MULTI-OBJECTIVE CONSTRAINT INFERENCE USING INVERSE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Constraint inference from expert demonstrations is widely considered essential to align reinforcement learning agents with safety boundaries and operational guidelines. However, existing approaches typically assume homogeneous demonstrations, generated by a single expert or multiple experts with identical objectives. They also have limited ability in capturing individual preferences and often suffer from computational inefficiencies. In this paper, we introduce Multi-Objective Constraint Inference (MOCI), a novel framework to jointly extract shared constraints and individual preferences from heterogeneous expert trajectories, where multiple experts pursue different objectives while adhering to the same safety constraints. MOCI effectively models and learns from diverse, and potentially conflicting, behaviors. Empirical evaluations demonstrate that MOCI improved predictive performance, showed competitive computational efficiency, and recovered clinically credible preferences and constraints in a longitudinal lung cancer treatment setting. These results establish MOCI as an accurate, flexible, and practical approach to the real-world constraint inference and preference learning tasks in inverse reinforcement learning with heterogeneous experts.

1 INTRODUCTION

Inferring safety boundaries and operational guidelines from expert demonstrations is essential for aligning reinforcement learning agents with real-world requirements. In domains ranging from clinical decision-making to autonomous systems, expert behavior is driven by both hard physical or operational constraints and multi-objective trade-offs among competing goals. However, disentangling safety constraints from underlying reward preferences remains a fundamental challenge. Existing work (Scobee & Sastry, 2020; McPherson et al., 2021; Malik et al., 2021) suffers from two restrictive assumptions: (1) the dataset is homogeneous, i.e., generated by a single type of expert, and (2) the expert's reward function is completely known *a priori*.

These assumptions are problematic in real-world settings where observed data typically emerge from a heterogeneous population of experts, and the reward function is not known. For instance, in advanced cancer treatment, patients may have substantially different treatment preferences despite being subject to the same underlying clinical constraints, such as biomarker eligibility, organ-function thresholds, and toxicity limits. Some patients may prefer the most aggressive treatment options available (e.g., surgery) to maximize survival, even at the cost of substantial side effects, whereas others may prioritize preserving quality of life and therefore favor less intensive treatment (e.g., palliative care). Furthermore, assuming a pre-specified reward function forces learned constraints to absorb unmodeled preference variations, resulting in erroneously inferred safety constraints and poor policy generalizability.

To address these limitations, we present a framework for Multi-Objective Constraint Inference (MOCI). MOCI formalize the joint inference of shared hard state-action constraints $C \subseteq \mathcal{S} \times \mathcal{A}$ and individual preference vectors $\{\boldsymbol{w}_k\}_{k=1}^{K}$ from unlabeled expert trajectories. Additional algorithmic contributions of MOCI include:

1. **Counterfactual constraint evaluation** a combined global path evaluation and local feature-based policy decompositions to evaluate candidate constraints at unvisited state-action pairs in determining the likelihood of observing a trajectory given a candidate set of constraints during inverse RL.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2. **Noise Robustness & Optimization Modes:** Integrates softened violation penalties (η) to prevent likelihood collapse under demonstration noise
3. **Separation of constraints and Preferences:** a frozen baseline mode prevents reward functions from absorbing hard physical or clinical constraints, resulting in more simple reward models.
4. **Targeted Search & Rule Generalization:** Optimizes constraint inference by targeting state-action pairs with high model-expert disagreement ($\Delta\pi(s, a)$), generalizing learned rules across state signatures, and regularizing model complexity with λ.

MOCI demonstrates high constraint precision and accurate preference recovery on a multi-feature gridworlds and a synthetic longitudinal lung cancer treatment use-case, which highlights its significance and importance in furthering the field of learning suitable constraints and reward functions from expert demonstrations with heterogeneous preferences.

2 PRELIMINARIES

2.1 CONSTRAINED MARKOV DECISION PROCESSES

A standard Markov Decision Process (MDP) is defined as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, R \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}(s' \mid s, a) \in \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition probability distribution, $R(s, a) \in \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function (Singh et al., 2022). The behavior of an agent is defined by a policy $\pi(a \mid s) \in \mathcal{S} \to \Delta(\mathcal{A})$, which assigns states to a probability distribution over actions.
A *Constrained Markov Decision Process* (CMDP) (Wachi & Sui, 2020) extends this framework by restricting the set of allowable policies. Although CMDP can be formulated using cost functions and budget thresholds, we focus on the context of hard environmental constraints (such as walls or strict physical limitations), defined as a set of forbidden or unsafe states or actions denoted $C \subset \mathcal{S} \times \mathcal{A}$ following recent related work by (Qadri et al., 2025).
The CMDP is thus augmented to $\mathcal{M}_C = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, R, C \rangle$.
A trajectory $\xi = \{(s_0, a_0), (s_1, a_1), \ldots, (s_T, a_T\}$ of length $T \in \mathbb{N}$ is considered valid if and only if it does not violate any constraints in C (Schlaginhaufen & Kamgarpour, 2023). This is formalized using a binary indicator function:

$$\mathbb{I}^C(\xi) = \begin{cases} 1 & \text{if } (s_t, a_t) \notin C \text{ for all } t \in \{0, \ldots, T\} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

2.2 MULTI-OBJECTIVE MDP

A *Multi-Objective Markov Decision Process* (MOMDP) extends traditional MDPs with a vectorial reward function $\boldsymbol{R} \in \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ to model environments with d potentially competing objectives (e.g., minimizing travel time while maximizing safety) (Barrett & Narayanan, 2008; Hayes et al., 2022). This vectorial reward can be collapsed into a scalar reward if the particular way in which these multiple objectives are to be combined for some particular individual or use case, are known. In case this so-called *scalarization* function is linear in the objectives, we refer to its weights $w := [w_1, \ldots, w_p]$ as the preferences of that individual such that their reward function $R = w^\top \boldsymbol{R}$.

In practice, this is often expressed via a feature formulation where $\phi(s, a) \in \mathbb{R}^d$ represents a vector of state-action features, and an agent's specific preference is defined by a weight vector w. The scalarized reward (Shah et al., 2023) for an agent with preference w can be defined as:

$$R_w(s, a) = w^\top \phi(s, a) \tag{2}$$

In a heterogeneous multi-agent setting (Zhong et al., 2024), different experts $k \in \{1, \ldots, K\}$ share the same state-action features ϕ but possess distinct, private preference weights w_k, leading to diverse optimal policies within the same underlying environment.

2.3 INVERSE REINFORCEMENT LEARNING

Inverse Reinforcement Learning (IRL) addresses the problem of extracting an agent's underlying reward function given its demonstrated behavior (Arora & Doshi, 2021). Formally, we assume access

to the MDPs transition function but not its reward function, denoted as $\mathcal{M} \setminus R$, and a dataset of expert trajectories $\mathcal{D} = \{\xi_1, \ldots, \xi_N\}$ generated by an expert policy π_E.

The goal of IRL is to find a reward function R^* such that the expert's policy π_E is optimalAdams et al. (2022). If the reward is parameterized linearly as $R(s,a) = w^\top \phi(s,a)$, the IRL problem can be framed as finding a weight vector w^* such that the expected feature counts of the expert match the expected feature counts of a policy optimizing w^*. However, this problem is inherently ill-posed, as multiple reward functions (including a trivial reward of all zeros) can explain the same behavior.

To resolve the ambiguity of the IRL problem, (Ziebart et al., 2008) introduced *Maximum Entropy Inverse Reinforcement Learning* (MaxEnt IRL). MaxEnt IRL applies the principle of maximum entropy to select the probability distribution over trajectories that matches the expert's empirical feature expectations while making no other assumptions (i.e., being as random as possible otherwise). Under the MaxEnt framework, the probability of an agent choosing a specific trajectory ξ is exponentially proportional to the total accumulated reward of that trajectory:

$$P(\xi \mid w) = \frac{1}{Z(w)} \exp\left(\sum_{(s,a)\in\xi} w^\top \phi(s,a)\right) = \frac{1}{Z(w)} e^{R_w(\xi)} \tag{3}$$

where $Z(w)$ is the partition function, representing the integral (or sum) over all possible trajectories.

$$Z(w) = \sum_{\xi' \in \Xi} e^{R_w(\xi')} \tag{4}$$

The reward weights w are then found by maximizing the log-likelihood of the demonstrated trajectories $\mathcal{D}$:

$$\mathcal{L}(w) = \sum_{i=1}^{|\mathcal{D}|} \log P(\xi_i \mid w) = \sum_{i=1}^{|\mathcal{D}|} \left(w^\top \phi(\xi_i) - \log Z(w)\right) \tag{5}$$

The gradient of this log-likelihood neatly reduces to the difference between the empirical feature counts of the demonstrations and the expected feature counts under the current weight vector w, allowing for efficient optimization via gradient ascent.

3 Multi-Objective Constraint Inference (MOCI)

Let $\mathcal{D} = \{\xi_1, \ldots, \xi_N\}$ be a dataset of N demonstrated trajectories of maximum horizon H, each defined as a sequence of state-action pairs $\xi_i = \{(s_0, a_0), (s_1, a_1), \ldots, (s_{T-1}, a_{T-1}), (s_T, -1)\}$, where $a_t \in \mathcal{A}$ denotes the action executed at state $s_t \in \mathcal{S}$ at some $t \leq H$ (Shah et al., 2022). We model the environment as a Constrained Multi-Objective Markov Decision Process (CMOMDP) $\mathcal{M}_C = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, \boldsymbol{\phi}, R, C \rangle$, where $\boldsymbol{\phi}(s,a) \in \mathbb{R}^d$ is a d-dimensional state-action feature mapping, and $C \subseteq \mathcal{S} \times \mathcal{A}$ represents a shared set of hard state-action constraints (or action infeasibility masks). We assume K latent expert clusters. Each cluster $k \in \{1, \ldots, K\}$ is defined by an unobserved, unknown preference weight vector $\boldsymbol{w}_k \in \mathbb{R}^d$, giving a scalarized linear reward function $R_k(s,a) = \boldsymbol{w}_k^\top \boldsymbol{\phi}(s,a)$, and a cluster prior probability $\pi_k = P(k)$ with $\sum_{k=1}^K \pi_k = 1$. Trajectory demonstrations are unlabeled with respect to cluster assignments k.

MOCI provides two distinct formulations for trajectory likelihood computation:
(a) Global Path Likelihood (`path`): Following the MaxEnt IRL framework (see Section 2.3), the probability of observing trajectory ξ generated by cluster k subject to constraints C is :

$$P_{path}(\xi \mid C, \boldsymbol{w}_k) = \frac{1}{Z(C, \boldsymbol{w}_k)} \exp\left(\sum_{t=0}^{T-1} \boldsymbol{w}_k^\top \boldsymbol{\phi}(s_t, a_t)\right) \prod_{t=0}^{T-1} \mathbb{I}_\eta^C(s_t, a_t) \tag{6}$$

A detailed analysis and proofs are given in appendix-B. In equation-6, $Z(C, \boldsymbol{w}_k) = Z[s_0, 0]$ is the partition function evaluated at the initial state s_0 at time $t = 0$ via backward pass recursion, and

$$\log Z[s,t] = \log \left[\sum_{a \in \mathcal{A}_{\text{allowed}}(s,C)} \exp\left(\boldsymbol{w}_k^\top \boldsymbol{\phi}(s,a) + \log Z[\mathcal{T}(s,a), t+1]\right)\right]. \tag{7}$$

(b) Local Policy Likelihood (`local`): When evaluating candidate constraints at states that are not reachable from s_0 under current dynamics, the global path partition function $Z[s_0, 0]$ remain invariant to changes in candidate constraints, yielding zero likelihood gain. To counterfactually evaluate candidate constraints at unvisited or off-path states, MOCI decomposes the likelihood into localized step-wise MaxEnt policy choices :

$$P_{local}(\xi \mid C, \boldsymbol{w}_k) = \prod_{t=0}^{T-1} \pi_k(a_t \mid s_t, t) \tag{8}$$

$$\pi_k(a \mid s, t) = \frac{\exp\left(\boldsymbol{w}_k^\top \boldsymbol{\phi}(s,a) + \log Z[\mathcal{T}(s,a), t+1]\right)}{\sum_{a' \in \mathcal{A}_{allowed}(s,C)} \exp\left(\boldsymbol{w}_k^\top \boldsymbol{\phi}(s,a') + \log Z[\mathcal{T}(s,a'), t+1]\right)} \tag{9}$$

To prevent total likelihood collapse ($P(\xi) = 0$) when expert demonstrations contain either observation noise, sub-optimal actions, or rare constraint violations, the zero indicator functions is softened via a violation hyperparameter $\eta \in (0, 1)$:

$$\mathbb{I}_\eta^C(s_t, a_t) = \begin{cases} 1 & \text{if } (s_t, a_t) \notin C \text{ and } (s_t, a_t) \text{ is feasible} \\ \eta & \text{otherwise} \end{cases} \tag{10}$$

A violating step incurs a finite log-penalty $\log \eta$ in order to prevent that the entire trajectory probability is fixed to zero.

Marginalizing over the latent cluster assignments $k \in \{1, \ldots, K\}$, the normalized average log-likelihood across dataset $\mathcal{D}$ is:

$$\mathcal{L}_{avg}(C, \{\boldsymbol{w}_k\}, \{\pi_k\}) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \log \left(\sum_{k=1}^{K} \pi_k P(\xi_i \mid C, \boldsymbol{w}_k) \right), \tag{11}$$

resulting in a total log-likelihood of $\mathcal{L}_{total} = |\mathcal{D}| \cdot \mathcal{L}_{avg}$.

MOCI optimizes the joint log-likelihood via Expectation-Maximization (EM) (Toussaint et al., 2011). Given current parameters $(C, \{\boldsymbol{w}_k\}, \{\pi_k\})$, we compute responsibility $\gamma_{i,k}$, representing the posterior probability that trajectory ξ_i belongs to cluster k:

$$\gamma_{i,k} = \frac{\pi_k P(\xi_i \mid C, \boldsymbol{w}_k)}{\sum_{j=1}^{K} \pi_j P(\xi_i \mid C, \boldsymbol{w}_j)} \tag{12}$$

Cluster priors are updated as empirical responsibility means:

$$\pi_k = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \gamma_{i,k} \tag{13}$$

And reward weights for each cluster are updated via gradient ascent with elementwise weight clipping $|w_{j,k}| \leq w_{max}$:

$$\nabla_{\boldsymbol{w}_k} \mathcal{L}_{total} = \sum_{i=1}^{|\mathcal{D}|} \gamma_{i,k} \left(\boldsymbol{\phi}(\xi_i) - \mathbb{E}_{P(\xi \mid C, \boldsymbol{w}_k)}[\boldsymbol{\phi}(\xi)] \right) \tag{14}$$

$$\boldsymbol{w}_k \leftarrow \text{Clip}\left(\boldsymbol{w}_k + \alpha \frac{\nabla_{\boldsymbol{w}_k} \mathcal{L}_{total}}{|\mathcal{D}|}, -w_{max}, w_{max} \right) \tag{15}$$

where expected feature counts $\mathbb{E}_{P(\xi \mid C, \boldsymbol{w}_k)}[\boldsymbol{\phi}(\xi)]$ are calculated using Monte Carlo trajectory sampling (Browne et al., 2012) over N_{sim} paths per cluster step to ensure gradient stability. Optimization is performed in two modes: `joint`, which jointly re-fits the reward weights, priors, and constraints at each optimization cycle, and `frozen`, which fixes the unconstrained baseline rewards and priors during constraint search, preventing the reward function from absorbing hard restrictions.

CONSTRAINT CANDIDATE SCREENING & SAFETY PRE-SCREENING

Candidates constraints are evaluated as state-action pairs $(s, a) \in \mathcal{S} \times \mathcal{A}$. To allow candidate generation in noisy domains, a state-action pair (s, a) is admitted to the candidate pool $\mathcal{C}_{pool}$ if $s \in \mathcal{S}_{visited}$ and its empirical visitation count satisfies $n(s, a) \leq n_{max}$ or if its frequency satisfies $\frac{n(s,a)}{n(s)} < \epsilon$, subject to a domain safety pre-screen filter $\mathcal{S}_{safe}(s, a)$:

$$\mathcal{C}_{pool} = \left\{ (s, a) \in \mathcal{S}_{visited} \times \mathcal{A} \;\middle|\; \left(n(s, a) \leq n_{max} \vee \frac{n(s, a)}{n(s)} < \epsilon \right) \wedge \mathcal{S}_{safe}(s, a) \right\} \tag{16}$$

Where, $\mathcal{S}_{\text{safe}}(s, a)$ is a boolean function (*True* or *False*) that allows the algorithm to incorporate known physical, clinical, or domain-level safety guidelines directly into candidate selection. Candidates are prioritized by state-level model-expert policy discrepancy:

$$\Delta\pi(s, a) = \pi_{model}(a \mid s) - \pi_{expert}(a \mid s) \tag{17}$$

$$\pi_{model}(a \mid s) = \sum_{k=1}^{K} \pi_k \left(\frac{1}{H} \sum_{t=0}^{H-1} \pi_k(a \mid s, t) \right), \quad \pi_{expert}(a \mid s) = \frac{n(s, a)}{\max(n(s), 1)} \tag{18}$$

Candidates with high positive discrepancy, i.e. where the current policy assigns probability to actions unobserved in expert data, are selected first for constraint evaluation. Additionally, candidates that share a clinical or feature signature $\sigma(s, a)$ (e.g., matching biomarker eligibility rules or organ reserve limits) are grouped:

- *Pointwise Expansion (`expand`):* When a candidate pair (s, a) is accepted into C, MOCI automatically expands C to all candidate pairs $(s', a) \in \mathcal{C}_{pool}$ sharing signature $\sigma(s', a) = \sigma(s, a)$.
- *Rule-Based Units (`rule`):* Candidate constraints are grouped into units using their signatures $U = \{(s', a) \mid \sigma(s', a) = \sigma_{target}\}$.

For each candidate unit u (individual pair or signature group), its score evaluates the total log-likelihood gain against a complexity penalty λ:

$$\text{Score}(u) = \Delta\mathcal{L}_{total}(u) - \lambda = |\mathcal{D}| \cdot (\mathcal{L}_{avg}(C \cup u) - \mathcal{L}_{avg}(C)) - \lambda \tag{19}$$

All candidates in unit $u^* = \arg\max_u \text{Score}(u)$ are accepted into C if $\text{Score}(u^*) > 0$.

3.1 COMPUTATIONAL COMPLEXITY ANALYSIS

We now analyse the computational complexity of MOCI. First, let $|\mathcal{S}|$ be the state space size, $|\mathcal{A}|$ the action space size, H the trajectory horizon, $N = |\mathcal{D}|$ number of demonstrations, K the number of latent clusters, I_{IRL} the number of gradient steps, N_{sim} the number of Monte Carlo rollout samples, M_{cand} the number candidate units, and C_{added} the inferred constraints.

Log partition matrices $Z(C, \boldsymbol{w}_k)$ require backward passes over horizon H, with complexity $\mathcal{O}(K \cdot H \cdot |\mathcal{S}| \cdot |\mathcal{A}|)$. Trajectory probability evaluation across N demonstrations has complexity $\mathcal{O}(K \cdot N \cdot H)$.

$$\mathcal{O}_E = \mathcal{O}\left(K \cdot H \cdot (|\mathcal{S}| \cdot |\mathcal{A}| + N)\right) \tag{20}$$

For each cluster, MOCI performs I_{IRL} gradient steps. Feature expectation computation via N_{sim} Monte Carlo trajectory samples requires $\mathcal{O}(K \cdot I_{IRL} \cdot N_{sim} \cdot H)$ operations. Empirical gradient accumulation has complexity $\mathcal{O}(K \cdot I_{IRL} \cdot N \cdot H)$. The maximization step therefore has complexity:

$$\mathcal{O}_M = \mathcal{O}\left(K \cdot I_{IRL} \cdot H \cdot (N_{sim} + N)\right) \tag{21}$$

Evaluating M_{cand} candidates across C_{added} greedy iterations requires $C_{added} \cdot M_{cand}$ backward pass evaluations.

$$\mathcal{O}_C = \mathcal{O}\left(C_{added} \cdot M_{cand} \cdot K \cdot |\mathcal{S}| \cdot |\mathcal{A}| \cdot H\right) \tag{22}$$

Summing across I_{EM} iterations, treating $K, I_{IRL}, N_{sim}, I_{EM}, C_{added}$ as domain constants, and noting that state-action transition evaluations $|\mathcal{S}| \cdot |\mathcal{A}| \cdot H$ dominate sampled demonstrations N, the asymptotic computational complexity reduces to:

$$\text{Total Cost} = \mathcal{O}\left(|\mathcal{S}| \cdot |\mathcal{A}| \cdot H \cdot M_{cand}\right) \tag{23}$$

In tabular grid environments ($|\mathcal{S}| = N_g^2$), MOCI scales quadratically with grid dimension N_g. In the longitudinal clinical state spaces ($|\mathcal{S}| = N_{patients} \times T_{visits}$), MOCI scales linearly with total patient-visit timepoints.

4 USE CASES:

In this section, we present two use cases. Gridworld serves as a controlled testbed for evaluating MOCI under well-defined experimental conditions, while the lung cancer use case demonstrates its applicability in a clinical setting. All results reported in this article were obtained by running experiments on a *MacBook M3* using *Python 3.11.5*. The source code and implementation details are available in an anonymized code repository: `https://anonymous.4open.science/r/MuitlObjective_constraint_Learning-3839/README.md`.

4.1 HETEROGENEOUS GRIDWORLD

To validate the MOCI framework, we designed a simulated Markov Decision Process (MDP) on a multi-feature $N \times N$ Gridworld environment as shown in figure-1. The state space $\mathcal{S}$ consists of grid cells, and the action space $\mathcal{A}$ allows for standard movement: $\{$*Up, Down, Left, Right, Stay*$\}$. The environment features a "Start" state in the top-left corner and a "Goal" state in the bottom-right corner. For this domain, state-action features simplify to state-level terrain features such that $\boldsymbol{\phi}(s, a) = \boldsymbol{\phi}(s)$, that represents four different types of terrain: grasslands (green), Rocks(brown), Water (blue) and Normal (light beige) tiles. Ground-truth hard constraints $C^* \subseteq \mathcal{S} \times \mathcal{A}$ govern safety by prohibiting transitions involving impassable water tiles.
To simulate a heterogeneous population, we defined two distinct expert types with divergent reward preferences, parameterized by their ground-truth weight vectors $\boldsymbol{w}_k^* \in \mathbb{R}^4$:

1. **Expert 1 (Grass profile):** Highly prefers traveling on the Grass tiles
2. **Expert 2 (Rock profile):** Highly prefers traveling on the Rock tiles

The agent's objectives are to reach the goal state by taking the shortest path while navigating their preferred terrain. Both experts share the same terminal goal state and are fundamentally restricted by the Water tiles (C^*). In Figure 1, the trajectories of Expert 1 (Grass profile) are shown in lime, while the trajectories of Expert 2 (Rock profile) are shown in orange.

Results for Heterogeneous Gridworld: During testing, the constraint and preferences information is not provided to the MDP-environment. Given only the 10 expert trajectories, MOCI simultaneously learns the shared constraints and the individual preferences of each expert. Figure 2 visualizes the constraints inferred by MOCI after observing the expert trajectories shown in Figure 1. The algorithm successfully places "Inferred Constraints" (marked by red hatched boxes) over the blue water tiles. The red hatched boxes also appear on non-water tiles (e.g., a normal tile on the top right). The accuracy of the inferred constraints can be further improved by the appropriate number of expert trajectories.

Figure 3 illustrates the ability of the MOCI algorithm to disentangle and recover the distinct underlying reward functions of heterogeneous experts operating in the same environment. For Expert 1 (Grass profile), MOCI learns a strong positive weight for Grass ($+0.69$), a near-zero weight for Rocks ($+0.01$), and a strong negative weight for Water (-0.72), capturing the expert's preference for grass

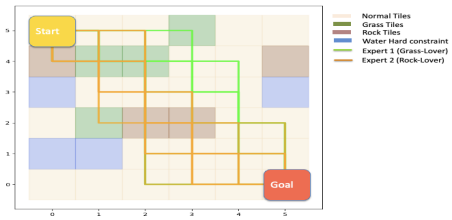

Figure 1: Demonstration of the MOCI algorithm in a 6×6 Grid-world.

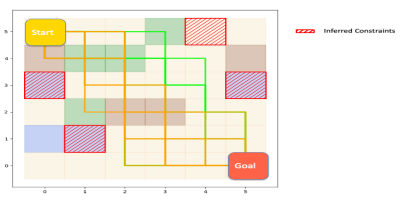

Figure 2: MOCI-inferred constraints (red hatched), showing recovery of true constraints.

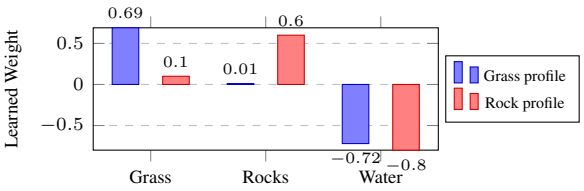

Figure 3: Joint recovery of heterogeneous preferences with MOCI, comparing learned weights for two distinct expert profiles.

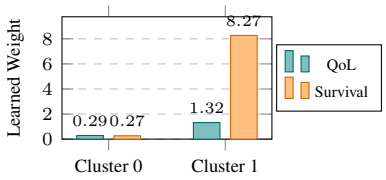

Figure 4: Learned preference weights across the two inferred patient clusters, showing distinct latent trade-offs.

and aversion to water. Expert 2 (Rock profile) exhibits a distinct profile, with positive weights for Grass ($+0.10$) and Rocks ($+0.60$), and a strong negative weight for Water (-0.80), reflecting a clear preference for rocks and weaker preference for grass. Overall, the learned reward vectors successfully distinguish the two experts' terrain preferences, while the consistently negative Water weights are expected because water tiles are hard constraints and are never visited by either expert.

4.2 Lung Cancer

Advanced lung cancer treatment is a sequential decision-making problem in which clinicians repeatedly balance competing objectives, particularly preserving quality of life and prolonging survival. In clinical practice, treatment decisions are further constrained by factors such as biomarker status, toxicity burden, organ function, and performance status, making some treatment actions infeasible for a given patient at a particular time point. Inferring both shared clinical constraints and heterogeneous treatment preferences from historical medical data is therefore important for understanding clinical decision-making and for developing reinforcement learning agents that respect safety boundaries and operational guidelines. Dataset and treatment-eligibility rules are described in Appendix D.

The motivation for this use case is threefold: 1. Recover latent clinical preference patterns from observed trajectories rather than assuming a single universal objective. 2. Model realistic treatment feasibility so that the learned policy explanations remain clinically credible. 3. Support longitudinal decision support, where recommendations are conditioned on the evolving state of the patient. This framework is valuable for research because it moves from static retrospective association to trajectory-level preference inference under medical constraints.

Results for lung cancer: Figure 4 illustrates the learned preference profiles across the two groups. Cluster 0 exhibits a relatively balanced preference between QoL and survival, indicating comparable importance assigned to both outcomes. In contrast, Cluster 1 demonstrates a substantially stronger preference for survival over QoL, reflecting a more survival-oriented treatment preference.
Next, we evaluated how effectively the learned constraint set recovers the underlying treatment-eligibility restrictions and results are reported in figure 5. Across 326 patient states and four treatments, 1,304 pairs were evaluated: 850 infeasible and 454 feasible. Of the 850 infeasible pairs, 619 were correctly identified (TPR: 72.8%, 95% CI: 69.7–75.7), while 231 were missed (FNR: 27.2%). Among the 454 feasible pairs, 451 were correctly left unconstrained (TNR: 99.3%, 95% CI: 98.1–99.8), with only three false positives (FPR: 0.7%, 95% CI: 0.2–1.9). Overall, precision was 99.5% (619/622), indicating that most learned constraints were clinically justified. Additionally, the evaluation involved

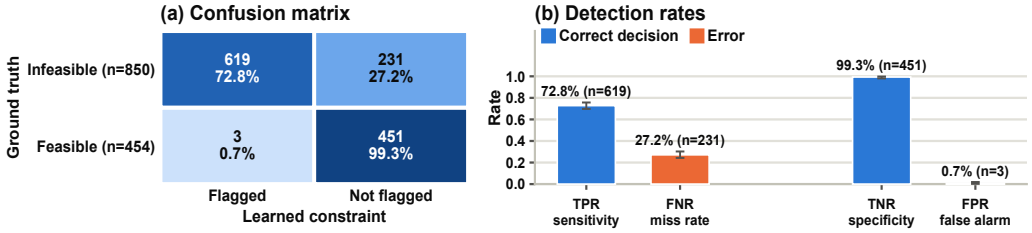

Figure 5: Detection of treatment constraints from clinical trajectories. (a) Confusion matrix of the learned constraints against the clinical eligibility masks. (b) Detection rates with 95% Wilson confidence intervals. Blue bars are correct decisions and orange bars are errors.

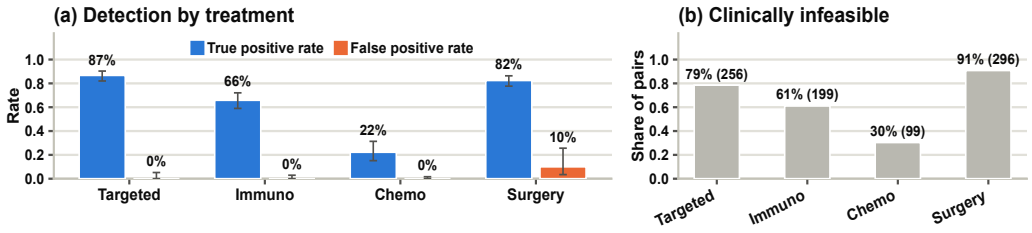

Figure 6: Constraint detection by treatment. (a) True-positive rate (blue) and false-positive rate (orange) for each treatment, with 95% Wilson confidence intervals. (b) Share of state–action pairs that are clinically infeasible for each treatment (ground truth), with the number of infeasible pairs in brackets. Each treatment is evaluated on the same 326 patient states.

100 patients, so the confidence intervals reflect variability across state–action pairs rather than across patients.

Detection performance varies substantially by treatment as shown in the figure 6. Recall is highest for Targeted therapy (86.7%, 95% CI: 82–90) and Surgery (82.4%, 78–86), followed by Immunotherapy (65.8%, 59–72) and Chemotherapy (22.2%, 15–31). This pattern broadly follows the proportion of infeasible state–action pairs: 91% for Surgery, 79% for Targeted therapy, 61% for Immunotherapy, and 30% for Chemotherapy. Recovery also varies by the underlying clinical rule. The Targeted therapy rule is recovered in 99% of pairs (222/224), compared with 94% for Surgery and 89% for Immunotherapy. In contrast, only 22% of Chemotherapy constraints are recovered (22/99), consistent with its low recall, although the analysis does not establish causality. False positives are rare, with none for Targeted therapy, Immunotherapy, or Chemotherapy and three for Surgery. The Surgery FPR is 10.0% (3/30; 95% CI: 3.5–25.6), but the small number of feasible pairs yields a wide interval.

5 Comparison with existing techniques

To rigorously evaluate the efficacy of the proposed Multi-Objective Constraint Inference (MOCI) framework, it is imperative to benchmark its performance against established state-of-the-art techniques. For a fair comparison, we evaluated these techniques on the same GridWorld (8×8) environment, which serves as the common testbed for all methods. Table 1 presents the performance comparison with two existing baselines, Maximum Likelihood Constraint Inference (MLCI) (Scobee & Sastry, 2020) and Inverse Constrained Reinforcement Learning (ICRL) (Malik et al., 2021) across four evaluation metrics.

Table 1 reports the mean $\pm$ standard deviation over 15 random seeds, 20 each expert trajecroies, for each of the three constraint-inference methods on the shared 8×8 GridWorld MDP. MOCI achieves the lowest constraint mean-squared error (MSE = 0.0156) with zero variance across all seeds, alongside a precision of 1.0000, the highest F1 score (0.8889), and the highest Matthews Correlation Coefficient (MCC = 0.8869). Although its recall (0.8000) is lower than that of the baseline methods, this stems from a single consistently missed constraint cell rather than seed-to-seed instability; the inferred constraint set remains identical across runs.

While MLCI and ICRL achieve higher recall (0.9467 and 0.9600, respectively), their lower precision degrades overall classification performance. MLCI obtains a precision of 0.5527 (F1 = 0.6927),

Table 1: Performance comparison across existing Approaches

Metric	MOCI (our)	MLCI (Scobee & Sastry, 2020)	ICRL (Malik et al., 2021)
Run-time (s) (↓)	3.5236 ± 0.1741	$\mathbf{0.3451 \pm 0.1376}$	17.0955 ± 0.4309
MSE (↓)	$\mathbf{0.0156 \pm 0.0000}$	0.0677 ± 0.0247	0.3563 ± 0.0685
Precision (↑)	$\mathbf{1.0000 \pm 0.0000}$	0.5527 ± 0.0949	0.1790 ± 0.0315
Recall (↑)	0.8000 ± 0.0000	0.9467 ± 0.0884	$\mathbf{0.9600 \pm 0.1083}$
F1 Score (↑)	$\mathbf{0.8889 \pm 0.0000}$	0.6927 ± 0.0849	0.3006 ± 0.0474
MCC (↑)	$\mathbf{0.8869 \pm 0.0000}$	0.6915 ± 0.0869	0.3166 ± 0.0698

whereas ICRL yields a precision of 0.1790, resulting in the lowest F1 (0.3006) and MCC (0.3166) despite its high recall. Consequently, reporting F1 and MCC alongside precision and recall provides a more balanced evaluation of constraint recovery.

Additionally, MOCI exhibits zero variance across all classification metrics, whereas MLCI and ICRL display notable seed-to-seed variation, with MSE standard deviations of 0.0247 and 0.0685, respectively. Run-time differences directly align with the algorithmic design of each approach. MLCI records the shortest mean run-time $(0.3451$ s), followed by MOCI $(3.5236$ s) and ICRL $(17.0955$ s). MLCI relies on a bounded search over unvisited states without gradient-based optimization, whereas MOCI executes Expectation-Maximization (EM) iterations combining constraint search with reward-weight updates. Under this evaluation setup, ICRL incurs the highest computational cost, requiring considerably more execution time than the other frameworks for the current testbed.

MOCI explicitly models demonstrations from multiple latent experts, and it uniquely enables recovery of both expert reward vectors and cluster assignments, a capability absent in single-reward baselines (MLCI and ICRL). Recovery performance, shown in figure 7 is evaluated using two metrics: Cosine Similarity measures directional alignment between MOCI's learned reward vectors and the ground-truth vectors for Expert 1 (Grass) and Expert 2 (Rock) after optimal cluster matching. Medians reach ≈ 0.81 across both experts. Adjusted Rand Index (ARI) evaluates agreement between MOCI's inferred cluster responsibilities and true expert identities, accounting for chance. The median ARI reaches ≈ 0.51. Across all 15 random seeds, both metrics remain consistently well above the zero baseline (0.0), confirming that MOCI reliably recovers latent expert preferences and identities without absorbing constraints into the reward structure.

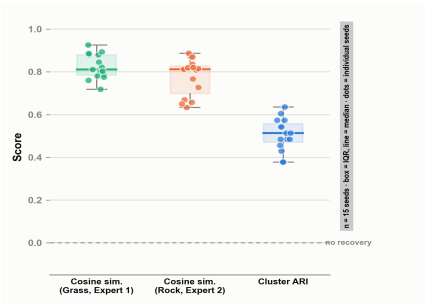

Figure 7: Reward and cluster recovery under MOCI ($n = 15$ random seeds; boxes show median and IQR; points show individual seeds)

6 Conclusion

In this paper, we introduced Multi-Objective Constraint Inference (MOCI), a novel framework to jointly infer shared hard constraints over the state-action space, and to recover latent heterogeneous expert preference weights from unlabeled demonstration trajectories. By integrating dual global path and local step-wise policy likelihood formulations, softened constraint violation penalties, and model-expert policy discrepancy ranking ($\Delta\pi(s, a)$), MOCI resolves existing issues with efficiently idenfying constraints and multi-objective reward weights intraditional constraint inference baselines. Empirical evaluations on multi-feature gridworld benchmarks demonstrate that MOCI is highly effective at constraint inference, while it is competitive with recovery of latent preferences and expert preference profiles. Furthermore, when applied to a longitudinal non-small cell lung cancer treatment setting, MOCI accurately identfies distinct patient clinical trade-offs (quality of life vs. survival), and treatment eligibility rules with high precision across state-action pairs.

Despite its theoretical and empirical strengths, MOCI has several limitations. Candidate screening and rule-based expansion depend on pre-defined feature signatures $\sigma(s, a)$ and empirical visitation thresholds $(n_{\max}, \epsilon)$, which may require domain expertise to configure in complex, unstructured environments. Moreover, MOCI parameterizes multi-objective rewards linearly $(R_k(s, a) = w_k^\top \phi(s, a))$, which may not fully capture non-linear trade-offs, thresholded preferences, or dynamic preference shifts over time.

Future work will expand MOCI in several key directions: (1) Extending MOCI to continuous state-action spaces using deep inverse reinforcement learning and neural function approximations for dynamic value and partition estimation. (2) Incorporating non-linear scalarization functions (e.g., Chebyshev scalarization) and modeling time-varying preference evolution across multi-horizon trajectories. (3) Benchmarking MOCI on retrospective observational Electronic Health Record (EHR) cohorts to support safe, interpretable prospective clinical decision support systems.

AI USE STATEMENT

In this work, we used generative AI tools for text refinement, grammar check, code debugging. We have not used generative AI tools for the development of the core concepts or for the generation of the reported results. Additionally, we used generative AI tools for rule extraction related to the lung cancer use case, reported in Appendix D.1, which we reviewed according to our understanding and expertise. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The lung cancer use case in this work relies entirely on synthetic data. No human subjects were involved, and no patient data was accessed. All presented results are intended for technical evaluation only and are not for clinical use

REPRODUCIBILITY STATEMENT

All code, model architectures, and random seed configurations required to reproduce the results in this paper are publicly accessible at `https://anonymous.4open.science/r/MuitlObjective_constraint_Learning-3839/README.md`.

REFERENCES

Stephen Adams, Tyler Cody, and Peter A Beling. A survey of inverse reinforcement learning. *Artificial Intelligence Review*, 55(6):4307–4346, 2022.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

Leon Barrett and Srini Narayanan. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th international conference on Machine learning*, pp. 41–47, 2008.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

Glen Chou, Dmitry Berenson, and Necmiye Ozay. Learning constraints from demonstrations. In *International Workshop on the Algorithmic Foundations of Robotics*, pp. 228–245. Springer, 2018.

Glen Chou, Necmiye Ozay, and Dmitry Berenson. Learning parametric constraints in high dimensions from demonstrations. In *Conference on robot learning*, pp. 1211–1230. PMLR, 2020.

Gene L Colice, Shirin Shafazand, John P Griffin, Robert Keenan, and Chris T Bolliger. Physiologic evaluation of the patient with lung cancer being considered for resectional surgery: Accp evidenced-based clinical practice guidelines. *Chest*, 132(3):161S–177S, 2007.

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

Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning: Cf hayes et al. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

Shehryar Malik, Usman Anwar, Alireza Aghasi, and Ali Ahmed. Inverse constrained reinforcement learning. In *International conference on machine learning*, pp. 7390–7399. PMLR, 2021.

David L McPherson, Kaylene C Stocking, and S Shankar Sastry. Maximum likelihood constraint inference from stochastic demonstrations. In *2021 IEEE conference on control technology and applications (CCTA)*, pp. 1208–1213. IEEE, 2021.

Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Dimitris Papadimitriou, Usman Anwar, and Daniel S Brown. Bayesian methods for constraint inference in reinforcement learning. *Transactions on Machine Learning Research*, 2022.

Mohamad Qadri, Gokul Swamy, Jonathan Francis, Michael Kaess, and Andrea Bajcsy. Your learned constraint is secretly a backward reachable tube. *Reinforcement Learning Journal*, 6:478–492, 2025.

Deepak Ramachandran, Eyal Amir, et al. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pp. 2586–2591, 2007.

Andreas Schlaginhaufen and Maryam Kamgarpour. Identifiability and generalizability in constrained inverse reinforcement learning. In *International conference on machine learning*, pp. 30224–30251. PMLR, 2023.

Dexter R.R. Scobee and S. Shankar Sastry. Maximum likelihood constraint inference for inverse reinforcement learning. In *International Conference on Learning Representations*, 2020.

Syed Ihtesham Hussain Shah, Antonio Coronato, Muddasar Naeem, and Giuseppe De Pietro. Learning and assessing optimal dynamic treatment regimes through cooperative imitation learning. *IEEE Access*, 10:78148–78158, 2022.

Syed Ihtesham Hussain Shah, Giuseppe De Pietro, Giovanni Paragliola, and Antonio Coronato. Projection based inverse reinforcement learning for the analysis of dynamic treatment regimes. *Applied Intelligence*, 53(11):14072–14084, 2023.

Rahul Singh, Abhishek Gupta, and Ness B Shroff. Learning in constrained markov decision processes. *IEEE Transactions on Control of Network Systems*, 10(1):441–453, 2022.

Marc Toussaint, Amos Storkey, and Stefan Harmeling. Expectation-maximization methods for solving (po) mdps and optimal control problems. *Bayesian Time Series Models*, pp. 388–413, 2011.

Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained markov decision processes. In *International Conference on Machine Learning*, pp. 9797–9806. PMLR, 2020.

Hanwen Wang, Theinmozhi Arulraj, Holly Kimko, and Aleksander S Popel. Generating immunogenomic data-guided virtual patients using a qsp model to predict response of advanced nsclc to pd-l1 inhibition. *NPJ Precision Oncology*, 7(1):55, 2023.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

APPENDIX

A NOTATIONS AND DESCRIPTION

Table 2 summarizes the mathematical notation, variables, parameters, and descriptions used throughout the Multi-Objective Constraint Inference (MOCI) framework.

Table 2: Notations and descriptions for MOCI.

Notation	Description	Notation	Description
Environment & CMOMDP Framework			
$\mathcal{M}_C$	CMOMDP tuple	$\mathcal{S}, \mathcal{A}$	State space and action space
$\mathcal{T}(s' \mid s, a)$	Transition probability distribution		
$\boldsymbol{\phi}(s, a)$	d-dim. feature map ($\in \mathbb{R}^d$)	d	Feature dimension (# objectives)
C	Shared hard state-action constraints	$\mathcal{A}_{\text{allowed}}(s, C)$	Allowed action set at state s given C
Demonstrations & Trajectories			
$\mathcal{D}, N$	Dataset $\mathcal{D} = \{\xi_i\}_{i=1}^N$, $N = \lvert\mathcal{D}\rvert$	ξ_i, ξ	Demonstrated trajectory sequence
H, T	Max horizon H; trajectory length T	$\mathbb{I}_\eta^C(s_t, a_t)$	Softened constraint indicator
η	Soft constraint parameter ($\eta \in (0, 1)$)		
Latent Expert Clusters & Preference Modeling			
K, k	# clusters K; cluster index k	$\boldsymbol{w}_k$	Preference weights for cluster k
$R_k(s, a)$	Linear reward $\boldsymbol{w}_k^\top \boldsymbol{\phi}(s, a)$		
Trajectory Likelihoods & Policy Distributions			
$P_{\text{path}}(\xi \mid C, \boldsymbol{w}_k)$	Global path likelihood	$P_{\text{local}}(\xi \mid C, \boldsymbol{w}_k)$	Local step-wise likelihood
$Z(C, \boldsymbol{w}_k), Z[s, t]$	Partition function via backward pass	$\pi_k(a \mid s, t)$	Local MaxEnt policy for cluster k
$\mathcal{L}_{\text{avg}}, \mathcal{L}_{\text{total}}$	Avg. and total dataset log-likelihood		
Expectation-Maximization (EM) & Optimization			
$\gamma_{i,k}$	Posterior responsibility $P(k \mid \xi_i)$	α	Gradient ascent learning rate
$\mathbb{E}_P[\boldsymbol{\phi}(\xi)]$	Expected rollout feature counts		
Candidate Screening, Discrepancy & Generalization			
$\mathcal{S}_{\text{visited}}$	Visited state set in demonstrations	$n(s, a), n(s)$	State-action & state visitation counts
$n_{\max}, \epsilon$	Screening count & frequency thresholds	$\mathcal{S}_{\text{safe}}(s, a)$	Domain safety pre-screen filter
$\mathcal{C}_{\text{pool}}, \mathcal{C}_{\text{cand}}$	Active candidate pool / units	$\Delta\pi(s, a)$	Policy discrepancy ($\pi_{\text{model}} - \pi_{\text{expert}}$)
$\pi_{\text{model}}, \pi_{\text{expert}}$	Marginalized model vs. expert policy	$\sigma(s, a)$	Feature signature mapping
u, u^*	Candidate unit / optimal unit	$\text{Score}(u)$	Candidate score vs. penalty λ
λ, d_{DKL}	Complexity penalty / divergence threshold		
Computational Complexity Analysis			
$\lvert\mathcal{S}\rvert, \lvert\mathcal{A}\rvert$	State and action space cardinalities	M_{cand}	Total candidate units evaluated
C_{added}	Inferred constraints accepted into C	N_g	Grid dim. ($\lvert\mathcal{S}\rvert = N_g^2$)

B PATH LIKELIHOOD UNDER SOFT CONSTRAINTS

By the Principle of Maximum Entropy, the probability distribution $P(\xi)$ over the space of candidate trajectories Ξ maximizes Shannon entropy:

$$\max_P \mathcal{H}(P) = -\sum_{\xi \in \Xi} P(\xi) \log P(\xi) \tag{24}$$

subject to the normalization constraint:

$$\sum_{\xi \in \Xi} P(\xi) = 1 \tag{25}$$

and matching the expected feature count vector to empirical expert observations $\boldsymbol{\phi}_{\text{expert}}$:

$$\sum_{\xi \in \Xi} P(\xi) \boldsymbol{\phi}(\xi) = \boldsymbol{\phi}_{\text{expert}} \tag{26}$$

where $\boldsymbol{\phi}(\xi) = \sum_{t=0}^{T-1} \boldsymbol{\phi}(s_t, a_t)$ represents the cumulative trajectory feature vector.

To accommodate noisy observations and potential constraint violations without probability collapse, hard constraints are relaxed using the soft indicator function $\mathbb{I}_\eta^C(s_t, a_t)$:

$$\mathbb{I}_\eta^C(s_t, a_t) = \begin{cases} 1 & \text{if } (s_t, a_t) \notin C \\ \eta & \text{if } (s_t, a_t) \in C \end{cases} \tag{27}$$

The cumulative soft indicator along trajectory ξ is given by:

$$\mathbb{I}_\eta^C(\xi) = \prod_{t=0}^{T-1} \mathbb{I}_\eta^C(s_t, a_t) \tag{28}$$

Forming the Lagrangian $\mathcal{L}(P, \boldsymbol{w}_k, \lambda)$ with dual multiplier vector $\boldsymbol{w}_k$ for feature matching, scalar multiplier λ for normalization, and log-penalties for constraint violations:

$$\mathcal{L}(P, \boldsymbol{w}_k, \lambda) = \sum_{\xi \in \Xi} P(\xi) \left(-\log P(\xi) + \boldsymbol{w}_k^\top \boldsymbol{\phi}(\xi) + \log \mathbb{I}_\eta^C(\xi) \right) + \lambda \left(\sum_{\xi \in \Xi} P(\xi) - 1 \right) - \boldsymbol{w}_k^\top \boldsymbol{\phi}_{\text{expert}} \tag{29}$$

Setting the functional derivative of $\mathcal{L}$ with respect to $P(\xi)$ to zero:

$$\frac{\partial \mathcal{L}}{\partial P(\xi)} = -\log P(\xi) - 1 + \boldsymbol{w}_k^\top \boldsymbol{\phi}(\xi) + \lambda + \log \mathbb{I}_\eta^C(\xi) = 0 \tag{30}$$

Solving for $\log P(\xi)$:

$$\log P(\xi) = \boldsymbol{w}_k^\top \boldsymbol{\phi}(\xi) + \log \mathbb{I}_\eta^C(\xi) + (\lambda - 1) \tag{31}$$

Exponentiating both sides yields the unnormalized probability density:

$$P(\xi \mid C, \boldsymbol{w}_k) \propto \exp\left(\boldsymbol{w}_k^\top \boldsymbol{\phi}(\xi) \right) \cdot \mathbb{I}_\eta^C(\xi) \tag{32}$$

Decomposing $\boldsymbol{\phi}(\xi)$ and $\mathbb{I}_\eta^C(\xi)$ into explicit step-wise components over horizon T:

$$P_{\text{unnorm}}(\xi \mid C, \boldsymbol{w}_k) = \exp\left(\sum_{t=0}^{T-1} \boldsymbol{w}_k^\top \boldsymbol{\phi}(s_t, a_t) \right) \prod_{t=0}^{T-1} \mathbb{I}_\eta^C(s_t, a_t) \tag{33}$$

Normalizing across all possible trajectories starting at initial state s_0 yields the partition function $Z(C, \boldsymbol{w}_k) = \exp(1 - \lambda)$:

$$Z(C, \boldsymbol{w}_k) = \sum_{\xi' \in \Xi} \exp\left(\sum_{t=0}^{T-1} \boldsymbol{w}_k^\top \boldsymbol{\phi}(s'_t, a'_t) \right) \prod_{t=0}^{T-1} \mathbb{I}_\eta^C(s'_t, a'_t) \tag{34}$$

Dividing the unnormalized path density by $Z(C, \boldsymbol{w}_k)$ completes the derivation:

$$P_{\text{path}}(\xi \mid C, \boldsymbol{w}_k) = \frac{1}{Z(C, \boldsymbol{w}_k)} \exp\left(\sum_{t=0}^{T-1} \boldsymbol{w}_k^\top \boldsymbol{\phi}(s_t, a_t) \right) \prod_{t=0}^{T-1} \mathbb{I}_\eta^C(s_t, a_t) \tag{35}$$

This derivation generalizes the Maximum Entropy Inverse Reinforcement Learning framework of Ziebart et al. by incorporating soft multiplicative penalties to model stochastic constraint violations without zero-probability collapse.

C Additional Experiments

C.1 Scalability effect on runtime:

Figure-8 illustrates the execution time (in seconds) of the MOCI algorithm as the complexity of the environment increases from a 5×5 grid to a 10×10 grid. The blue line (short trajectory) represents an optimal direct path to the goal, where the horizon scales roughly twice the dimension of the grid ($H \approx 2N$).

The red line (long trajectory) represents a sub-optimal or highly exploratory path, where the horizon could scale at five times the grid dimension ($H \approx 5N$). Both trajectories show increasing computational cost as the grid size grows. For the short trajectory, runtime increases from approximately 0.35 seconds on a 5×5 grid to 1.25 seconds on a 10×10 grid. For the long trajectory, runtime increases from approximately 0.35 seconds to 1.65 seconds over the same range. The larger increase for the long trajectory reflects the additional computational cost associated with processing longer trajectories.

Table 3: MOCI performance progression across EM iterations (mean ± std). Best values per metric are highlighted in **bold**.

	Constraint Recovery				Reward / Cluster Recovery		
EM Iters	**MSE** (↓)	**Precision** (↑)	**Recall** (↑)	**F1** (↑)	**Cos. Sim.** W_1 (↑)	**Cos. Sim.** W_2 (↑)	**Cluster ARI** (↑)
1	0.0156 ± 0.0000	1.0000 ± 0.0000	0.8000 ± 0.0000	0.8889 ± 0.0000	0.6588 ± 0.1693	0.2727 ± 0.2100	0.2500 ± 0.1468
3	0.0156 ± 0.0000	1.0000 ± 0.0000	0.8000 ± 0.0000	0.8889 ± 0.0000	0.7076 ± 0.1380	0.5373 ± 0.1797	0.3913 ± 0.1047
6	0.0156 ± 0.0000	1.0000 ± 0.0000	0.8000 ± 0.0000	0.8889 ± 0.0000	0.7810 ± 0.0970	0.6927 ± 0.1258	0.4582 ± 0.1013
9	0.0156 ± 0.0000	1.0000 ± 0.0000	0.8000 ± 0.0000	0.8889 ± 0.0000	0.8191 ± 0.0611	0.7665 ± 0.0876	$\mathbf{0.4991 \pm 0.0749}$
12	0.0156 ± 0.0000	1.0000 ± 0.0000	0.8000 ± 0.0000	0.8889 ± 0.0000	0.8319 ± 0.0490	$\mathbf{0.7863 \pm 0.0771}$	0.4979 ± 0.1070
15	0.0156 ± 0.0000	1.0000 ± 0.0000	0.8000 ± 0.0000	0.8889 ± 0.0000	$\mathbf{0.8364 \pm 0.0444}$	$\mathbf{0.7863 \pm 0.0740}$	0.4977 ± 0.1046

C.2 Ablation: constraint detection vs. reward/cluster recovery across EM iterations

To characterize the computational budget required by each component of MOCI, we evaluate performance on the GridWorld benchmark across a range of EM iterations (1 to 15) over 15 random seeds (Table 3).

Constraint identification (Table 3, left) is not compute-bound in this environment. All four constraint metrics—MSE (0.0156), precision (1.0), recall (0.8), and F1 score (0.889)—reach their final values within a single EM iteration and remain completely static ($\sigma = 0$) regardless of further optimization.

In contrast, latent per-expert reward inference (Table 3, right) and demonstration clustering improve significantly with additional EM iterations. Increasing the budget from 1 to 9 iterations yields marked performance gains: cosine similarity rises from 0.66 to 0.82 for W_1 (Grass) and from 0.27 to 0.77 for W_2 (Rock), while the cluster Adjusted Rand Index (ARI) increases from 0.25 to 0.50 alongside reduced variance across seeds. Beyond 9 iterations, metrics plateau.

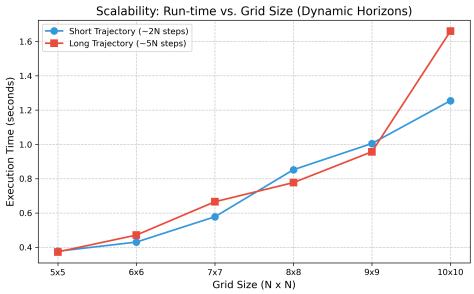

Figure 8: Scalability analysis of run-time versus grid size with dynamically scaling trajectory lengths.

These findings indicate that while constraint recovery requires minimal computation, parameter estimation for latent reward preferences requires iterative refinement. Consequently, we adopt 9–10 EM iterations as a practical default budget—capturing virtually all achievable reward-recovery gains without wasting compute on redundant iterations.

C.3 Ablation: latent mixture versus a single preference model

Heterogeneous demonstrations are modeled as a mixture over K latent linear preferences with shared hard constraints. A natural baseline is to *remove* mixture capacity by setting $K{=}1$, i.e., to explain the same pooled trajectories with a single MaxEnt component while retaining the remainder of the MOCI procedure (EM updates, constraint search, and the same hyperparameters otherwise). We instantiate MOCI with $K{\in}\{1, 2, 3\}$ on the *same* unlabeled dataset $\mathcal{D}$ for each random demo seed: trajectories are drawn i.i.d. from three expert policies with distinct terrain preferences, then pooled and shuffled. This isolates whether mixture structure materially affects fit and constraint recovery when the generative process truly uses three preference types.

To evaluate the effect of latent mixture capacity when scaling expert heterogeneity, we test MOCI with $K \in \{1, 2, 3\}$ on demonstrations pooled from three distinct experts (Grass, Rock, and Sand) under a shared Water constraint (Table 4). Each setting is evaluated across 50 independent demonstration seeds.

Moving from a single-preference model ($K = 1$) to a multi-component mixture ($K \geq 2$) substantially improves average marginal log-likelihood per trajectory, jumping from -166.6964 ± 0.0113 at $K = 1$

Table 4: Performance comparison across cluster settings K (mean $\pm$ std). Best-performing values per metric are highlighted in **bold**.

Metric	$K = 1$	$K = 2$	$K = 3$
Avg. Log-Likelihood (↑)	-166.6964 ± 0.0113	$\mathbf{-166.6454 \pm 0.0389}$	-166.6562 ± 0.0441
CMSE (↓)	0.0156 ± 0.0000	0.0150 ± 0.0031	$\mathbf{0.0141 \pm 0.0047}$
Precision (↑)	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
Recall (↑)	0.8000 ± 0.0000	0.8080 ± 0.0392	$\mathbf{0.8200 \pm 0.0600}$
F1 Score (↑)	0.8889 ± 0.0000	0.8933 ± 0.0218	$\mathbf{0.9000 \pm 0.0333}$

to -166.6454 ± 0.0389 at $K = 2$ and remaining near peak fit at $K = 3$ (-166.6562 ± 0.0441). This confirms that latent mixture capacity is essential for accurately modeling multi-expert demonstration data.

Unlike the two-expert scenario, increasing K in the three-expert setting yields progressive gains across constraint detection metrics. As K increases from 1 to 3, CMSE decreases monotonically from 0.0156 ± 0.0000 to 0.0141 ± 0.0047, Recall rises from 0.8000 ± 0.0000 to 0.8200 ± 0.0600, and the F1 Score improves from 0.8889 ± 0.0000 to 0.9000 ± 0.0333. Precision remains strictly 1.0000 ± 0.0000 across all values of K. Consequently, all recovery gains stem from successfully identifying additional true hard-constraint states rather than introducing false positives. The single-mixture baseline ($K = 1$) exhibits exactly zero variance across seeds, as it consistently converges to the same degenerate fit. In contrast, $K = 2$ and $K = 3$ introduce non-zero standard deviations across seeds, reflecting the varying difficulty of disentangling three distinct expert preference distributions from finite demonstration samples.

Overall, under-specifying K carries an increasing penalty as expert heterogeneity grows, reinforcing the need for sufficient mixture capacity to isolate shared constraints from unobserved preference variations.

D Details of the Lung Cancer Clinical Dataset

We generated a synthetic longitudinal Non-Small Cell Lung Cancer (NSCLC) cohort (1,000 patients $\times$ 5 timepoints = 5,000 patient-visits) using a sequential, Quantitative systems pharmacology (QSP) (Wang et al., 2023) mechanism-based simulation rather than independent random sampling of each field. Each patient record includes demographic variables (age, sex, smoking history), tumor characteristics (TNM stage, metastatic status, tumor size), molecular markers (EGFR mutation, ALK translocation, PD-L1 expression), physiological covariates, and treatment-related variables (ECOG performance status, CTCAE toxicity grade, treatment line, treatment intensity, and treatment response). Four candidate treatment actions were modeled including; targeted therapy, immunotherapy, chemotherapy, and surgery, each governed by a binary eligibility/feasibility indicator derived from guideline-concordant clinical rules (e.g., targeted therapy requires EGFR or ALK alteration; immunotherapy requires PD-L1 $> 1\%$; surgical eligibility requires early T-stage, absence of metastasis, and adequate performance status). Longitudinal outcomes included quality-of-life scores (0–100 scale), RECIST-based treatment response, and time-to-event survival data with a 24-month cumulative mortality of 27.2%.

The lung cancer Markov Decision Process (MDP) models longitudinal treatment planning for non-small cell lung cancer patients based on clinical visits and observational patient trajectories.

Each state represents a patient-visit defined by a 9-element:

$$s = (\text{month}, \text{perf_idx}, \text{tumor_idx}, \text{qol_idx}, \text{risk_idx}, \text{pdl1_idx}, \text{egfr_mut}, \text{alk}, \text{tox_idx})$$

- **`month` (Visit Month / Timepoint)**: Tracks longitudinal treatment duration by recording the elapsed time in months from baseline/diagnosis (`timepoint_month`).
- **`perf_idx` (Performance Status Index)**: Discretized ECOG performance status (`perf_status`), measuring the patient's physical capability, daily functionality, and functional reserve.
- **`tumor_idx` (Tumor Burden Index)**: Discretized index of primary tumor dimension/volume (`tumor_size`), tracking localized disease burden and response to therapy.

Table 5: Ground-truth treatment eligibility rules used for constraint evaluation.

Ineligibility Condition	Reason Label
Targeted Therapy	
ctcae_toxicity_grade ≥ 3	toxicity_grade>=3
$\neg$(egfr_mutation $= 1 \vee$ alk_translocation $= 1$)	EGFR/ALK not eligible
Immunotherapy	
ctcae_toxicity_grade ≥ 3	toxicity_grade>=3
pdl1_expression_pct < 1.0	PD-L1 < 1%
Chemotherapy	
ctcae_toxicity_grade ≥ 3	toxicity_grade>=3
egfr_lab_ml_min < 50	eGFR < 50
fev1_pct_predicted < 45	FEV1 < 45%
perf_status > 2	ECOG/perf_status > 2
Surgery	
timepoint_month > 0	not first-line month
ctcae_toxicity_grade ≥ 3	toxicity_grade>=3
perf_status > 2	ECOG/perf_status > 2
fev1_pct_predicted < 50	FEV1 < 50%

- **qol_idx (Quality of Life Index)**: Discretized patient-reported Quality of Life score (QoL), reflecting subjective health state, functional capacity, and symptom distress.
- **risk_idx (Event / Metastatic Risk Index)**: Discretized stratification score derived from disease progression risk and metastatic event indicators (event_indicator).
- **pdl1_idx (PD-L1 Expression Index)**: Discretized level of PD-L1 tumor cell expression percentage (pdl1_expression_pct), serving as the primary biomarker for immunotherapy response.
- **egfr_mut (EGFR Mutation Status)**: Binary indicator (0 = wild-type/absent, 1 = mutated/present) for EGFR gene mutations, determining eligibility for EGFR-targeted therapies.
- **alk (ALK Translocation Status)**: Binary indicator (0 = wild-type/absent, 1 = rearranged/present) for ALK gene translocations, determining eligibility for ALK-targeted therapies.
- **tox_idx (Toxicity Grade Index)**: Discretized index representing the CTCAE toxicity grade (ctcae_toxicity_grade), quantifying treatment-induced adverse event severity.

Action Space ($\mathcal{A}$): A discrete set of 4 primary oncology treatment options ($|\mathcal{A}| = 4$): **Targeted** (Targeted therapy); **Immuno** (Immunotherapy); **Chemo** (Chemotherapy); **Surgery** (Surgical intervention). Feature Map & Rewards ($\phi(s, a)$) uses a 2-dimensional feature vector $\phi(s, a) = [f_{\text{QoL}}, f_{\text{Survival}}]^\top$. f_{QoL}: Mean normalized patient Quality of Life score (scaled to $[0, 1]$). f_{Survival}: Mean normalized patient survival duration.

D.1 TREATMENT ELIGIBILITY RULES

Table 5 defines a formal set of domain-specific clinical eligibility criteria designed to govern treatment infeasibility for non-small cell lung cancer (NSCLC) management. These hard and conditional action-state constraints were drafted with LLM assistance from published clinical practice (Colice et al., 2007) and guidelines established by the National Comprehensive Cancer Network (NCCN), then reviewed for consistency with standard eligibility criteria. Clinically, the rules translate medical protocols into algorithmic boundaries.

E Related Work

Inverse Reinforcement Learning (IRL) infer an underlying reward function in a Markov Decision Process (MDP) directly from expert demonstrations. Various foundational paradigms have been developed to address the IRL problem, including maximum margin optimization (Ng et al., 2000),

Maximum Entropy (Ziebart et al., 2008), Bayesian inference (Ramachandran et al., 2007), and adversarial learning techniques (Fu et al., 2017).

Although classical IRL recovers a reward function to rationalize expert demonstrations (Ziebart et al., 2008), safety is often better expressed through explicit feasibility limits rather than soft reward shaping. Early foundational work by (Chou et al., 2018) addressed the forward optimal control problem using hit-and-run sampling to generate lower-cost trajectories that respect learned constraints while partitioning discrete state spaces to fit a feasibility classifier. (Chou et al., 2020) extended this framework to continuous state spaces by employing parametric constraint functions and uncertainty-aware constraints driven by robust optimization and Bayesian inference. (Papadimitriou et al., 2022) introduced an incremental greedy constraint inference algorithm that progressively expands the constraint set by minimizing the Karush-Kuhn-Tucker (KKT) residuals of the underlying optimal control problem.

MLCI (Scobee & Sastry, 2020) infers hard constraints by greedily improving trajectory likelihood under a maximum-entropy expert model, but treats the task reward as known and effectively explains data from a single behavior model in a deterministic environment. McPherson et al. (2021) extended this work to stochastic environment. ICRL (Malik et al., 2021) removes tabular and known-dynamics limitations by learning high-dimensional constraint representations, yet the standard setup still matches demonstrations with one constraint hypothesis and one expert type. In contrast, we drop the known-reward assumption and explain the same dataset with a mixture of linear preferences while inferring constraints.

F MOCI ALGORITHM

Algorithm 1: Multi-Objective Constraint Inference (MOCI)

Input : CMOMDP $\mathcal{M}$, Demos $\mathcal{D} = \{\xi_i\}_{i=1}^N$, Clusters K, Penalty λ, Max Iters I_{EM}, Options $\mathcal{O}$
Output : Constraints C, Weights $\{\boldsymbol{w}_k\}_{k=1}^K$, Priors $\{\pi_k\}_{k=1}^K$

$C \leftarrow \emptyset$; $\quad \boldsymbol{w}_k \sim \mathcal{N}(0, 0.01\boldsymbol{I})$, $\quad \pi_k \leftarrow 1/K \quad \forall k \in \{1, \dots, K\}$;
if $\mathcal{O}$.*mode* $==$ "frozen" **then**
  Fit baseline weights $\{\boldsymbol{w}_k\}$ and priors $\{\pi_k\}$ on unconstrained MDP ($C = \emptyset$);
for *iter* $= 1$ ***to*** I_{EM} **do**
  if $\mathcal{O}$.*mode* $==$ "joint" **then**
    $\gamma_{i,k} \leftarrow \frac{\pi_k P(\xi_i|C,\boldsymbol{w}_k)}{\sum_{j=1}^K \pi_j P(\xi_i|C,\boldsymbol{w}_j)} \quad \forall i, k$; // Compute Responsibilities (E-step)
    $\pi_k \leftarrow \frac{1}{N}\sum_{i=1}^N \gamma_{i,k}$; $\quad \boldsymbol{w}_k \leftarrow \text{Clip}\left(\boldsymbol{w}_k + \frac{\alpha}{N}\sum_{i=1}^N \gamma_{i,k}(\boldsymbol{\phi}(\xi_i) - \mathbb{E}[\boldsymbol{\phi}]), -w_{\max}, w_{\max}\right) \quad \forall k$; // M-step
  Extract candidate units $\mathcal{C}_{\text{cand}}$ via soft screening and pre-screen $\mathcal{S}_{\text{safe}}$;
  if $\mathcal{O}$.*rank_by_discrepancy* **then**
    Rank $\mathcal{C}_{\text{cand}}$ descending by $\Delta\pi(s,a) = \pi_{\text{model}}(a|s) - \pi_{\text{expert}}(a|s)$;
  while $\mathcal{C}_{cand} \neq \emptyset$ **do**
    $u^* \leftarrow \arg\max_{u \in \mathcal{C}_{\text{cand}}} \mathcal{L}_{\text{total}}(C \cup u, \cdot)$; $\quad \Delta\mathcal{L}_{\text{total}} \leftarrow |\mathcal{D}| \cdot (\mathcal{L}_{\text{avg}}(C \cup u^*, \cdot) - \mathcal{L}_{\text{avg}}(C, \cdot))$;
    if $\Delta\mathcal{L}_{total} - \lambda > 0$ **then**
      $C \leftarrow C \cup u^*$;
      if $\mathcal{O}$.*generalize* $==$ "expand" **then**
        Expand C to all matching signature pairs $\sigma(s', a) = \sigma(u^*)$;
      $\mathcal{C}_{\text{cand}} \leftarrow \mathcal{C}_{\text{cand}} \setminus \{u \mid u \subseteq C\}$;
    else
      break ; // Stop when gain fails penalty threshold
  if $\mathcal{O}$.*mode* $==$ "frozen" **then**
    break ; // Single pass over frozen baseline
if $\mathcal{O}$.*mode* $==$ "frozen" ***and*** $\mathcal{O}$.*refit_iters* > 0 **then**
  Re-fit weights $\{\boldsymbol{w}_k\}$ under final constraints C;
return $C, \{\boldsymbol{w}_k\}, \{\pi_k\}$;