences of decision making and which are reward-based influences is a more difficult task. Naïvely, considering the powerset of states is computationally burdensome due to the $\Theta(3^{|\mathcal{S}|})$ subset choices. We draw upon approaches that have been successfully employed for Bayesian structure learning [11], which is a similarly difficult task of determining how a joint distribution should factor into a product of conditional probabilities. Specifically, Markov chain Monte Carlo (MCMC) simulation [8], has been employed for Bayesian network structure learning [7] and provides a general approach for addressing the influence type inference setting of this paper as well.

We are interested in obtaining a posterior distribution of state types given available evidence. Here, we represent that evidence as $\tilde{\pi}$, the demonstrated policy. Using Bayes' rule, the posterior distribution is:

$$P(\text{type}(\mathcal{S})|\tilde{\pi}, \theta, \phi) \propto P(\tilde{\pi}|\text{type}(\mathcal{S}), \theta, \phi) P(\text{type}(\mathcal{S})). \tag{22}$$

We rely on a key property of the reward/value-based IOC distribution (Theorem 3) relating the log likelihood to reward and softened state values.

Theorem 3: The log probability of a policy under the hybrid reward-based/value-based inverse optimal control ap-

proach is related to the policy's expected rewards as follows:

$$\log P(\tilde{\pi}|\text{type}(\mathcal{S}),\theta,\phi) = \quad (23)$$
$$\mathbb{E}_{P(\mathbf{S},\mathbf{A})}\left[\sum_{t=1}^{T-1}\theta^\top \mathbf{f}_{s_t,a_t} + \phi^\top \mathbf{f}_{s_T,a_T} - V^{\text{hyb}}_{\theta,\phi}(s_1), \middle| \tilde{\pi}\right]$$

where $V^{\text{hyb}}_{\theta,\phi}$ is defined according to Theorem 1.

Using standard Markov chain Monte Carlo simulation techniques, an approximation to the posterior distribution of reward/value state type given demonstrated policies is obtained using Algorithm 3.

Algorithm 3 Posterior reward/value set MCMC procedure

Require: MDP $\mathcal{M}_{\text{MDP}}$, reward parameters θ, action value parameters ϕ_Q, state value parameters ϕ_V, demonstrated policy $\tilde{\pi}(A|S)$, burn-in size N_b, sample size N_s

Ensure: $\hat{P}(\text{type}(\mathcal{S}))$ based on samples from Bayesian posterior.

1: $\forall s \in \mathcal{S}$, set type$(s)$ = "Action value"
2: $t \leftarrow 1$
3: **while** $t \leq (N_b + N_s)$ **do**
4: Sample $i \in \{1,\ldots,|\mathcal{S}|\}$ uniformly at random
5: Set type(s_i) = "Reward"
6: Compute $P_R = P(\tilde{\pi}|\text{type}(\mathcal{S}),\theta,\phi)\ P(\text{type}(\mathcal{S}))$ via Equation 23
7: Set type(s_i) = "Action value"
8: Compute $P_Q = P(\tilde{\pi}|\text{type}(\mathcal{S}),\theta,\phi)\ P(\text{type}(\mathcal{S}))$ via Equation 23
9: Set type(s_i) = "State value"
10: Compute $P_V = P(\tilde{\pi}|\text{type}(\mathcal{S}),\theta,\phi)\ P(\text{type}(\mathcal{S}))$ via Equation 23
11: Sample type(s_i) from the distribution: Multinomial $\left(\frac{P_R}{P_R+P_Q+P_V}, \frac{P_Q}{P_R+P_Q+P_V}, \frac{P_V}{P_R+P_Q+P_V}\right)$
12: **if** $t > N_b$ **then**
13: Add sample type$(\mathcal{S})$ to distribution $\hat{P}(\text{type}(\mathcal{S}))$
14: **end if**
15: $t \leftarrow t+1$
16: **end while**

Algorithm 3 assumes that the reward and value weights, $\theta, \phi_Q,$ and ϕ_V are known. Often this is not the case. Instead, we would like to estimate both those reward and value weights as well as the type of each state.

Algorithm 4 employs an expectation-maximization approach [6] to obtain both state types and parameter weights. It obtains the maximum likelihood weights for a particular set of state beliefs in line 3 (the *maximization* step) and calculates the expectation of state types given those weight estimates in line 4 (the *expectation* step). When both the expectation and maximization steps are exact, this procedure is guaranteed to converge to a local optima of the likelihood function of reward/value weights and state types. In this setting, such a guarantee can be provided when the Markov chain Monte Carlo procedure of Algorithm 3 mixes well and the true distribution of state types factors independently (as assumed in line 5 of Algorithm 4 for computational efficiency benefits).

Algorithm 4 Reward/value set and parameter EM procedure

Require: MDP $\mathcal{M}_{\text{MDP}}$, demonstrated sequences $\tilde{P}(\mathbf{S},\mathbf{A})$ / policy $\tilde{\pi}$

Ensure: State set estimate $\hat{P}(\text{type}(\mathcal{S}))$ and reward/value parameter estimates $\hat{\theta},\hat{\phi}$ that are (approximately) local optima.

1: $\forall s \in \mathcal{S}$, let $P(\text{type}(s))$ be distributed uniformly at random
2: **while** type$(\mathcal{S})$, $\hat{\theta}$, and $\hat{\phi}$ not converged **do**
3: Obtain reward/value weights $\hat{\theta},\hat{\phi}$ given $\hat{P}(\text{type}(\mathcal{S}))$ and $\tilde{P}(\mathbf{S},\mathbf{A})$ via Algorithm 2 using Equation 18
4: Obtain $\hat{P}(\text{type}(\mathcal{S}))$ given $\tilde{\pi}$ and parameters $\hat{\theta},\hat{\phi}$ via Algorithm 3
5: Approximate $\hat{P}(\text{type}(\mathcal{S}))$ as an independent distribution with $\hat{P}(\text{type}(s_i))$ as the marginal distribution of $\hat{P}(\text{type}(\mathcal{S}))$
6: **end while**

IV. Discussion

In this paper, we have combined reward-based inverse optimal control with value-based inverse optimal control as a statistical estimation task using the principle of maximum causal entropy. We allow each state to influence behavior as either a reward, an action value, or a state value. This allows computationally advantageous value-based estimates to be employed in some portions of the decision space while retaining the sequential reasoning of cost-based estimates in other portions of the state space. When the type of each state is known, learning reward/value weights is accomplished by convex optimization. When state types are unknown, Markov chain Monte Carlo and Expectation-Maximization approaches can be applied with weaker guarantees.

One of the greatest benefits of inverse optimal control is the ability to transfer learned reward/value weights to different decision processes that are characterized by reward/value features from the same feature space. This benefit is seemingly lost by the hybrid formulation of inverse optimal control in this paper because the belief in the type of each state is specific to the decision process of the demonstrated decision sequences and does not transfer. Estimating state types based on available (and transferable) information, such as state-action features, is an important future direction for enabling this approach to the transfer setting.

References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proc. International Conference on Machine Learning*, pages 1–8, 2004.

[2] C. Baker, J. Tenenbaum, and R. Saxe. Goal inference as inverse planning. In *Proceedings of the 29th annual meeting of the cognitive science society*, 2007.

[3] R. Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6:679–684, 1957.

[4] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. Linear matrix inequalities in system and control theory. *SIAM*, 15, 1994.

[5] U. Chajewska, D. Koller, and D. Ormoneit. Learning an agent's utility function by observing behavior. In *International Conference on Machine Learning*, pages 35–42, 2001.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[7] N. Friedman and D. Koller. Being Bayesian about network structure. a Bayesian approach to structure discovery in Bayesian networks. *Machine learning*, 50(1):95–125, 2003.

[8] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, 1996.

[9] P. D. Grünwald and A. P. Dawid. Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory. *Annals of Statistics*, 32:1367–1433, 2003.

[10] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:100–107, 1968.

[11] D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[12] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.

[13] R. Kalman. When is a linear control system optimal? *Trans. ASME, J. Basic Engrg.*, 86:51–60, 1964.

[14] G. Kramer. *Directed Information for Channels with Feedback*. PhD thesis, Swiss Federal Institute of Technology (ETH) Zurich, 1998.

[15] D. Krishnamurthy and E. Todorov. Inverse optimal control with linearly-solvable MDPs. In *Proc. International Conference on Machine Learning*, pages 335–342, 2010.

[16] H. Marko. The bidirectional communication theory – a generalization of information theory. In *IEEE Transactions on Communications*, pages 1345–1351, 1973.

[17] J. L. Massey. Causality, feedback and directed information. In *Proc. IEEE International Symposium on Information Theory and Its Applications*, pages 27–30, 1990.

[18] G. Neu and C. Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proc. UAI*, pages 295–302, 2007.

[19] A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proc. International Conference on Machine Learning*, pages 663–670, 2000.

[20] D. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems I*, 1:305, 1989.

[21] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *Proc. International Joint Conference on Artificial Intelligence*, pages 2586–2591, 2007.

[22] N. Ratliff, J. A. Bagnell, and M. Zinkevich. Maximum margin planning. In *Proc. International Conference on Machine Learning*, pages 729–736, 2006.

[23] N. Ratliff, D. Bradley, J. Bagnell, and J. Chestnutt. Boosting structured prediction for imitation learning. In *In Advances in Neural Information Processing Systems (NIPS) 19*, 2007.

[24] J. Rust. Maximum likelihood estimation of discrete control processes. *SIAM Journal on Control and Optimization*, 26:1006, 1988.

[25] U. Syed and R. Schapire. A game-theoretic approach to apprenticeship learning. *Advances in Neural Information Processing Systems*, 20:1449–1456, 2008.

[26] B. D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Machine Learning Department, Carnegie Mellon University, Dec 2010.

[27] B. D. Ziebart, J. A. Bagnell, and A. K. Dey. Modeling interaction via the principle of maximum causal entropy. In *Proc. International Conference on Machine Learning*, pages 1255–1262, 2010.

[28] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. Conference on Artificial Intelligence (AAAI)*, pages 1433–1438, 2008.

[29] B. D. Ziebart, A. Maas, A. K. Dey, and J. A. Bagnell. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proc. International Conference on Ubiquitous Computing*, 2008.

[30] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Proc. of the International Conference on Intelligent Robots and Systems*, 2009.

APPENDIX

Theorem 1: The hybrid IOC optimization of Definition 6 can be formulated as a softened version of the Bellman equation [3] via convex duality:

$$Q^{\text{hyb}}_{\theta,\phi}(a,s) = \left\{ \begin{array}{ll} \theta^\top \mathbf{f}_{s,a} + \mathbb{E}_{P(S'|s,a)}\left[V^{\text{hyb}}_{\theta,\phi}(s')\right] & \text{if } s \in \mathcal{S}_R \\ \phi_Q^\top \mathbf{f}_{s,a} & \text{if } s \in \mathcal{S}_Q \\ \sum_{s' \in \mathcal{S}} P(s'|s,a)\ \phi_V^\top \mathbf{f}_{s'} & \text{if } s \in \mathcal{S}_V \end{array} \right\} \tag{24}$$

$$V^{\text{hyb}}_{\theta,\phi}(s) = \operatorname*{softmax}_a Q^{\text{hyb}}_{\theta,\phi}(a,s),$$

with $\operatorname{softmax}_x f(x) = \log \sum_x e^{f(x)}$, the reward function parameterized by reward weights θ, and the value functions parameterized by value weights $\phi = \{\phi_Q, \phi_V\}$.

Proof: After setting features over the entire sequence as

$$\mathbf{F}(\mathbf{s},\mathbf{a}) = \begin{pmatrix} \sum_{t=1}^{T-1} \mathbf{f}_{s_t,a_t} \\ I_{\mathcal{S}_Q}(s_T)\ \mathbf{f}_{s_T} \\ I_{\mathcal{S}_V}(s_T)\ \sum_{s' \in \mathcal{S}} P(s'|s_T,a_T)\ \mathbf{f}_{s'} \end{pmatrix}, \tag{25}$$

this is a corollary of Theorem 6.8 [26]. ∎

Theorem 2: When state type beliefs are independent, $P(\text{type}(\mathcal{S})) = \prod_{s \in \mathcal{S}} P(\text{type}(s))$, the state-action value of Equation 16 can be re-written as:

$$\begin{aligned} Q_{\theta,\phi}(a,s) =& P(s \in \mathcal{S}_R)\ (\theta^\top \mathbf{f}_{s,a} + \mathbb{E}[V(s')|s,a]), \\ &+ P(s \in \mathcal{S}_Q)\ \phi_Q^\top \mathbf{f}_{s,a} \\ &+ P(s \in \mathcal{S}_V) \sum_{s' \in \mathcal{S}} P(s'|s,a)\ \phi_V^\top \mathbf{f}_{s'} \end{aligned} \tag{26}$$

with the interpretation that at each visit to a state, a sample from $P(\text{type}(s))$ is drawn indicating whether the state is treated as a reward-based influence, an action value-based influence, or a state value-based influence.

Proof: (sketch) The situation where state type is distributed according to an independent belief distribution can be represented with an augmented set of dynamics over the joint of states and state types:

$$P_{\text{expand}}(s', \text{type}(s')|s,a) = P(s'|s,a) P(\text{type}(s')), \tag{27}$$

where $P(s'|s,a)$ is the original dynamics and $P(\text{type}(s'))$ is the belief distribution over types. Following Theorem 1 in this expanded state space completes the proof. ∎

Theorem 3: The log probability of a policy under the hybrid reward-based/value-based inverse optimal control approach is related to the policy's expected rewards as follows:

$$\begin{aligned} &\log P(\tilde{\pi}|\text{type}(\mathcal{S}),\theta,\phi) = \\ &\mathbb{E}_{P(\mathbf{S},\mathbf{A})}\left[\sum_{t=1}^{T-1} \theta^\top \mathbf{f}_{s_t,a_t} + \phi^\top \mathbf{f}_{s_T,a_T} - V^{\text{hyb}}_{\theta,\phi}(s_1), \middle| \tilde{\pi}\right] \end{aligned} \tag{28}$$

where $V^{\text{hyb}}_{\theta,\phi}$ is defined according to Theorem 1.

Proof: This is a corollary of Theorem 6.10 [26] with features defined according to Equation 25. ∎